PROTOKOL KERBEROS V – ANALÝZA A KONFIGURÁCIA

PISARČÍK, Peter, (SK)

Abstrakt. Článok ponúka pohľad na centrálny autentifikačný protokol s názvom Kerberos vyvinutom na Massachusetts Institute of Technology, pričom sa hlavne venuje analýze protokolu vo verzii 5. Cieľom článku je aj v krátkosti predstaviť grafické webovo orientované konfiguračné rozhranie pre administráciu distribučného centra kľúčov MIT implementácie Kerbera V.

1 Úvod

Internet je v súčasnosti pre mnohých takmer každodenným „elektronickým chlebom“, ba pre určité skupiny medzinárodnej spoločnosti sa stal neodmysliteľnou súčasťou ich života. Toto konštatovanie zároveň paradoxne koreluje s konštatovaním možného zlyhania celosvetového sieťového média vo forme najrôznejších útokov – útokov ľudského charakteru. Práve z tohto titulu sa v progrese Internetu stále viac venuje pozornosť autentifikačným procesom, na základe ktorých je možné eliminovať negatívny dopad kyberokriminality. Systém Kerberos je silným autentifikačným mechanizmom, ktorý okrem bezpečnej autentifikácie ponúka aj službu jedného prihlásenia (single sign-on). Je teda jedným z vyhovujúcich riešení v procese zabezpečenia siete Internet.

2 Protokol Kerberos V

Pojem Kerberos

Kerberos (grécky $\kappa\varepsilon\rho\beta\varepsilon\rho o\varsigma$, Kérberos, latinsky Cerberus) je meno bytosti z gréckej a rímskej mytológie, ktorá je bežne zobrazovaná v podobe viachlavého psa. Úlohou tejto kreatúry bolo stráženie vstupnej brány do podsvetia, v ktorom vládol boh Hádes so svojou manželkou Perzephoneou.

História

Projekt Kerberos je pokračovaním projektu Athéna, ktorého vývoj a výskum začal v roku 1983. Projekt zastrešovali spoločnosti IBM, Digital Equipment Corporation a MIT. Cieľom projektu bolo vytvorenie počítačového prostredia, ktoré by bolo zložené až z tisíc pracovných staníc s využitím heterogénneho hardvéru; teda cieľom bolo zlepšenie kvality vzdelávania na inštitúte. Výsledkom projektu Athéna bolo vytvorenie mnohých technológií, ktoré sú široko využívané dodnes, ako je napríklad X Window Systém a protokol Kerberos. Keď projekt Athéna skončil v roku 1991 výpočtové prostredie bolo premenované na Athéna systém a je stále využívané mnohými v MIT komunite.

Ako už bolo vyššie uvedené, sieťový protokol Kerberos vznikol na výskumnej pôde projektu Athéna. Aktuálnou verziou protokolu je verzia 5, ktorá bola vytvorená Johnom Kohlom a Cliffordom Neumanom v roku 1993 a je dokumentovaná ako RFC 1510 (novelizovaná verzia RFC 4120).

Základné ciele protokolu Kerberos

1. **Obojstranná autentifikácia.** Nakoľko je protokol Kerberos autentifikačným protokolom, ponúka overenie identity klienta pristupujúceho k určitej službe, avšak okrem centrálnej autentifikácie klientov Kerberos ide ďalej, keď ponúka aj autentifikáciu opačným smerom, teda klientov dokáže uistiť o identite služieb, ku ktorým sa pripájajú.

2. **Ochrana hesiel.** Jedným z častých problémov sieťových protokolov je hrozba odpočúvania autentifikačných údajov. Preto bol Kerberos navrhnutý s ohľadom na ochranu autentifikačných údajov. V protokole Kerberos sa diskrétne heslo nikdy neposiela priamo po sieti, avšak sa špeciálnym spôsobom upravuje a následne využíva ako zdieľaný šifrovací kľúč, ako bude popísané ďalej.

3. **Systém jedného prihlásenia.** Systém jedného prihlásenia je najzaujímavejším bodom návrhu protokolu Kerberos. Vychádza z predpokladu, že užívateľ po prihlásení k svojmu počítaču využíva služby mnohých ďalších počítačov (sieťových systémov). Autentizovať sa pri prístupe ku každej z týchto služieb a ešte prípadne opakovane je prinajmenšom nepohodlné. Samozrejme teraz neberieme do úvahy, že niektoré aplikačné programy, z hľadiska zvýšenia užívateľského komfortu, ponúkajú uloženie hesiel, čo je potenciálne bezpečnostné riziko. Kerberos sa problém snaží vyriešiť prostredníctvom systému jedného prihlásenia – akonáhle sa niekto, alebo niečo, raz autentifikuje voči systému Kerberos, už nie je potrebná žiadna ďalšia autentifikácia pri prístupe k službám, ktoré využívajú systém Kerberos.

Inovácie v protokole Kerberos V

1. **Otvorenosť pre nové kryptografické algoritmy** – dátové štruktúry sú navrhnuté tak,

aby bola možná implementácia akéhokoľvek kryptografického algoritmu, na rozdiel od verzie 4, v ktorej bola dátová štruktúra pevne zviazaná s algoritmom DES. Výhodou oproti verzii 4 je tiež možnosť využitia rozdielnych kryptografických algoritmov v každej z nasledujúcich správ: lístok, odpoveď, relačný kľúč.

2. **Zápis protokolu pomocou technológie ASN.1** (Abstract Syntax Notation One), ASN.1 popisuje reprezentáciu štruktúry dát, kódovanie a dekódovanie dát a ich prenos. Je súborom formálnych pravidiel umožňujúcich popis objektov nezávisle na ich technickej architektúre (viď. piata sekcia RFC 4120 dokumentu).

3. **Spätná kompatibilita s verziou 4**, zabezpečená tzv. 5-to-4 transformátorom lístkov (implementačne známe ako krb524).

4. **Zmena tvaru principálov**[1] na nasledujúcu formu:
   meno_užívateľa/meno_inštancie@názov_kerberovej_domény

5. **eliminovanie dvojitého šifrovania**, ktoré sa vyskytovalo v komunikácii autentifikačného servera (AS) a servera pre prideľovanie lístkov (TGS)

6. **Prepožičiavateľné lístky** (forwardable) – užívateľ môže požiadať o tento typ lístka, ktorý mu umožní jeho prepožičiavanie inému systému, čím sa daný systém stane oprávneným na základe tohto lístka požadovať služby bez nutnosti opätovného zadávania hesla užívateľa; špeciálnym prípadom takéhoto lístka je TGT[2].

7. **Proxy lístky** (proxiable) – sú podobné prepožičiavateľným lístkom v tom, že môžu byť prenesené na iný hostiteľský počítač. Avšak proxy TGT lístok môže byť použitý len na získanie lístka služby, v žiadnom prípade nemôže byť použitý na získanie nového TGT lístka pre vzdialený hostiteľský počítač.

8. **Obnoviteľné lístky** (renewable) – v Kerberos IV. bola životnosť lístkov limitovaná ako ochrana pred odcudzením. Kerberos V prináša dve rôzne schémy životnosti, ktoré kombinujú dlhú životnosť s bezpečnosťou lístkov s krátkou životnosťou. Ak užívateľ požiada o obnovenie lístka, získa lístok so štandardnou životnosťou a obnoviteľnú životnosť. Lístok je tak platný len počas štandardnej životnosti avšak môže byť predložený KDC[3] zo žiadosťou o predĺženie životnosti. KDC môže takúto žiadosť odmietnuť. Ak však je žiadosť schválená KDC vráti iný lístok, na základe ktorého sa užívateľ

[1] Je to reťazec, ktorý slúži k identifikácii užívateľa alebo služby. Každý principál má tvar: `meno/inštancia@REALM`, kde meno je obvykle užívateľské meno, alebo meno služby, REALM udáva meno kerberovej domény a inštancia je nepovinný reťazec, vďaka ktorému môže mať jeden užívateľ viac principálov.

[2] TGT (Ticket Granting Ticket) – je lístkom na lístky, čo znamená, že ide o špeciálny typ lístka, vďaka ktorému je možné získať ďalšie lístky. Až keď klient vlastní TGT lístok, ktorý sa získava od KDC, môže žiadať a získať lístky pre autentifikáciu voči rôznym sieťovým službám.

[3] KDC (Key Distribution Center) je centrum pre distribúciu kľúčov a je tak srdcom Kerbera. Úlohou KDC je manažovanie užívateľských účtov a účtov aplikačných serverov. Zároveň sa stará o prihlasovanie užívateľov do kerberovej domény. Tento pojem (KDC) však v sebe skrýva dve služby a to: autentifikačná služba (AS –

môže ďalej identifikovať. Tento proces môže byť opakovaný až kým obnoviteľnosť lístka úplne neskončí.

9. **Lístky s preddefinovanou životnosťou** – každý lístok môže byť akceptovaný len v čase ktorý je definovaný lístkom. Štandardne sa pri požiadaní o vydanie lístka vydá lístok platný od aktuálneho času s konfiguračne preddefinovanou časovou exspiráciou. V prípade lístkov s preddefinovanou životnosťou ide o určenie začiatku platnosti lístka v budúcnosti. Tento typ lístka sa zvykne využívať napríklad v prípade periodického spúšťania určitých sieťových systémových služieb. Napriek uvedenej výhode, tento typ lístka nie je často v praxi využívaný a niektoré implementácie ako napr. Active Directory od spoločnosti Microsoft túto podporu nezahŕňajú.

10. **Preautentifikácia** – pôvodný protokol Kerberos IV. nebol odolný voči lokálnym útokom hrubou silou a slovníkovým útokom. Tento nedostatok spôsobil, že bolo možné získať kombináciu užívateľského mena a hesla (autentifikačný server KDC vždy odoslal šifrovanú správu, na ktorú bolo možné off-line aplikovať spomínané útoky). K eliminácii týchto typov útokov došlo v protokole Kerberos V zavedením tzv. preautentifikácie. Preautentifikácia vyžaduje, aby žiadateľ dokázal svoju identitu pred tým ako mu KDC vydá TGT lístok.

    V špecifikácii protokolu sa nachádza niekoľko typov preautentifikácií, avšak reálne došlo k implementácii len šifrovaného časového odtlačku (PA-ENC-TIMESTAMP). Preautentifikácia je riadená zo strany KDC špeciálnymi pravidlami. Ak užívateľ požaduje získanie TGT lístka vo forme autentifikácie voči AS, ale KDC vyžaduje preautentifikáciu, zašle KDC žiadateľovi lístka chybovú správu (KRB_ERROR) namiesto štandardnej AS_REP. Táto chybová správa hovorí klientovi, že je nutná preautentifikácia. Klient musí teda vygenerovať požadované autentifikačné dáta a znova odoslať AS_REQ správu spolu s preautentifikačnými dátami. Ak je preautentifikácia akceptovaná, začína štandardná výmena správ protokolu Kerberos.

Činnosť protokolu Kerberos V

Protokol Kerberos pracuje vo viacerých fázach. Súhrne môžeme hovoriť, že ide o tri fázy, pričom každá z fáz sa skladá z otázky a odpovede.

AS_REQ

Ide o počiatočnú požiadavku zo strany klienta adresovanú autentifikačnému serveru, ktorej cieľom je získanie TGT lístka. V tejto fáze klient zasiela svoje autorizačné dáta. Celá táto požiadavka putuje sieťou bez šifrovania a vyzerá nasledovne:

AS_REQ = (PrincipálKlient , PrincipálSlužba , IP, LT, (TS))

Authentication Service) a služba pre výdaj lístkov (TGS – Ticket Granting Service). V niektorých sieťach je viac ako jedno KDC. V tomto prípade hovoríme o KDC hlavnom (master) a KDC podriadených (slave).

PrincipálKlient je principál asociovaný s užívateľom, ktorý sa autentifikuje; PrincipálSlužba je principál asociovaný so službou, o ktorú klient žiada (ide o reťazec kbrbtgt/REAL@REALM); IP_zoznam (IP) je zoznamom IP adries, ktoré určujú hostiteľský počítač, na ktorom je možné využiť získaný lístok; a nakoniec životnosť (LT) určuje maximálny platný čas pre lístok, ktorý sa bude používať. Je potrebné podotknúť, že hoci sa zdá zbytočné pridávať do požiadavky principál služby, keď v konečnom dôsledku je jasné, že pôjde o KRBTGT principál, predsa je potrebné si uvedomiť, že toto miesto môže byť využité na zadanie konkrétnej služby, ktorú jedinú chce užívateľ využiť, a preto žiada autentifikačný server priamo o lístok pre konkrétnu službu. Tým sa preskočí fáza žiadania TGS.

Ďalšia skutočnosť sa týka IP zoznamu, ktorý môže byť aj prázdny. V tomto prípade môže byť získaný lístok využitý na akomkoľvek hostiteľskom počítači. Toto riešenie umožňuje klientom, ktorí sa nachádzajú za NAT bezproblémovo využívať požadované služby; Niekedy klient pridáva do požiadavky aj svoj aktuálny čas – časové razítko (TS), čo je však len z dôvodu, že KDC, ak časové razítko klienta nie je v povolenom rozsahu, môže klienta hneď varovať, že pre využívanie služieb daného systému Kerberos nie je synchronizovaný.

AS_REP

Ide o odpoveď na predchádzajúcu požiadavku, ktorú odosiela autentifikačný server klientovi, potom čo prekontroloval či sa principál klienta a služby nachádzajú v KDC databáze (ak čo i len jeden principál v databáze chýba je klientovi zaslané chybové hlásenie). Autentifikačný server vytvára odpoveď nasledovne:

1. Náhodne vytvorí relačný kľúč, ktorý sa stáva tajným kľúčom zdieľaným klientom a TGS (SKTGS);

2. Vytvorí TGT lístok, ktorý obsahuje klientsky principál a principál služby, ďalej zoznam IP adries (dáta sú preberané z AS_REQ). Pridá tiež dátum a čas ako časové razítko (TS), životnosť (LT) a nakoniec relačný kľúč (session key - SKTGS), čím vznikne nasledujúca konštrukcia:

$$\mathbf{TGT = (PrincipálKlient, krbtgt/REALM@REALM, IP, TS, LT, SKTGS)}$$

3. Autentifikačný server vygeneruje a odošle klientovi odpoveď: TGT lístok, ktorý bol vyššie popísaný, zašifrovaný s použitím tajného kľúča TGS (KTGS); principál_služby (krbtgt/REALM@REALM), časové razítko (TS), životnosť (LT) a relačný kľúč (SKTGS), to všetko šifrované použitím tajného kľúča klienta (využitím funkcie string2key), ktorý žiadal o službu. Teda odpoveď vyzerá:

$$\mathbf{AS\_REP = \{\ PrincipálSlužba, TS, LT, SKTGS\}KKlient\ \{\ TGT\ \}KTGS}$$

Z týchto konštrukcií môže na prvý pohľad vyvstávať otázka duplicity informácií, ale je potrebné si uvedomiť, že tým že sú informácie v TGT šifrované použitím tajného kľúča servera, nie sú čitateľné pre klienta, a preto musia byť zopakované, čo slúži pre verifikáciu spojenia medzi klientom a KDC.

Vo chvíli, keď klientský počítač prijme odpoveď autentifikačného servera vyžiada si od užívateľa heslo. Heslo slúži ako jeden zo vstupných parametrov, na základe ktorých sa vytvorí špeciálny reťazec, ktorý využíva funkcia string2key(), ktorej výstupom je tajný kľúč klienta. Takto sa dešifruje časť správy, ktorú KDC zašifrovalo využitím tajného kľúča klienta. Ak teda užívateľ je skutočne tým, za ktorého sa vydáva a teda zadal správne heslo, dešifrovanie je úspešné a on získava relačný kľúč a TGT, prostredníctvom ktorých môže žiadať o ďalšie služby (lístky).

TGS_REQ

Užívateľ sa v predchádzajúcej fáze úspešne autentifikoval a získal teda TGT lístok. V tejto chvíli vlastní užívateľ lístok (TGT), prostredníctvom ktorého môže žiadať od TGS (KDC) lístky pre rôzne služby, voči ktorým je samozrejme oprávnený. Práve to je druhá fáza, ktorá začína žiadosťou klienta, ktorú odosiela TGS v podobe TGS_REQ, v ktorej žiada o lístok pre konkrétnu službu.

Požiadavka je konštruovaná nasledovne. Najprv sa vytvára tzv. „autentifikátor“ zložený z užívateľského principála a časového razítka (TS) klientského počítača, pričom tieto dva údaje sú šifrované relačným kľúčom (SKTGS), ktorý užívateľ prijal v predchádzajúcej fáze:

$$\textbf{Autentifikátor} = \{\textbf{PrincipálKlient, TS}\}\textbf{SKTGS}$$

Následne sa vytvorí celá požiadavka, ktorá obsahuje: principál služby, o ktorej lístok sa žiada, životnosť (LT) a autentifikátor, ktorý bol popísaný v predchádzajúcom bode, k čomu sa pripája TGT lístok, ktorý je šifrovaný kľúčom TGS (KTGS)

$$\textbf{TGS\_REQ} = (\ \textbf{PrincipálSlužba, LT, Autentifikátor}) \{\ \textbf{TGT}\ \}\textbf{KTGS}$$

TGS_REP

Vo chvíli keď príde požiadavka klienta na TGS, TGS najprv prekontroluje či principál požadovanej služby existuje v KDC databáze. Ak existuje, z databázy sa načíta tajný kľúč, ktorým sa dešifruje TGT, čím sa extrahuje relačný kľúč, ktorý sa využije na dešifrovanie autentifikátora. Pred generovaním odpovede klientovi sa verifikujú nasledujúce podmienky:

- či platnosť TGT neskončila,
- či principál klienta obsiahnutý v autentifikátore koreluje s tým, ktorý sa nachádza v TGT,
- či autentifikátor sa nenachádza vo vyrovnávacej pamäti odpovedí,
- či zoznam IP adries nie je prázdny a v prípade, že nie, či zdrojová IP adresa žiadateľa sa nachádza v tomto zozname.

Vyššieuvedené podmienky potvrdzujú, že TGT skutočne patrí užívateľovi, ktorý vytvoril požiadavku, a teda TGS môže vytvoriť adekvátnu odpoveď. Táto prebieha v nasledujúcich krokoch:

1. Náhodne sa vytvorí relačný kľúč, ktorý bude tajným zdieľaným kľúčom medzi klientom a službou (SKSlužba).

2. Vytvorí sa lístok služby (TSlužba) obsahujúci: principál klienta, principál služby, zoznam IP adries (IP), časové razítko KDC (TS), životnosť (LT) a nakoniec relačný kľúč (SKSlužba)

TSlužba = (PrincipálKlient, PrincipálSlužba, IP, TS, LT, SKSlužba)

3. Vytvorí sa celková správa – odpoveď obsahujúca: lístok služby (ako ukazuje schéma vyššie) šifrovaný využitím tajného kľúča služby (KSlužba) a záznam obsahujúci principál služby, časové razítko (TS), životnosť (LT) a nový relačný kľúč (SKSlužba), všetko zašifrované s použitím relačného kľúča extrahovaného z TGT (SKTGS). Odpoveď teda vyzerá takto:

TGS_REP = { **PrincipálSlužba, TS, LT, SKSlužba**}**SKTGS** { **TSlužba**}**KSlužba**

Klient – žiadateľ príjme takúto odpoveď pričom aplikuje na ňu relačný kľúč, ktorý má uložený vo vyrovnávacej pamäti, a pomocou ktorého dešifruje časť odpovede, ktorá obsahuje nový relačný kľúč, prostredníctvom ktorého bude následne komunikovať s aplikačným serverom služby. Úlohou klienta v tomto kroku je: do svojej vyrovnávacej pamäte, vložiť nový relačný kľúč ako aj lístok služby, ktorý sa použije pri prístupe k danej službe.

AP_REQ

Klient, ktorý získal lístok pre prístup k požadovanej službe (t. j. lístok a príslušný relačný kľúč), sa v tejto fáze kontaktuje s aplikačným serverom pre prístup k požadovanej službe prostredníctvom AP_REQ správy. Táto správa je tvorená ad hoc na rozdiel od predchádzajúcej správy, do ktorej bol zapojený KDC, a teda variuje v závislosti od aplikácie (služby aplikačného servera). Môžeme však uvažovať nad nasledujúcou stratégiou:

1. Klient vytvorí autentifikátor obsahujúci užívateľský principál a časové razítko (TS), to všetko šifrované relačným kľúčom (SKSlužba), ktorý je zdieľaný s aplikačným serverom

Autentifikátor = { **PrincipálKlient , TS** }**SKSlužba**

2. Klient vytvorí správu požiadavky obsahujúcu: lístok služby (TSlužba), ktorý je šifrovaný tajným kľúčom danej služby (KSlužba) a autentifikátor, ktorý bol vytvorený klientom

AP_REQ = Autentifikátor { **TSlužba** }**KSlužba**

Vo chvíli doručenia takejto požiadavky, aplikačný server dešifruje lístok služby s použitím svojho tajného kľúča, na základe čoho získa relačný kľúč pre komunikáciu s klientom, a ktorý súčasne použije na dešifrovanie autentifikátora. Na overenie či žiadateľ je ten, za ktorého sa vydáva a súčasne či má žiadateľ právo pristúpiť k požadovanej službe, aplikačný server verifikuje nasledujúce podmienky:

- či platnosť lístku služby nevypršala,
- či principál klienta obsiahnutý v autentifikátore je totožný s tým, ktorý obsahuje lístok,
- či sa autentifikátor nenachádza vo vyrovnávacej pamäti, prípadne či mu nevypršala platnosť,
- či zoznam IP adries (extrahovaný z lístka) nie je prázdny a v prípade, že nie, či klient – žiadateľ komunikuje z niektorej z IP adries zoznamu.

AP_REP

Poslednou z trojice odpovedí v procese autentifikácie protokolom Kerberos je odpoveď aplikačného servera, v ktorej dosvedčuje klientovi, že je naozaj tým serverom, ktorý klient požadoval. Avšak táto správa nie je vždy požadovaná. Klient žiada o ňu len v prípade, ak je nevyhnutná obojstranná autentifikácia. Tým je zjavná jedna z výhod protokolu Kerberos, kedy sa neoveruje len identita klienta, ale súčasne aj identita aplikačného servera.

Bezpečnosť protokolu Kerberos V

Protokol Kerberos je kryptografickým protokolom, a tým spadá do oblasti exaktných empirických vied. Vyžaduje si teda formálny dôkaz svojej bezpečnosti. Takýto dôkaz je možné vytvoriť za pomoci špeciálnej metodiky, tzv. BAN logiky, alebo progresívnejšej GNY logiky. Obidve metodiky si kladú za cieľ formálne dokázať bezpečnosť objektu, pričom sa zameriavajú na: presnú definíciu cieľov, ktoré má objekt dosiahnuť; definíciu postupov a vzťahov subobjkektov objektu; dokazovanie celkovej bezpečnosti cez čiastkové činnosti; vylúčenie závislosti na nestabilných a neoverených predpokladoch atď. Protokol Kerberos V. je bezpečný nakoľko existuje formálny dôkaz jeho bezpečnosti realizovaný pomocou spomenutých metodík. V ďalšom texte sa však skôr pozrieme na konkrétnu implementáciu ochrany proti možným útokom.

Slovníkový útok a útok hrubou silou

Kerberos V. tým, že je otvorený pre najnovšie kryptografické algoritmy, efektívne predlžuje dobu pre uhádnutie kľúča; súčasne je pridaná podpora preautentifikácie, ktorá znemožňuje off-line útoky na vydané TGT lístky.

„Replay“ útok

Protokol Kerberos má niekoľko zabudovaných ochrán, ktorými predchádza úspešnosti „replay útoku“ a administrátor by nikdy nemal zanedbávať aktivovanie týchto ochrán:

1. **Zoznam adries v lístku** – ak klient požaduje lístok od KDC môže vložiť do žiadosti zoznam sieťových adries, z ktorých bude komunikácia platná. Tento zoznam sieťových adries je prenášaný cez celú komunikáciu protokolu Kerberos.

2. **Ochrana založená na čase** – ak klient požaduje využitie kerberizovanej služby vygeneruje súčasne autentifikátor, ktorý je odosielaný s lístkom k požadovanej službe ako súčasť autentifikácie. Autentifikátor obsahuje časové razítko, ktoré je šifrované relačným kľúčom generovaným KDC. Keď požadovaná služba získa autentifikátor dešifruje ho za pomoci získaného relačného kľúča a časové razítko overí voči svojmu lokálnemu času. Ak rozdiel týchto dvoch časov je viac ako päť minút služba zamietne lístok a odmietne autentifikovať užívateľa.

3. **„Replay" vyrovnávacia pamäť** – každá kerberizovaná služba udržiava vyrovnávaciu pamäť prijatých autentifikátorov. Ak služba prijme autentifikátor, ktorý sa už nachádza vo vyrovnávacej pamäti zamietne požiadavku o prístup. V opačnom prípade služba akceptuje požiadavku a autentifikátor pridá do vyrovnávacej pamäte.

Útok „man-in-the-middle"

Základným cieľom útoku je sfalšovanie identity požadovaného servera, ku ktorému sa klient chce pripojiť. Teda útok prebieha spôsobom, pri ktorom útočník skrýva svoju identitu a vydáva sa za aplikačný server, ku ktorému klient pristúpi. Týmto útočník začína komunikáciu s klientom. Následne sa útočník pokúsi upraviť správy zasielané užívateľom tak, aby získal prístup na reálne požadovaný server. Útočník je teda v pozícii prostredníka, na základe čoho sa nazýva aj tento útok. Dobrou správou je, že Kerberos protokol má už priamo zabudovanú ochranu proti tomuto typu útoku. Ako je známe Kerberos ponúka obojstrannú autentifikáciu, čo teda znamená, že nielen klient je povinný sa autentifikovať, ale aj aplikačný server musí potvrdiť svoju identitu (samozrejme ak je to vyžadované).

3 Konfiguračné rozhranie

Z predchádzajúcich kapitol je možné vidieť, že protokol Kerberos V. je silným nástrojom pre administrátorov počítačovej siete a zároveň užitočným pomocníkom pre tých, ktorý často pracujú s rozličnými sieťovými (ale aj lokálnymi) službami vyžadujúcimi autentifikáciu. A práve z tohto hľadiska bolo navrhnuté riešenie: rozhranie pre konfiguráciu protokolu Kerberos V., alebo presnejšie jeho MIT implementácie.

Nakoľko konfiguračné rozhranie sa neustále vyvíja, na tomto mieste by som rád uviedol len URL, kde sa projekt nachádza a odkiaľ je možné stiahnuť zdrojové kódy a nájsť zaujímavosti týkajúce sa vývoja spomínaného rozhrania

`http://project.xdata.sk/kerberos`.

4 Záver

Protokol Kerberos sa vyvíja už takmer 30 rokov a ponúka funkcionalitu, ktorá má čo ponúknuť aj súčasnému kyberopriestoru. A práve z tohto hľadiska bolo žiadúce vytvorenie

nástroja, ktorý by uľahčil implementáciu a konfiguráciu tohto protokolu do konkrétnych počítačových sietí a tým poukázal na možnosti, ktoré v sebe protokol Kerberos, zvlášť vo verzii V. skrýva. Verím, že vytvorený konfiguračný systém nájde uplatnenie v reálnej praxi, a stane sa každodenným pomocníkom mnohých sieťových administrátorov.

Literatúra

[1] BURDA, Z.: Kerberos: Instalace a použití. In.: `http://kerberos.zdenda.com` (2006)

[2] GARMAN, J.: Kerberos: The Definitive Guide. Ó Reilly : USA, 2003

[3] SMITH, R.: LINUX ve světe WINDOWS. GRADA : Praha, 2006

[4] MIT: In.: `http://web.mit.edu/kerberos/www/`

[5] RICCARDI F.: The Kerberos protocol and its implementations. In.: `http://www.kerberos.org/software/tutorial.html` (2007)

[6] MIGEON, J.: The MIT Kerberos Administrator's How-to Guide. MIT Kerberos Consortium, 2008

[7] DOSTÁLEK, L.: Velký průvodce protokoly TCP/IP: Bezpečnost. Computer Press, Praha, 2001

[8] WIKIPEDIA: In.: `en.wikipedia.org`; `cs.wikipedia.org`; `sk.wikipedia.org`

Kontaktná adresa

Peter PISARČÍK (RNDr., PhDr.),
Ústav informatiky, Prírodovedecká fakulta, UPJŠ v Košiciach,
Jesenná 5, 040 01 Košice,
`pisarcik@gmx.net`

Otvorený softvér vo vzdelávaní, výskume a v IT riešeniach
1.–4. júla 2010, Žilina, Slovensko

Organizátori: Miloš Šrámek, Spoločnosť pre otvorené informačné technológie
Tatiana Šrámková, Katedra fyziky, FEI STU Bratislava
Michal Kaukič, Aleš Kozubík, Tomáš Majer, Žilinská univerzita
Lýdia Gábrišová, Ľubica Michálková, Žilinská univerzita
Juraj Bednár, Digmia, Slovensko
Miloslav Ofúkaný, GeoCommunity, Slovensko
Peter Mráz, Kremnica
Slavko Fedorik, SOŠ elektrotechnická, Poprad
Peter Štrba, Spojená škola/Gymnázium M. Galandu, Turčianske Teplice
Ladislav Ševčovič, FEI, Technická univerzita v Košiciach

Editori: Michal Kaukič
Miloš Šrámek
Slavko Fedorik
Ladislav Ševčovič

Recenzenti: Mgr. Juraj Bednár
Mgr. Rudolf Blaško, PhD.
RNDr. Ján Buša, CSc.
Ing. Slavko Fedorik
Ing. Karol Grondžák, PhD.
Mgr. Michal Kaukič, CSc.
Ing. Tomáš Kliment
RNDr. Aleš Kozubík, PhD.
Mgr. Juraj Michálek
doc. RNDr. Štefan Peško, CSc.
Ing. Pavel Stříž, PhD.
RNDr. Ladislav Ševčovič
Ing. Michal Žarnay, PhD.

Vydavateľ: Spoločnosť pre otvorené informačné technológie – SOIT, Bratislava

ISBN 978-80-970457-0-8

Sadzba programom pdfTEX Ladislav Ševčovič
