Tehtävä 1: Pääsynvalvonta

Pääsynvalvonta on eräs tärkeimmistä tehtävistä luotaessa turvallista tietoteknistä ympäristöä. Pääsynvalvonnan tavoitteena on estää asiaankuulumattomien kirjautuminen tietokoneelle sekä luvaton ohjelmien ja tiedostojen käyttö. Tavallisimmin tämä tapahtuu siten, että tietokone tai ohjelma pyytää käyttäjältä *käyttäjätunnusta* ja *salasanaa*. Mikäli nämä annetaan oikein, käyttäjä pääsee kirjautumaan koneelle ja hänelle annetaan käyttäjätunnusta vastaavat oikeudet. Oikeudet määrittelevät, mitä hakemistoja ja tiedostoja käyttäjä pääsee katselemaan, mitä ohjelmia hän voi suorittaa ja mitä ulkoisia laitteita, esimerkiksi kirjoittimia tai ulkoisia muisteja, hän voi käyttää. Jatkossa hakemistoja, tiedostoja, ohjelmistoja ja ulkoisia laitteita kutsutaan yhteisellä nimellä *resursseiksi*.

Yksinkertaisin hyökkäys tietokonejärjestelmiä vastaan on yrittää arvata käyttäjätunnus ja salasana -yhdistelmä. Arvaaminen voidaan automatisoida, jolloin arvauksia on mahdollista tehdä valtava määrä kohtuullisessa ajassa. Nykyään arvaamisen yhteydessä käytetään niin sanottuja sanakirjoja. Nämä ovat laajoja listoja, joissa on esimerkiksi lähes kaikki kielen sanat. Tavallisten sanojen lisäksi kokeillaan muita, tavanomaisia yhdistelmiä, kuten sanoja lopusta alkuun sekä sana ja numero -yhdistelmiä. Tällaista hyökkäystä kutsutaan *raa'an voiman hyökkäykseksi*. Tällaisten hyökkäysten torjumiseksi suositellaan, että salasana ei ole minkään kielen sana, vaan mielivaltainen yhdistelmä kirjaimia, numeroita ja erikoismerkkejä. Lisäksi salasanan tulee olla riittävän pitkä, vähintään 10 tai 12 merkkiä.

Paitsi käyttäjän, myös käyttäjätunnuksen ja salasanan tarkistavan järjestelmän täytyy ottaa mahdolliset hyökkäykset huomioon. Käyttäjätunnukset, niihin liittyvät oikeudet ja salasanat on tallennettu tiedostoihin. Jos hyökkääjä onnistuu, esimerkiksi jonkin ohjelmiston tietoturvavirheen kautta, saamaan itselleen käyttöoikeudet järjestelmään, hän voi kenties lukea myös salasanatiedoston. Tästä syystä salasanoja ei saa tallentaa selväkielisessä muodossa. Salasanat ovat tiedostossa joko salakirjoitettuna tai nykyään yleisemmin niin sanottuina tiivisteinä. Tiiviste lasketaan mielivaltaisen pitkästä merkkijonosta ja tulos on määrämittainen, tyypillisesti 128, 256 tai 512 bitin mittainen bittijono (eli merkeistä 0 ja 1 koostuva jono). Tiivistefunktion tulee täyttää muutamia oleellisia ehtoja ennenkuin sitä voidaan käyttää salasanojen yhteydessä. Jos f on tiivistefunktio, M merkkijono, josta tiiviste lasketaan, ja $f(M)$ M:n tiiviste, niin tuloksesta $f(M)$ ei voida kohtuullisessa ajassa päätellä M:ää. Eli tiivistefunktio on *yksisuuntainen*: tiiviste on suhteellisen helppo laskea, mutta tiivisteestä ei ole mahdollista päätellä alkuperäistä merkkijonoa. Toinen tiivistefunktiolta vaadittava ehto on sellainen, ettei voida kohtuudella muodostaa kahta merkkijonoa, joiden tiivisteet ovat samoja. Eli ei ole mahdollista kohtuullisessa ajassa löytää kahta merkkijonoa M_1 ja M_2, joille pätee

$f(M_1) = f(M_2)$. Kohtuullinen aika tässä yhteydessä tarkoittaa vuosia.

Esimerkki yksinkertaisesta tiivistefunktiosta on seuraava. Olkoon M merkkijono ja L sen pituus. Pilkotaan M n:n pituisiin lohkoihin $M_1, M_2, ..., M_k$, missä n on paljon pienempi kuin L. Lohkoja on siis k kappaletta. Täytetään viimeinen lohko biteillä siten, että senkin pituus on sama kuin muiden eli n. Käsitellään nyt lohkoja bitti kerrallaan siten, että lohkojen vastaavat bitit lasketaan yhteen seuraavan säännön mukaan: $0 + 0 = 0$, $0 + 1 = 1$, $1 + 0 = 1$, $1 + 1 = 0$. Esimerkiksi $M = 1010\ 1110\ 0011\ 0101$, $L = 16$, $n = 4$, $k = 4$. Edelleen $1010+1110+0011+0101 = 0010$ ($1+1+0+0 = 0$, $0+1+0+1 = 0$, $1 + 1 + 1 + 0 = 1$, $0 + 0 + 1 + 1 = 0$). Tämä tiiviste on liian lyhyt, ja tiivistefunktio on liian yksinkertainen todelliseksi tiivistefunktioksi, mutta se toimii eräänlaisena perustana monimutkaisemmille tiivistefunktioille. Todellisissakin tiivistefunktioissa tehdään samanlaisia bittien yhteenlaskuja, mutta lisäksi muita operaatioita kuten esimerkiksi bittien paikkojen vaihtamista.

Pelkkä salasanojen korvaaminen tiivisteillä ei torju raa'an voiman hyökkäyksiä. Jotta tällaiset hyökkäykset eivät onnistuisi, tiivisteiden laskemisesta tehdään tahallaan hidasta. Esimerkiksi tiivistefunktio voidaan laskea useaan kertaan. Vaikka ihminen ei huomaakaan mitään hidastusta, niin todellisuudessa hidastus voi olla enemmän kuin satakertainen, mikä estää raa'an voiman hyökkäyksiä onnistumasta kohtuullisessa ajassa. Toinen yksinkertainen toimenpide on sellainen, ettei kirjautumisen epäonnistuessa ilmoiteta tarkasti, mikä meni väärin. Esimerkiksi ei ilmoiteta, että salasana oli väärin tai että käyttäjätunnus oli väärin, vaan yksinkertaisesti, ettei käyttäjätunnus-salasanapari kelpaa. Tällöin hyökkääjä ei tiedä, oliko jompikumpi oikein vai molemmat vääriä.

Toisenlainen tilanne syntyy, jos hyökkääjä saa anastettua salasanatiedoston. Nyt hyökkääjällä on tehokkaammat mahdollisuudet testata salasanoja ja edellä mainitun sanakirjan käyttö on tässä yhteydessä viety vielä pitemmälle. Hyökkääjillä on sanakirjasta valmiiksi laskettuja tiivisteitä järjestettynä sellaiseen muotoon, että on nopeaa verrata salasanatiedostosta löytyviä tiivisteitä valmiiksi laskettuihin tiivisteisiin. Tiivistetaulukoiden koko voi olla teratavun luokkaa. Jotta tällaisia hyökkäyksiä vaikeutettaisiin, lisätään salasanan yhteyteen ennen tiivisteen laskemista ylimääräinen bittijono, jota kutsutaan *suolaksi*. Suolan pituus on 64 tai 128 bittiä ja jokaiselle käyttäjälle luodaan oma yksilöllinen suola-arvo. Siten salasanatiedostossa (tai tiedostoissa, jos käyttäjätunnuksiin liittyvä tieto on jaettu useaan tiedostoon) on ainakin käyttäjätunnus, suola ja salasana-suola -tiiviste.

Pääsynvalvontaan kuuluu myös mekanismit, joilla pidetään kirjaa oikeuksista. Esimerkiksi tiedostojen yhteydessä järjestelmän täytyy tietää, kenellä on mitäkin oikeuksia tiedostoon. Tavallisimpia rakenteita tällaiseen kirjanpitoon ovat *pääsymatriisit, pääsylistat* ja *valtuuslistat*. Pääsymatriisi on $(m \times n)$-taulukko, jossa m on subjektien ja n objektien lukumäärä. Subjektit ovat tässä

käyttäjiä, joskin ne voivat olla toisissa yhteyksissä esimerkiksi myös ohjelmia. Objektit ovat puolestaan resursseja, jotka on aikaisemmin määritelty. Kohtaan (i, j) merkitään oikeus, joka subjektilla i on objektiin j. Ongelmana pääsymatriiseissa on se, että niistä tulee helposti hyvin suuria, vaikka useimmat kohdat ovat vailla oikeuksia. Pääsylistoissa sen sijaan säästyy tilaa. Objektiin liittyvässä pääsylistassa on kaikki subjektit, joilla on jokin oikeus objektiin. Subjektiin liittyvä valtuuslista sen sijaan on päinvastainen lista, jossa on kaikki objektit, joihin subjektilla on jokin oikeus.

Ennen pääsynvalvontamekanismien valintaa ja toteutusta on aiheellista hahmotella pääsynvalvonnan periaatteet. Tätä suunnittelua helpottavat erilaiset pääsynvalvontamallit. Kolme tärkeintä mallia ovat *sääntöpohjainen, yksilöpohjainen* ja *roolipohjainen* pääsynvalvonta. Sääntöpohjaisessa mallissa annetaan etukäteen säännöt, joiden mukaan oikeudet määräytyvät. Subjekti ei voi muuttaa objektin oikeuksia, vaikka vaikka hänellä olisi maksimaaliset oikeudet objektiin. Yksilöpohjaisessa mallissa subjektit voivat määritellä objektiensa oikeudet. Subjektit voivat myös muuttaa objektiensa oikeuksia. Roolipohjaisessa mallissa oikeudet ovat rooleilla. Roolin takana oleva subjekti voi vaihtua ja useammalla subjektilla voi olla sama rooli. Subjektilla voi olla myös monta roolia. Roolipohjainen malli voi muuten olla joko sääntö- tai yksilöpohjainen.

Lopuksi voidaan vielä todeta, ettei oikeuksien määritteleminenkään ole aivan suoraviivaista. Esimerkiksi tiedostoihin liittyy yleensä kolme oikeutta, lukuoikeus, kirjoitusoikeus ja suoritusoikeus. Samat oikeudet liittyvät myös hakemistoihin. Jo tässä syntyy tulkintaongelma: mitä suoritusoikeus tarkoittaa hakemistojen kohdalla? Toisenlainen ongelma syntyy, jos käyttäjällä on oikeus kirjoittaa tiedostoon, mutta ei kirjoitusoikeutta hakemistoon, jossa tiedosto sijaitsee. Saako hän tällöin kirjoittaa tiedostoon vai ei? Esimerkiksi Windows ja Linux ratkaisevat nämä ongelmat eri tavalla.

Linuxissa hakemiston lukuoikeus tarkoittaa, että subjekti voi tarkastella hakemiston sisältöä, toisin sanoen tarkastella, mitä tiedostoja ja alihakemistoja hakemistossa on. Kirjoitusoikeus hakemistoon tarkoittaa, että subjekti voi luoda tiedoston hakemistoon. Suoritusoikeus puolestaan merkitsee, että subjekti voi edetä hakemistosta sen alihakemistoon. Erityinen piirre Linuxissa on se, että tiedoston voi lukea vain, jos omaa suoritusoikeuden jokaiseen hakemistoon hakemistopolulla juuresta tiedostoon.

Windowsissa oikeudet ovat monimutkaisemmat. Oikeuksia on enemmän ja subjekteilla on suuremmat mahdollisuudet luoda sääntöjä esimerkiksi käyttäjäryhmille. Lisäksi Windowsissa on suoranaisia kieltosääntöjä, mitkä puuttuvat Linuxista. Linuxissa kaikki sellainen on automaattisesti kielletty, mitä ei ole erikseen sallittu. Erityisesti hakemistojen käsittely Windowsissa poikkeaa Linuxista: on mahdollista lukea tiedosto hakemistosta, vaikka hakemistoon ei olisikaan mitään oikeuksia.

Kysymykset

Kysymys 1. Jos näppäimistössä on m merkkiä, niin kuinka monta erilaista k:n mittaista salasanaa on olemassa? (maksimipistemäärä 2)

Kysymys 2. Kuvaile, minkä tyyppiset salasanat ovat huonoja, vaikka olisivatkin riittävän pitkiä. Anna myös hyvän salasanan tunnusmerkit. (maksimipistemäärä 3)

Kysymys 3. Materiaalissa esitetylle yksinkertaiselle tiivistefunktiolle on helppo keksiä sellaiset bittijonot, jotka tuottavat saman tiivisteen. Modifioidaan nyt tiivistefunktiota siten, että ennen yhteenlaskuja lohkoa i rullataan i askelta vasemmalle. Esimerkiksi jos 1. lohko on 1100, niin siitä tulee 1001. Jos lohko olisikin toinen lohko, siitä tulisi 0011. Jos se olisi kolmas lohko, niin tulos olisi 0110. Neljäs lohko pysyisi muuttumattomana. Laadi tälle modifioidulle tiivistefunktiolle kaksi erilaista 16 bitin bittijonoa, jotka tuottavat saman neljän bitin pituisen tiivisteen. (maksimipistemäärä 3)

Kysymys 4. Miten suola vaikeuttaa salasanojen arvaamista siinä tapauksessa, että salasanatiedosto on anastettu, vaikka käyttäjätunnus ja suola ovat näkyvissä selväkielisenä salasanatiedostossa? (maksimipistemäärä 3)

Kysymys 5. Vertaile pääsylistoja ja valtuuslistoja keskenään. Käytä vertailujen pohjana tilankäyttöä ja hakujen tehokkuutta. Esimerkkejä hauista: kenellä on lukuoikeus tiedostoon tai mihin objekteihin jollakin subjektilla on jokin oikeus. (maksimipistemäärä 3)

Kysymys 6.

a) Anna esimerkki sääntöpohjaista ja yksilöpohjaista mallia noudattavasta järjestelmästä tai sovelluksesta tietokoneympäristössä. (maksimipistemäärä 3)
b) Anna vastaavat esimerkit tietokoneympäristöön kuulumattomista organisaatioista tai järjestelmistä. (maksimipistemäärä 3)

Kysymys 7.

a) Materiaalin lopussa puhuttiin hakemistoista ja hakemistopoluista. Määrittele nämä käsitteet. (maksimipistemäärä 2)
b) Windowsissa oikeudet ovat kattavampia ja monimutkaisempia kuin Linuxissa. Pohdi lyhyesti tämän ratkaisun hyviä ja huonoja puolia. (maksimipistemäärä 3)