Lappeenrannan teknillinen yliopisto
School of Business and Management
Tietotekniikan koulutusohjelma

Kandidaatintyö

Ilkka Virta

KERTAKÄYTTÖSALASANAT KÄYTTÄJÄNTUNNISTAMISEN TUKENA

Työn tarkastaja: Tutkijatohtori Ari Happonen

Työn ohjaaja: Tutkijatohtori Ari Happonen

TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto
School of Business and Management
Tietotekniikan koulutusohjelma

Ilkka Virta

Kertakäyttösalasanat käyttäjäntunnistamisen tukena

Kandidaatintyö

28 sivua, 3 kuvaa, 2 taulukkoa, 1 liite

Työn tarkastaja: Tutkijatohtori Ari Happonen

Hakusanat: tietoturva, käyttäjäntunnistaminen, kertakäyttösalasanat, Linux PAM, SSH
Keywords: computer security, user authentication, one-time passwords, Linux PAM, SSH

Yleisin tapa käyttäjäntunnistamiseen on perinteinen kiinteä salasana, mutta kertakäyttöiset salasanat ovat lisänneet suosiotaan myös suurelle yleisölle suunnatuissa palveluissa.
Tässä työssä käsitellään käyttäjäntunnistamiseen ja kirjautumistapahtumaan kohdistuvia hyökkäyksiä ja vastatoimia niille. Työ toteaa että kertakäyttösalasanat toimivat hyvin passiiviseen vakoiluun rajoittunutta hyökkääjää vastaan, mutta eivät korvaa puutteellista ohjelmistoturvallisuutta tai palvelinjärjestelmän luotettavaa tunnistamista aktiivista hyökkääjää vastaan.
Työssä rakennetaan myös Linux-alustalla toimiva kertakäyttösalasanakirjautumisjärjestelmä olemassaoleviin ohjelmistomoduuleihin perustuen. Tähän käytetään oath-toolkit -ohjelmiston tunnistusmoduulia sekä RADIUS-protokollaa tunnistamispalvelimen ja sitä käyttävien järjestelmien väliseen kommunikointiin.

ABSTRACT

Lappeenranta University of Technology
School of Business and Management
Degree Program in Computer Science

Ilkka Virta

One-time Passwords in User Authentication

Bachelor's Thesis

28 pages, 3 figures, 2 tables, 1 appendix

Examiner: Post Doctoral Researcher Ari Happonen

Keywords: computer security, user authentication, one-time passwords, Linux PAM, SSH

The most common method for user identification is a traditional fixed password, even though they have known issues. However, one-time passwords have lately become more prevalent, even in services aimed at ordinary users.

This work considers threats related to user authentication and the authentication process, and methods for mitigating different types of threats. The work shows that one-time passwords offer good protection against a passive eavesdropper but cannot replace inadequate software security or reliable identification of the remote system in case of an active attacker.

The work also compiles an authentication system for Linux servers using pre-existing software modules. The authentication module from the oath-toolkit library is used, as well as the RADIUS protocol for communication between the authentication server and its clients.

Sisällys

1 Johdanto

1.1 Työn tausta ja motivaatio

Selvästi yleisin tapa käyttäjien tunnistamiseksi tietojärjestelmissä ja verkkopalveluissa on perinteinen kiinteä salasana, joka vaihtuu vain harvoin. Tavallisesti käyttäjä voi vaihtaa salasanaa halutessaan ja joissakin järjestelmissä vaaditaan säännöllistä vaihtamista, vaikka nykytiedon valossa tästä ei ole juuri hyötyä eikä sitä suositella [33] [11]. Käyttömukavuuden vuoksi vaihtoväli lasketaan tällöinkin yleensä kuukausissa. Esimerkiksi Lappeenrannan teknillisen yliopiston järjestelmissä vaaditaan salasanan vaihtamista 3 kk:n välein [9]. Oleellista on, että samaa salasanaa käytetään kirjautumiseen useita kymmeniä kertoja.

Menetelmän yleisyydestä huolimatta kiinteillä salasanoilla on kuitenkin useita heikkouksia. Salasanan pitäisi olla riittävän satunnainen jotta se olisi vaikea arvata, mutta mitä satunnaisempi salasana on, sitä vaikeampi se on muistaa [30]. Myöskin salasanan vaihtaminen useasti vaikeuttaa sen muistamista. Toisaalta mikä tahansa salasana voidaan vakoilla sitä syötettäessä. Salasanan joutumista vääriin käsiin on lähes mahdotonta havaita, koska kopioitu salasana ei poikkea mitenkään alkuperäisestä eikä käyttäjältä katoa mitään. Mekaanisen avaimen kadotessa tilanne on toinen: avaimen omistaja voi huomata kadottaneensa jotakin, viimeistään siinä vaiheessa kun yrittää käyttää puuttuvaa avaintaan.

Kiinteisiin salasanoihin liittyvien riskien vuoksi yritysten sisäisissä järjestelmissä on melko tavallista käyttää toisistaan riippumattomia kertakäyttöisiä salasanoja. Samoin pankit ovat käyttäneet vaihtuvia tunnuslukuja kuluttajille suunnatuissa verkkopalveluissaan jo pitkään, koska suora pääsy rahan käsittelyyn on houkuttelevaa rikollisille. Myös kuluttajille suunnatut tavalliset verkkopalvelut, esimerkiksi LastPass, Google ja Facebook, tarjoavat joka kirjautumisella vaihtuvia kertakäyttöisiä salasanoja kiinteän salasanan vaihtoehdoksi tai sellaisen kanssa käytettäväksi [8] [5] [1].

Koska kertakäyttösalasana ei toimi enää toisella kirjautumisyrityksellä, eikä seuraava salasana riipu edellisestä, voidaan olettaa ettei käytetyn salasanan joutuminen vääriin käsiin ole ongelma. Näin olleen mahdollinen vakoilusta syntyvä haitta poistuu tai vähintään pienenee huomattavasti.

1.2 Työn tavoitteet ja rajaukset

Työn tavoitteena on tutkia kertakäyttösalasanajärjestelmien hyötyä käyttäjäntunnistamisessa tarkastelemalla millaisten uhkien ja hyökkäysten torjuntaan kertakäyttöiset salasanat sopivat. Tarkastellaan myös kertakäyttösalasanoihin kohdistuvia hyökkäyksiä ja riskejä.

Työn käytännöllisenä osuutena toteutetaan Linux-alustalle useamman laitteen ja käyttäjän autentikointipalvelimeksi soveltuva kertakäyttöisiä salasanoja hyödyntävä järjestelmä. Toteutuksen taustaksi tarkastallaan hieman erinäisiä olemassa olevia kertakäyttösalasana-algoritmeja ja -ohjelmistoja sekä eri toteutusten etuja ja heikkouksia. Työn painopiste on järjestelmissä, joiden lähdekoodi tai määritys on avoimesti saatavilla, mutta eräitä kaupallisia toteutuksia sivutaan niiden

laajan tunnettuuden vuoksi.

1.3 Työn rakenne

Tämän johdantoluvun lisäksi työssä on viisi lukua. Luku 2 käsittelee käyttäjäntunnistamista ja sen menetelmiä, mukaan lukien kertakäyttösalasanoja. Luvussa 3 käsitellään etäyhteyden kirjautumistapahtumaan kohdistuvia hyökkäyksiä, sekä pyritään selvittämään mitä uhkia vastaan kertakäyttösalasanajärjestelmä voi olla hyödyllinen. Samassa yhteydessä käsitellään myös erityisesti kertakäyttösalasanajärjestelmiin kohdistuvia hyökkäyksiä. Luvussa 4 tarkastellaan erinäisiä kertakäyttösalasanajärjestelmiä ja -ohjelmistoja, ja luvussa 5 toteutetaan varsinainen käytännön osuus. Luku 6 sisältää yhteenvedon ja johtopäätökset.

2 Käyttäjäntunnistamisen menetelmät ja ongelmat

Käyttäjän tunnistamisella (engl. *authentication*) tarkoitetaan sen selvittämistä, onko järjestelmään yhteyttä ottava käyttäjä se henkilö, joka hän väittää olevansa. Tuntemattoman käyttäjän identifiointi jätetään tässä huomiotta, ja oletetaan että käyttäjä antaa kirjautumistapahtuman yhteydessä käyttäjätunnuksen tai vastaavan tunnisteen, jolloin riittää selvittää onko esitetty identiteetti oikea.

2.1 Tunnistamismenetelmien eri tyypit

Kirjallisuudessa jaetaan käyttäjän tunnistamiseen käytetyt menetelmät yleisesti kolmeen ryhmään: käyttäjä voidaan tunnistaa sen perusteella

1) mitä hän tietää tai muistaa: salasanat, tunnusluvut
2) mitä hänellä on hallussaan: tavalliset avaimet, avainlukulistat, pankkikortit
3) mitä hän tekee tai on, eli biometriset keinot: sormenjäljet, ääni, silmät.

[16] [25] [20]

Käytännön elämässä ihmiset tunnistavat toisensa tavallisesti biometrisesti, ulkonäön tai äänen perusteella (kohta 3 yllä), mutta myös sosiaalisessa kanssakäymisessä aiemman tiedon perusteella (1). Virallisessa yhteydessä henkilöllisyys todistetaan kuvallisella henkilökortilla (2 ja 3). Mekaaniset lukot toimivat tavallisimmin mukana kuljetettavalla avaimella (2), mahdollisesti muistettavalla numerokoodilla (1), tai jopa sormenjälki- tai silmätunnistuksella (3). Kaikilla kolmella menetelmillä on omat hyvät ja huonot puolensa, jotka on esitetty taulukossa 1. Erityisesti biometristen tunnisteiden tarkkuus ja luotettavuus riippuvat suuresti käytetystä tunnisteesta ja tekniikasta.

2.2 Useamman menetelmän yhtäaikainen käyttö

Koska kaikilla tunnistamiskeinoilla on omat rakenteelliset heikkoutensa, käytetään paremman turvatason saavuttamiseksi usein yhdistelmää eri ryhmiin kuuluvista menetelmistä. Tällöin yksittäisen menetelmän heikkoutta voidaan paikata toisen menetelmän vahvuuksilla. [16]

Länsimaisesta käytännön elämästä tuttu esimerkki on tunnusluvulla suojattu pankkikortti. Mikäli kortti hukkuu tai varastetaan, ei sen löytäjä voi tehdä maksuja, sillä vain oikea omistaja tuntee oikean tunnusluvun. Toisaalta pelkän tunnusluvun selvittäminen ei myöskään auta, sillä maksun tekemiseen vaaditaan myös itse fyysinen kortti.

(1) Muistiin perustuvat salasanat ja tunnussanat:	
Edut	**Haitat**
• mieleen painettu salasana ei yleensä unohdu yhtäkkiä • järjestelmä on yleensä helppo toteuttaa	• uusi salasana voi olla vaikea muistaa • salasana voidaan salakuunnella sitä syötettäessä tai annettaessa • salasana voidaan monistaa kertomalla se toiselle, mikä laskee kynnystä luovuttaa se vapaaehtoisesti esim. • salasanan joutumista vääriin käsiin ei voi suoraan havaita
(2) Mukana kannettava (fyysinen) avain:	
Edut	**Haitat**
• käyttäjät eivät yleensä luovuta avaimiaan vapaaehtoisesti vieraille • mekaanisten laitteiden toiminta yleensä hyvin tunnettu • fyysisen esineen katoaminen yleensä huomataan	• voi unohtua esim. kotiin • voidaan kopioida jos avaimeen päästään käsiksi • voidaan varastaa • mekaanisten lukkojen uudelleen-sarjoitus on hankalaa
(3) Biometriset tunnisteet:	
Edut	**Haitat**
• parhaimmillaan vaikeita kopioida huomaamattomasti • kulkevat automaattisesti käyttäjän mukana	• lukulaitteet kalliita ja hankalia • kehittyvä teknologia helpottaa huomaamatonta kopiointia • erityisesti sormenjäljet leviävät tahattomasti ympäriinsä • vertailussa aina virhemahdollisuus • tunnisteita ei käytännössä voida vaihtaa • digitaaliseen muotoon siirretty tunniste voidaan kopioida kuin salasana

Taulukko 1: Tunnistamiskeinojen edut ja heikkoudet

2.3 Kertakäyttösalasanat

Yksinkertainen kehitys kiinteästä salasanasta eteenpäin on järjestelmä jossa jokaista salasanaa käytetään vain kerran, ja salasanat ovat toisistaan riippumattomia. Tälläinen järjestelmä on edelleen turvallinen vaikka käytetty salasana joutuisi kirjautumisen jälkeen kolmannen osapuolen käsiin. [34]

Kertakäyttöiset salasanat voidaan jakaa edelleen kahteen eri ryhmään: järjestelmät, joissa odotettu salasana tiedetään ennen kirjautumistapahtumaa, sekä haaste-vaste -järjestelmät joissa odotettu salasana (vaste) perustuu palvelimen kirjautumisen yhteydessä satunnaisesti muodostamaan haasteeseen.

Haasteettomassa järjestelmässä salasanat voidaan muodostaa toisistaan riippumattomina ja listata, tai kehittää laskennallisesti jostakin muuttuvasta arvosta, kuten jokaisen kirjautumisen yhteydessä tai ajan myötä kasvavasta laskurista. Joka tapauksessa molemmilla osapuolilla on oltava keino

selvittää tarvittu salasana, ja järjestelmän perustana olevan algoritmin on oltava sikäli turvallinen, että aiemmin käytetyistä salasanoista ei voida johtaa seuraavia salasanoja.

Salasanan vaihtuessa joka käyttökerralla tarvitaan luonnollisesti huomattavasti suurempi määrä yksittäisiä salasanoja kuin pelkkää kiinteää salasanaa käytettäessä. Monia kymmeniä salasanoja ei ole käytännössä mahdollista muistaa, joten käyttäjän on listattava salasanat esim. paperille, tai käytettävä jotakin laitetta joka luo salasanat laskennallisesti sitä mukaa kun niitä tarvitaan. Siten kyse on käytännössä aina jonkin esineen hallintaan perustuvasta tunnistamisesta, eikä muistamiseen liittyvästä kuten kiinteän salasanan tapauksessa. Kertakäyttöisiä ja kiinteitä salasanoja käytetäänkin usein yhdessä, jotta saadaan kahden tunnistamiskeinon hyöty.

3 Kirjautumiseen ja etäyhteyksiin kohdistuvia uhkia

Tässä luvussa käsitellään perinteiseen, kiinteää salasanaa käyttävään kirjautumiseen kohdistuvia hyökkäyksiä sekä keinoja niiden vastustamiseksi. Erityisesti tarkastellaan pätevätkö samat hyökkäykset kertakäyttösalasanoja käyttävää kirjautumista vastaan, ja tämän jälkeen erityisesti kertakäyttösalasanajärjestelmiin kohdistuvia hyökkäyksiä.

3.1 Hyökkäyksiä kiinteällä salasanalla kirjautumista vastaan

Seuraavassa käydään läpi yleisimpiä kiinteän salasanan järjestelmiin kohdistuvia uhkia. Näitä ovat salasanan arvaaminen, salasanatietokannan vuotaminen vääriin käsiin, salasanan vakoilu sitä syötettäessä tai siirrettäessä, sekä käyttäjän suostuttelu antamaan salasanansa väärälle taholle ja käyttäjätunnuksen lukkiutuminen epäonnistuneiden kirjautumisyritysten vuoksi.

a. Salasanan arvailu

Yksinkertaisin (ja naiivein) tapa yrittää kirjautua oikeudettomasti salasanalla suojattuun järjestelmään on ottaa yhteyttä siihen, ja yrittää arvata oikea salasana. Automatisoituna yrityksiä voidaan tehdä varsin nopeasti, mutta tunkeutujan kannalta ongelmana on, että järjestelmä voi rajoittaa hyökkäyksen tehokkuutta rajaamalla esimerkiksi kirjautumisyritysten määrää per aikayksikkö tai sulkemalla koko tunnuksen tai tunkeutujan verkko-osoitteen kokonaan riittävän monen epäonnistuneen kirjautumisen jälkeen. [34] [16]

b. Salasanatietokannan joutuminen vääriin käsiin

Mikäli hyökkääjä saa palvelimelle tallennetut salasanat haltuunsa esim. palvelimen ohjelmistovian vuoksi, hän voi käyttää niitä välittömästi oikeutettujen käyttäjien nimissä kirjautumiseen. Tämän vuoksi on tapana tallentaa vain salasanasta yksisuuntaisen funktion kautta muodostettu tiiviste. Kirjautumistilanteessa muodostetaan käyttäjän esittämää salasanaa vastaava tiiviste, ja verrataan sitä tallennettuun tiivisteeseen. Tiivisteestä ei voida johtaa alkuperäistä salasanaa, vaan ainoa keino löytää tiivistettä vastaava salasana on käydä läpi mahdollisia salasanoja, ja muodostaa niitä vastaavat tiivisteet. Mikäli näin saatu tiiviste vastaa palvelimelle tallennettua tiivistettä, on löydetty sitä vastaava salasana.

Muokkaamalla tiivistefunktiota laskennallisesti raskaammaksi ja vaatimalla salasanoilta riittävää pituutta ja monimuotoisuutta, voidaan kaikkien mahdollisten salasanojen läpikäynti tehdä niin raskaaksi että se ei onnistu kohtuullisessa ajassa, vaan vaatisi hyökkääjän odotettavissa olevalla laskentakapasiteetilla esim. kymmeniä vuosia. [34]

Palvelimen ohjelmistovikoihin liittyvä suurempi ongelma on se, että vika voi antaa hyökkääjälle myös muita mahdollisuuksia kuin pelkän tietojen lataamisen. Vika, joka sallii mielivaltaisen ohjelman ajamisen palvelimella mahdollistaa myös palvelimen käyttäjätietokannan tai ohjelmiston muokkaamisen. Hyökkääjä voi tällöin käyttää järjestelmää oman ohjelmistonsa kautta; lisätä itselleen ylimääräisen käyttäjätunnuksen; tai muokata varsinaista tunnistusjärjestelmää siten että

pääsynhallinta ohitetaan hyökkääjän niin halutessa. Ohjelmistovikojen kautta tapahtuva murtautuminen on siten merkittävä riski kirjautumismenetelmästä riippumatta.

c. Passiivinen vakoilu salasanaa syötettäessä

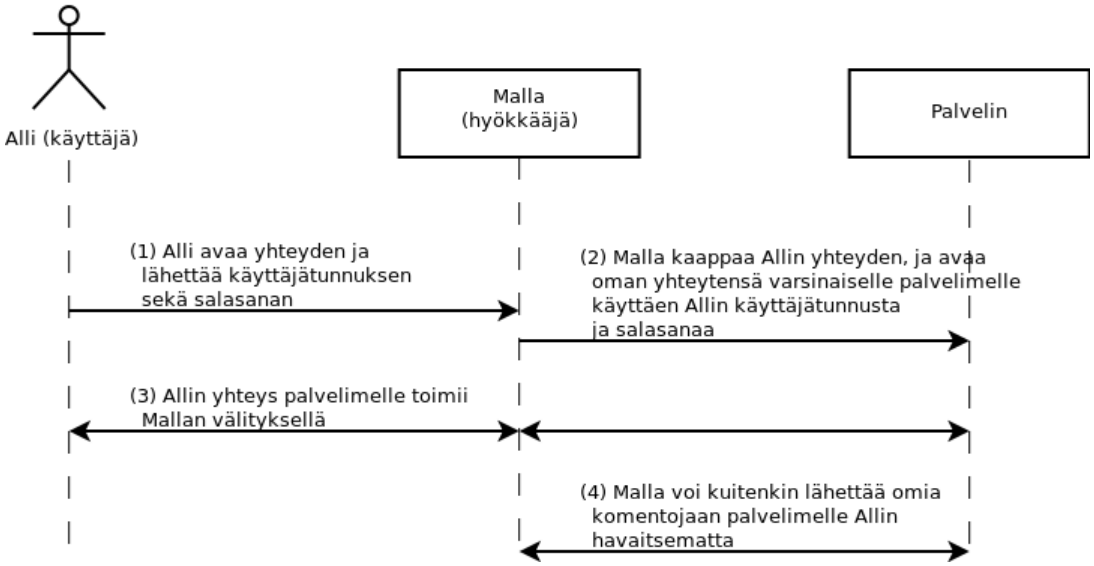
Suoraviivainen tapa selvittää käyttäjän salasana on yksinkertaisesti katsoa kun sitä syötetään. Riippumatta kirjautumisen logiikasta salasana siirtyy selväkielisenä näppäimistöltä käyttäjän laitteelle ja se voidaan havaita tällä välillä esim. kuvaamalla käyttäjän näppäimenpainalluksia tai tallentamalla näppäimistön lähettämä signaali sähköisesti. Näin saatu salasana voidaan tallentaa myöhemmin noudettavaksi tai siirtää langattomasti heti eteenpäin.

Tälläistä passiivista vakoilua vastaan kertakäyttöisiin salasanoihin perustuva järjestelmä antaa hyvän turvan, koska hyökkääjän tallentama salasana on seuraavalla kirjautumiskerralla hyödytön. Passiivinen vakoilu onkin usein ensimmäisenä mainittu uhka, jota vastaan eri kertakäyttö-salasanajärjestelmät on tarkoitettu, esim. [24] [19]. Näppäimenpainalluksia voidaan tallentaa myös ohjelmallisesti. Kuten palvelimeen kohdistuvan tietomurron tapauksessa, käyttäjän ohjelmiston muokkaaminen antaa passiivisen tallentamisen lisäksi myös muita mahdollisuuksia (kts. kohta *e* alla).

d. Tiedonsiirtoväylän kaappaaminen (man-in-the-middle -hyökkäys)

Man-in-the-middle -hyökkäyksellä tarkoitetaan tilannetta jossa hyökkääjä ohjaa käyttäjän tietoliikenteen kulkemaan hyökkääjän hallitseman laitteen kautta ja edelleen alkuperäiseen kohteeseen. Hyökkääjän välityspalvelin pääsee tällöin sekä lukemaan että käsittelemään kaikkea yhteyden yli siirrettävää dataa. Välityspalvelin voi esittää käyttäjän laitteelle alkuperäistä kohdepalvelinta ja päin vastoin, ja solmia salatun yhteyden molempiin suuntiin käyttäjän tai etäjärjestelmän havaitsematta mitään muutosta oletettuun tilanteeseen. Käytetyllä tunnistamis-menetelmällä ei tällöin ole merkitystä, koska välityspalvelin voi odottaa kunnes käyttäjä on tunnistautunut, ja muokata siirrettyä dataa vasta tunnistautumisen jälkeen. Kuva 1 esittää periaatemallin välityspalvelinhyökkäyksestä.

Välityspalvelinhyökkäykseltä voidaan suojautua sitomalla salattu yhteys johonkin palvelimen aiemmin tunnettuun tunnisteeseen, eli käytännössä esimerkiksi TLS-sertifikaattiin tai SSH-avaimeen.



Kuva 1: Kaaviokuva välityspalvelinhyökkäyksestä

e. Käyttäjän ohjelmiston muokkaaminen (troijan hevonen)

Mikäli hyökkääjä voi asentaa käyttäjän laitteelle ohjelmiston esimerkiksi syötettyjen salasanojen tallentamista varten, on todennäköistä että myös käyttäjän muiden ohjelmien muokkaaminen on samalla mahdollista.

Käyttäjän asiakasohjelmisto käsittelee välttämättä sekä salattua etäyhteyttä, että käyttäjän syöttämää selväkielistä dataa, jolloin muokattu ohjelma voi helposti tallentaa käyttäjän tietoliikenteen, mutta myös muokata siirrettävää dataa, tai välittää sen hyökkääjän hallitsemalle välityspalvelimelle kuten yllä. Tässäkään tapauksessa käytetyllä kirjautumismenetelmällä ei ole väliä hyökkäyksen onnistumisen kannalta, vaan käyttäjän tulee varmistua oman ohjelmistonsa turvallisuudesta.

Käytännössä esim. WWW-sovelluksissa käyttäjän istuntoon kuuluvat HTTP-kyselyt tunnistetaan kirjautumisen jälkeen istuntotunnisteella (engl. *session cookie*), jonka palvelin lähettää selaimelle kirjautumisen onnistuttua. Muokattu sovellus käyttäjän laitteella voi lähettää samaa istunto-tunnistetta käyttäen HTTP-pyyntöjä palvelimelle. SSH-yhteyden tapauksessa kaikki komennot siirretään saman TCP-yhteyden yli, mutta muokattu SSH-asiakas voi silti muuttaa palvelimelle lähetettäviä komentoja.

f. Salasanan kalastelu

Salasanojen kalastelulla (engl. *phishing*) tarkoitetaan toimintaa jossa hyökkääjä lähestyy oikeutettua käyttäjää esim. sähköpostitse esittäen järjestelmän ylläpitoa tai muuta sopivaa auktoriteettia, ja pyrkii erehdyttämään käyttäjän luovuttamaan kirjautumistietonsa. Käyttäjää voidaan pelotella tunnuksen sulkemisella, mikäli tunnusta ei ”vahvisteta” syöttämällä kirjautumis-tiedot palveluun, joka todellisuudessa on hyökkääjän hallinnassa, ja tallentaa tiedot myöhempää

käyttöä varten. [26]

Teoriassa yksinkertaisin ratkaisu kalasteluhyökkäyksiä vastaan on käyttäjien kouluttaminen olemaan syöttämättä tunnistautumistietojaan tuntemattomiin palveluihin. Käytännössä tällöin vaaditaan myös käyttäjän käsittelemän järjestelmän varmaa tunnistamista, jotta oikea järjestelmä voidaan erottaa yhtäläisen näköisestä kopiosta.

Palvelimen esittämään joka kerralla vaihtuvaan haasteeseen perustuva järjestelmä käytännössä estää kalastelun, ja toisaalta riittävän lyhyen ajan sisällä vanhentuva salasana tekee onnistuneesta kalastelusta vaikeampaa. Sen sijaan salasanat jotka ovat voimassa pidempään, esim. seuraavaan kirjautumiseen asti, eivät anna kovin suurta suojaa kalastelulta mikäli käyttäjä luovuttaa yhden tai useamman käyttämättömän salasanan. Käytännössä tälläisiä kalasteluja tiedetään toteutetun esim. useita suomalaisia verkkopankkeja vastaan [2] [22] [29].

g. Palvelunesto tunnuksen lukitsemisen kautta

Salasanan arvailun hillitseminen rajoittamalla kirjautumisyritysten määrää sisältää sen haitta-puolen, että hyökkääjä voi tahallaan tai tahattomasti estää myös oikeutetun käyttäjän kirjautumisen järjestelmään. [16]

Tämän estämiseksi järjestelmän täytyisi voida jollakin tapaa erottaa useampi kirjautumisyrityksiä tekevä taho toisistaan, ja lukita niistä vain yksi kerrallaan. Erottelu voitaisiin tehdä esim. käyttäjän IP-osoitteen perusteella, mutta tämäkin antaa vain rajallisen hyödyn hajautettua hyökkäystä vastaan.

Kertakäyttösalasanajärjestelmä yhdessä kiinteän salasanan kanssa käytettynä voi lievittää ongelmaa mikäli kirjautumisyrityksiä rajoitetaan vain silloin kun kirjautuja antaa oikean kertakäyttö-salasanan. Käytännössä tällöin kirjautumisten rajoittaminen suojaa sellaista hyökkääjää vastaan, joka on saanut kertakäyttösalasanat käsiinsä, ja pyrkii arvaamaan kiinteän salasanan. [16]

3.2 Uhkien lieventäminen ja kertakäyttösalasanat

Yllä esitetyistä hyökkäyksistä lähinnä kirjoitetun salasanan vakoilu kohdistuu nimenomaan varsinaiseen tunnistetietoon. Tätä passiivista vakoilua vastaan kirjautuminen kertakäyttöisillä salasanoilla toimiikin hyvin, koska vakoilemalla saatu tunniste on käyttökelvoton seuraavalla kerralla. Toisaalta myös suora ohjelmallinen tunnistautuminen (esim. SSH-avaimella) tai älykorttiin perustuva tunnistautuminen ovat melko suojattuja passiivista vakoilua vastaan, mikäli hyökkääjä ei pääse käsiksi itse avaimeen tai älykorttiin. Älykortin tapauksessa kuitenkin kortti fyysisenä esineenä on varastettavissa.

Sen sijaan ohjelmallisia hyökkäyksiä vastaan itse kirjautumismenetelmällä ei ole suurta merkitystä, koska hyökkääjän hallitsema ohjelmisto joko käyttäjän koneella tai palvelimella voi melko helposti ohittaa varsinaisen kirjautumisen kokonaan. Muuttuva tunniste auttaa lähinnä yksinkertaista

passiivista vakoiluohjelmaa vastaan.

Tunnusten kalastelua kertakäyttösalasanajärjestelmä voi toteutuksesta riippuen hankaloittaa, sillä kalastelijan on otettava tunnisteen muuttuminen huomioon, ja hyödynnettävä saatuja tunnuksia niiden voimassaolon aikana (mikäli se on ylipäänsä mahdollista). Käyttäjien kouluttamista olemaan antamatta salassa pidettävää tietoa kolmansille osapuolille on kuitenkin edelleen pidettävä oleellisimpana torjuntakeinona.

Etäpalvelimen tunnistamiseen liittyviin ongelmiin ja man-in-the-middle -hyökkäyksiin, samoin kuin palvelimelta tapahtuvaan tietojen vuotamiseen ei kirjautumismenetelmä voi varsinaisesti vaikuttaa lainkaan.

Riittävän hyviä salasanoja käytettäessä verkon yli tapahtuva salasanojen arvailu on estettävissä kirjautumisyrityksien määrää rajoittamalla sekä kiinteillä, että vaihtuvilla salasanoilla. Kirjautumis-kertojen rajoittamiseen liittyvää palvelunestovaikutusta ei tässä käsitellä sen enempää.

Taulukko 2 esittää koosteen eri uhkista, niihin soveltuvista vastakeinoista, ja muuttuvan tunnisteen kuten kertakäyttösalasanan soveltumisesta vastakeinoksi.

Uhka	**Soveltuvat vastakeinot**	**Muuttuvan tunnisteen hyöty**
a. Salasanan arvailu	Kirjautumisyritysten määrän rajoittaminen aikayksikköä ja lähdettä kohden	ei hyötyä
b. Salasanatietokannan joutuminen vääriin käsiin	Palvelimen yleinen ohjelmistoturvallisuus; salasanojen tallentaminen turvallisesti	ei hyötyä
c. Salasanan passiivinen vakoilu sitä syötettäessä	Käyttäjien kouluttaminen, kertakäyttösalasanat	hyvä
d. Man-in-the-middle -hyökkäys	Palvelimen kryptografinen tunnistaminen sertifikaateilla (TLS), palvelinavaimella (SSH) tai vastaavalla	ei hyötyä
e. Ohjelmallinen hyökkäys / troijalainen	Käyttäjän laitteen yleinen ohjelmistoturvallisuus.	rajallinen
f. Salasanan kalastelu	Käyttäjien kouluttaminen; kertakäyttösalasanat (haaste-vaste)	rajallinen
g. Palvelunesto / tunnuksen lukitseminen	Verkko-osoitekohtainen erottelu kirjautumisyritysten rajoittamisessa.	rajallinen

Taulukko 2: Kirjautumiseen liittyvien uhkien vastakeinoja

3.3 Hyökkäyksiä kertakäyttösalasanoja vastaan

Kertakäyttösalasanajärjestelmiä vastaan voidaan kohdistaa joitakin hyökkäyksiä, jotka ottavat huomioon erityisesti vaihtuvan tunnisteen ominaisuudet. Tärkeimpinä näistä käsitellään algoritmiin

perustuvat hyökkäykset sekä samanaikainen kirjautuminen.

a. Algoritmiset heikkoudet

Muuttuvan salasanan hyöty perustuu siihen, että tulevia salasanoja ei voida päätellä aiemmin käytetyistä salasanoista. Salasanoja tuottavan algoritmin tulee siis olla kryptografisesti turvallinen, ja perustua riittävän suuren siemenlukuun, jotta sitä ei voida laskennallisesti murtaa.

Käytännössä algoritmiset puutteet eivät ole ongelma yleisessä käytössä olevilla kertakäyttö-salasanajärjestelmillä jotka perustuvat tunnettuihin tiiviste- tai salausalgoritmeihin.

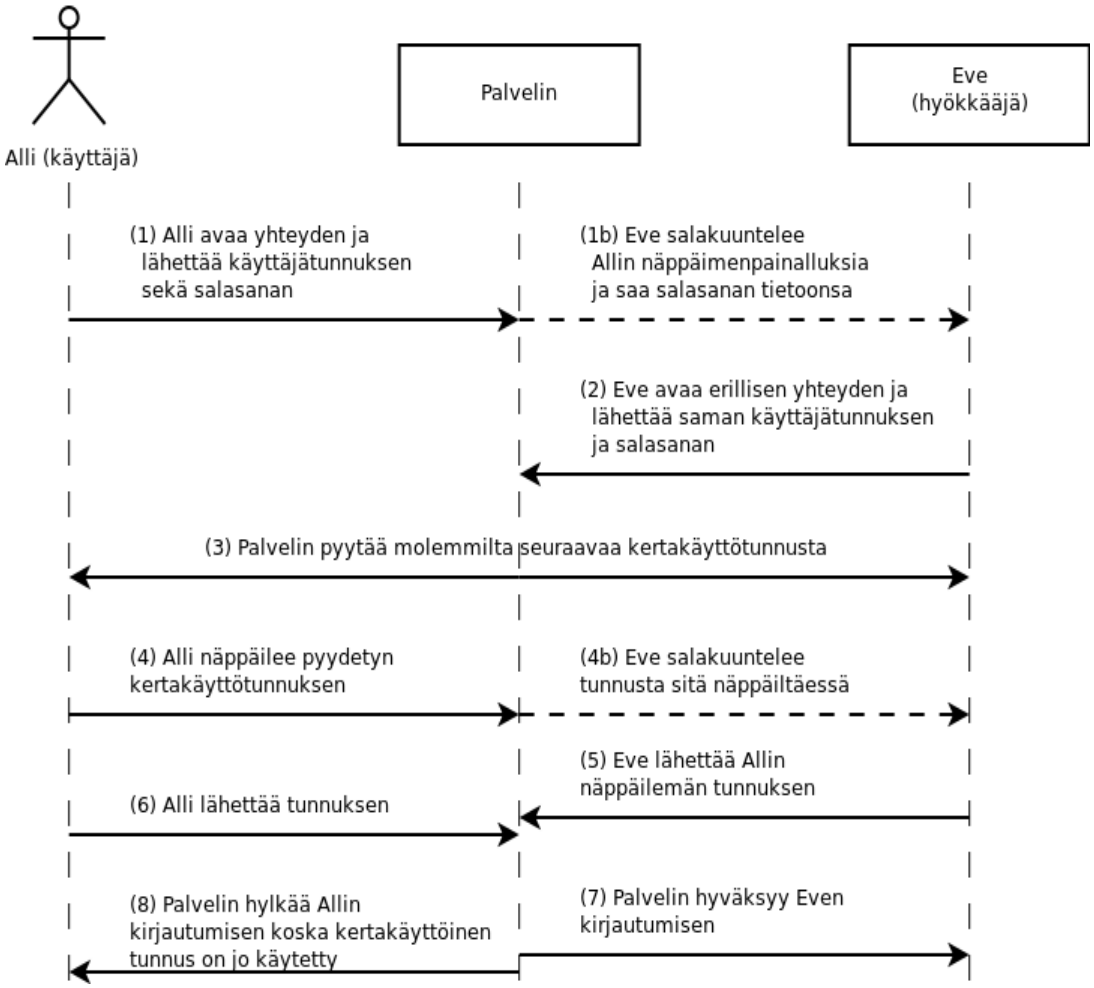
b. Yhtäaikainen kirjautuminen

Kiinteää salasanaa vakoiltaessa hyökkääjä voi yleensä käyttää saamiaan tunnistetietoja kirjautumiseen välittömästi tai vasta myöhemmin. Kertakäyttöisen salasanan kohdalla tilanne on toinen: mikäli käyttäjän syöttämä salasana päätyy hyökkääjälle, voi hyökkääjä käyttää kyseistä salasanaa kirjautumiseen vain siihen saakka kunnes se on merkitty palvelimella käytetyksi.

Käytännössä hyökkäys toteutetaan esim. siten, että tarkkaillaan käyttäjän näppäimenpainalluksia, ja pyritään automaattisesti lähettämään kirjautumispyyntö juuri ennen kuin oikean käyttäjän kirjautumispyyntö lähtee. Kyse on siis nopeuskilpailusta, jossa hyökkääjällä on etu, koska tietokoneohjelma voi suurella todennäköisyydellä lähettää pyynnön nopeammin kuin ihmiskäyttäjä. Hyökkääjä voi kasvattaa etuaan onnistumistodennäköisyyden kustannuksella arvaamalla salasanan viimeisen merkin jo ennen kuin varsinainen käyttäjä syöttää sen. [18] [19] Kuvassa 2 esitetään hyökkäykseen liittyvä viestien vaihto käyttäjän, palvelimen ja hyökkääjän välillä.

Onnistuessaan hyökkäys johtaa käyttäjän yrittämän kirjautumisen epäonnistumiseen, koska nyt tämän syöttämä salasana on vanhentunut. Käyttäjä saattaa kuitenkin olettaa kirjoittaneensa tunnisteen väärin, tai että kyseessä on jokin muu viaton ohimenevä vika. Aktiivinen hyökkääjä voisi myös katkaista käyttäjän verkkoyhteyden sopivalla hetkellä, jolloin myös epäonnistunut kirjautuminen saatettaisiin tulkita sattumanvaraisen verkkokatkon syyksi.

Haaste-vaste -menetelmät ovat immuuneja yhtäaikaiselle kirjautumisyritykselle, koska erilliset kirjautumistapahtumat saavat eri haasteen, eikä sama vastaus käy molempiin.



Kuva 2: Toimintakaavio yhtäaikaiseen kirjautumiseen perustuvasta hyökkäyksestä

3.4 Yhteenveto

Kertakäyttösalasanoja voidaan käyttää melko helposti parantamaan kirjautumistapahtuman turvallisuutta sellaista passiivista vakoilua vastaan, jossa hyökkääjä käyttää saamiaan tietoja vasta myöhemmin. Reaaliaikaista hyökkäystä vastaan vaaditaan kuitenkin haaste-vaste -menetelmään perustuvaa kirjautumista tai muita estokeinoja. Kirjautumismenetelmä ei myöskään vaikuta ohjelmistoturvallisuuteen tai etäjärjestelmän tunnistamiseen liittyviin ongelmiin.

4 Kertakäyttösalasanajärjestelmiä

Tässä luvussa esitellään eräitä olemassaolevia kertakäyttösalasanajärjestelmiä ja -algoritmeja, sekä niiden vahvuuksia ja heikkouksia.

4.1 S/Key / OPIE

S/Key lienee vanhin tiedossa oleva kertakäyttösalasanajärjestelmä. Se perustuu jo 1981 julkaistuun Lamportin menetelmään, jossa salasanat muodostetaan toisistaan kryptografisen tiivistefunktion avulla. Menetelmän turvallisuus perustuu siihen, että tiivistefunktion käänteisfunktiota on vaikea laskea. [24] [7] S/Keyn uudempi versio tunnetaan nimellä OPIE ja se on edelleen tuettu ainakin FreeBSD -käyttöjärjestelmässä [3] [17].

S/Key on suunniteltu aikana, jolloin etäyhteydet tavallisesti eivät olleet salattuja, ja sen rakenne sisältää tämän kannalta aikanaan hyödyllisiä ominaisuuksia. Ketjurakenteen vuoksi palvelin ei tunne seuraavaa hyväksyttävää salasanaa ennen kuin saa sen, mikä mahdollistaa salasanaketjun vaihtamisen välittämättä salaista tietoa verkon yli. Haittapuolena palvelimen ja käyttäjän välillä ei ole jaettua salaisuutta, mikä yhdessä lyhyiden salasanojen kanssa tekee S/Keyn nykyisestä turvallisuudesta kyseenalaisen. Järjestelmässä käytetyn 64-bittisen tiivistefunktion murtamista kaikki vaihtoehdot läpikäymällä ei voida pitää nykylaitteiden laskentakapasiteetilla mahdottomana, ja tiivistefunktion murtaminen kerran mahdollistaa kaikki S/Key:ta käyttävät tietokonejärjestelmät triviaalille hyökkäykselle.

4.2 RSA SecurID

RSA Securityn (nykyisin EMC:n tytäryhtiö) kaupallinen SecurID [23] perustuu AES-algoritmiin, ja kuten TOTP (alla), tuottaa symmetrisen avaimen avulla kellonajan perusteella muuttuvia numerokoodeja. SecurID:n käyttämää algoritmia ei ole virallisesti julkistettu, mutta se on mallinnettu ohjelmistototeutusten perusteella, ja ainakin yksi kolmannen osapuolen tekemä kloonitoteutus on olemassa [27]. RSA ei kuitenkaan virallisesti tue klooneja. Koska kyseessä on kaupallinen ja salainen algoritmi, ei sitä käsitellä tässä enempää.

4.3 OATH: HOTP ja TOTP

HOTP (*HMAC-based One-time Password Algorithm)* ja sitä läheisesti muistuttava TOTP (*Time-based One-Time Password algorithm*) ovat Initiative for Open Authentication -järjestön (OATH) määrittelemiä yksinkertaisia kertakäyttösalasanajärjestelmiä, jotka perustuvat HMAC-autentikointikoodilla varmennettuun kasvavaan laskuriin. Autentikointikoodi esitetään 6-9 numeron mittaisena numerosarjana. Kuten käyttäjä, palvelin tuntee autentikointikoodin avaimen, ja pystyy luomaan omaa laskuriaan vastaavan koodin verratakseen tarkistakseen sen avulla käyttäjän antaman koodin. HOTP-algoritmissa laskurina toimii jokaisella kirjautumisella kasvava arvo, ja TOTP-algoritmissa kellonaika 30 tai 60 sekunnin tarkkuudella. Molemmissa tapauksissa hyväksytään tavallisesti muutama seuraava arvo, siltä varalta että käyttäjän ja palvelimen laskurit joutuvat

epätahtiin. [6] [28] [13]

HOTP ja erityisesti TOTP ovat verrattain yleisesti käytettyjä ja mm. Googlen kaksikeinotunnistaminen käyttää TOTP-algoritmia. Algoritmien rakenne on myös hyvin yksinkertainen, ja perustuu tunnettuihin salausteknisiin rakenteisiin, joita pidetään yleisesti luotettavina muissakin yhteyksissä. Myös tunnistekoodeja tuottavia laitteita ja esim. älypuhelinohjelmistoja on saatavilla useita, ja yksinkertaisen rakenteen vuoksi uuden ohjelmistotototeutuksen laatiminen on tarvittaessa helppoa.

4.4 OATH: OCRA

Toinen OATH-järjestön määrittämä algoritmi on haaste-vaste -algoritmi OCRA. Myös se perustuu HOTP-algoritmin perusrakenteeseen, mutta tukee joustavasti yksittäisen haastekysymyksen lisäksi molemminpuolista autentikointia sekä kellonajan liittämistä vastauksen laskentaan. [15]

Toisin kuin muissa tässä mainituissa menetelmissä, haaste-vaste-tunnistuksessa oikea tunnistautumiskoodi riippuu palvelimen esittämästä haastekoodista, eikä ole tiedossa ennen kirjautumistapahtuman alkua. Siten palvelin voi myös erottaa samanaikaiset kirjautumistapahtumat antamalla niille eri haastekoodit. Eduistaan huolimatta OCRA ei vaikuta olevan erityisen laajalti levinnyt, ja valmiita toteutuksia ei juurikaan vaikuta olevan.

4.5 Yubikey

Ruotsalaisen Yubico -yhtiön Yubikey [31] [32] on ulkoisesti muistitikun näköinen kertakäyttösalasanageneraattori. Laite toimii USB-näppäimistönä ja syöttää sen kyljessä olevaa näppäintä painettaessa salasanan suoraan tietokoneelle, ilman että käyttäjän tarvitsee näppäillä koko salasanaa. Laitteen kertakäyttösalasanat perustuvat jokaisella painalluksella kasvavaan laskuriin, joka salataan yhdessä uniikin tunnisteen kanssa. Palvelin tarkistaa salauksen purettuaan, että laskurin arvo on suurempi kuin aiemmin käytetty.

Yubikey-laitteet ovat uudelleenohjelmoitavissa, eli käyttäjä voi itse tallentaa haluamansa salausavaimet. Myöskin käytetty algoritmi on julkinen, ja Yubicon tarjoamien autentikointipalvelimien lisäksi myös kolmansien osapuolien ohjelmistotototeutuksia on olemassa. Laite voidaan ohjelmoida myös antamaan kiinteän salasanan, ja uudemmat laitteistoversiot tukevat myös HOTP-algoritmia (kts. yllä) sekä toimivat matkapuhelimen kanssa NFC:tä käyttäen (Yubikey NEO).

4.6 OTPW

OTPW on Markus Kuhnin (University of Cambridge) kehittämä ohjelmisto, joka perustuu ennalta laadittuun salasanalistaan. Salasanat koostuvat satunnaisista kirjaimista, numeroista ja merkeistä (Base64 -merkistö), joita järjestelmä pyytää satunnaisessa järjestyksessä. Menetelmä on pohjimmiltaan yleistys yksinkertaisesta kiinteästä salasanasta: yhden salasanan sijaan palvelin vain tallentaa useamman. Ohjelmisto vaikuttaa pääosin hyvin suunnitellulta, mutta ei ole erityisen

tunnettu, ja sen muodostamat salasanat ovat epäkäytännöllisiä kirjainryppäitä. Lisäksi ennalta laadittuun salasanalistaan perustuva menetelmä ei ole erityisen joustava, eikä myöskään mahdollista aikaperustaisia tunnisteita. [19]

4.7 Toisen viestintäkanavan käyttäminen salasanan välitykseen

Eräs tapa kertakäyttöisen tunnisteen tuottamiseen on välittää se käyttäjälle jotakin rinnakkaista kommunikointikanavaa pitkin kirjautumishetkellä (engl. *out-of-band communication*). Käytännössä helpoin keino on matkapuhelimeen lähetettävä tekstiviesti tai robottipuhelu. Tämä on yksi Googlen tunnistamisjärjestelmän tarjoamista vaihtoehdoista [5]. Menetelmän haittapuolena on riippuvaisuus matkapuhelinverkosta, ja siitä mahdollisesti aiheutuvat kustannukset. Algoritmisesti tuotettavia tunnisteita voidaan tuottaa älypuhelimella myös paikallisesti, asentamalla puhelimeen niitä generoiva ohjelmisto jolloin kyse ei ole toisen viestintäkanavan käytöstä. Toisaalta älypuhelimet ja matkapuhelinverkot ovat alttiita tietomurroille, ja tämän vuoksi järjestelmän sitomista juuri puhelimeen voidaan pitää riskinä [10]. Myöskin standardointijärjestö NIST suosittaa julkisen puhelinverkon käyttämistä tunnistamistietojen välittämiseen vain rajoitetusti [12].

4.8 Yhteenveto

Yllä esitetyistä vapaista algoritmeista HOTP ja TOTP ovat yksinkertaisuutensa ja rakenteensa vuoksi selvästi vakuuttavimmat. Järjestelmien suosio helpottaa niiden käyttöä, sillä toteutuksia on helposti saatavilla. Sekä aikaperustaisella että laskuriperustaisella menetelmällä on kummallakin etunsa: aikaperustainen järjestelmä ei vaadi asiakkaan ja palvelimen laskurien pitämistä synkronoituina, olettaen että molemmilla on riittävän tarkka kello. Aikaperustaisessa järjestelmässä myös tunnisteet vanhenevat automaattisesti vaikka niitä ei käytetä. Toisaalta laskuriperusteinen järjestelmä mahdollistaa tunnistekoodien listaamisen vaikka paperille.

Myös Yubikey on omassa kontekstissaan varsin kätevä järjestelmä, mutta haittapuolena se vaatii USB-portin käyttämistä, mikä voi olla mobiililaitteilla hankalaa. Uudemmat Yubikey-laitteet pystyvät tuottamaan myös HOTP- ja TOTP-algoritmin mukaisia koodeja älypuhelimen kelloa käyttäen, mutta palvelintoteutukseen ei vaikuta se, millä laitteella tai ohjelmalla tunnistekoodeja tuotetaan.

5 Käytännön toteutus

5.1 Tavoitteet

Työn käytännöllisenä osana toteutetaan valmiita ohjelmistokomponentteja käyttäen Linux-järjestelmillä käytettävä kertakäyttösalasanajärjestelmä. Aiemmissa luvuissa esitetyn perusteella järjestelmän on syytä tukea kiinteiden ja vaihtuvien salasanojan yhtäaikaista käyttöä. Kiinteät salasanat tulee myös tallentaa turvallisesti sekä autentikointiyritysten määrää aikayksikköä kohti tulisi voida rajoittaa.

Yleiskäyttöisyyden vuoksi järjestelmän toivotaan tukevan keskitettyä kirjautumista useampaan järjestelmään samoilla tunnuksilla. Työn rajaamiseksi keskitytään SSH-kirjautumiseen, mutta toteutus on periaatteessa yleistettävissä myös muihin sovelluksiin.

5.2 Ratkaisut

Luvussa 4 esitetyn perusteella käytetään TOTP-algoritmia. Kuten muutkin laskuriin perustuvat järjestelmät, TOTP vaatii että viimeisintä käytettyä kertakäyttösalasanaa vastaava laskurin arvo on käytettävissä jokaisen kirjautumistapahtuman yhteydessä. Laskuri on siten tallennettava keskitetysti. Keskitetty tunnistaminen on helpointa tehdä käyttäen RADIUS-protokollaa [21], joka on iästään huolimatta yleisesti käytetty, ja jolle on saatavilla runsaasti valmiita toteutuksia. Erityisesti FreeRADIUS -ohjelmisto on helposti laajennettavissa erillisillä moduuleilla [4].

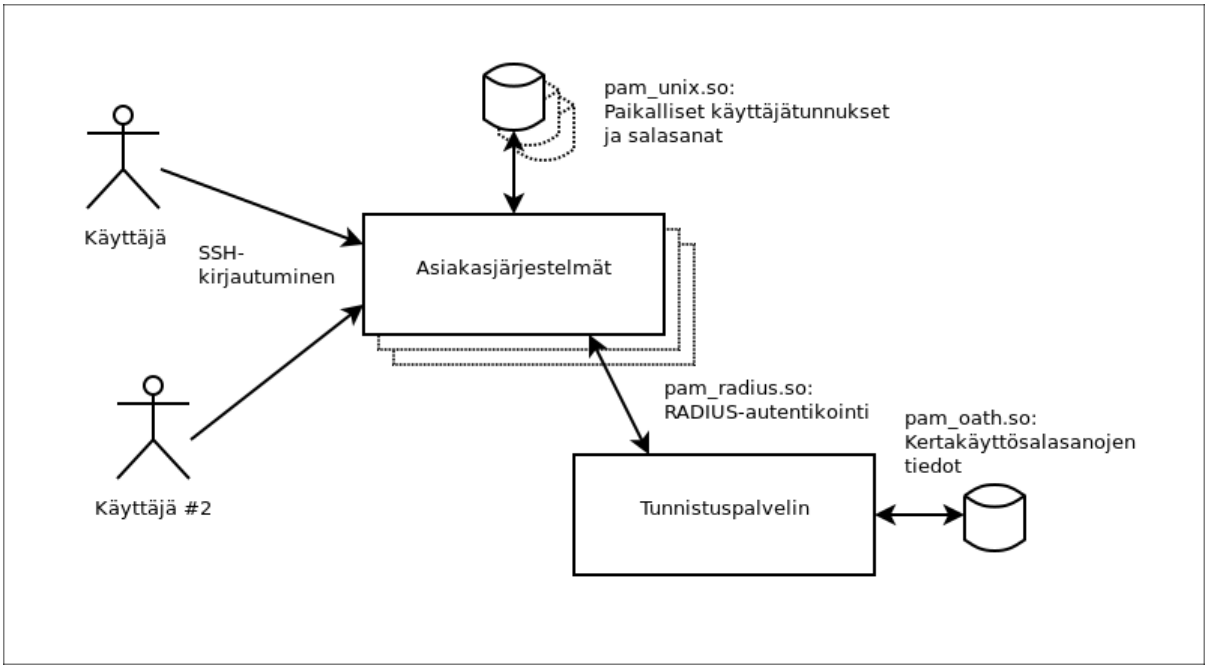
Varsinaisena autentikointikomponenttina käytetään oath-toolkit -ohjelmiston [14] PAM -moduulia (*Pluggable Authentication Module*, Linux-järjestelmissä käytetty autentikointirajapinta). Ohjelmisto ei sellaisenaan mahdollista kaikkia toivottavia ominaisuuksia, mutta niiden toteuttaminen vaatii laajempaa ohjelmistokehitystä, eikä mahdu tämän työn raameihin.

PAM tarjoaa mahdollisuuden pinota erillisiä autentikointimoduuleja, joko siten että yhden moduulin hyväksyntä riittää, tai kaikkien moduulien hyväksyntä vaaditaan. Ensin mainittua yhden moduulin hyväksyntää käytetään, kun esimerkiksi kun osa käyttäjätiedoista on tallennettu paikalliseen tietokantaan, ja osa erilliseen järjestelmään. Tässä tapauksessa sen sijaan halutaan vaatia sekä kiinteän että vaihtuvan salasanan olevan oikein, joten vaaditaan kummankin autentikointimoduulin yhtäaikainen hyväksyntä. Kiinteän salasanan tarkistamiseen voidaan siten käyttää samaa PAM-moduulia kuin tavanomaisessakin kirjautumisessa.

Vaikka vaihtuva salasana on tallennettava keskitetylle tunnistamispalvelimelle, voidaan kiinteät salasanat tallentaa joko keskitetysti samalle tunnistamispalvelimelle, tai erikseen jokaiselle tunnistamisjärjestelmää käyttävälle palvelimelle. FreeRADIUS-palvelin ei kuitenkaan toimi useamman PAM-moduulin kanssa halutulla tavalla, vaan välittää saman salasanan molemmille tunnistusmoduuleille. Keskitetty vaihtoehto ei siten ole tällä toteutuksella mahdollinen.

Järjestelmän olennaiset komponentit ovat siten keskitetty *tunnistuspalvelin*, sitä käyttävät *asiakasjärjestelmät*, sekä näihin yhteyttä ottavat *käyttäjät*. Nämä komponentit on esitetty kuvassa 3. Tunnistuspalvelin ajaa FreeRADIUS-ohjelmistoa, joka konfiguroidaan käyttämään pam_oath.so -moduulia ja tarjoamaan RADIUS-palvelua asiakasjärjestelmille. Asiakasjärjestelmien SSH-palvelin konfiguroidaan käyttämään PAM-autentikointia ja moduuleja pam_radius.so ja pam_unix.so, jotka toteuttavat tunnistamisen RADIUS-palvelinta sekä paikallisesti tallennettuja käyttäjätietoja vastaan.

Asiakasjärjestelmien ja tunnistuspalvelimen välistä liikennettä varten tulee jokaiselle asiakas-järjestelmälle luoda erillinen tunnuskoodi (*shared secret*), jonka avulla ne voivat tunnistaa tunnistuspalvelimen. RADIUS-protokolla tukee vain symmetristä salausta, joten on olennaista että jokaisen asiakasjärjestelmän käyttämä tunnuskoodi on erillinen. Käyttäjien osalta konfigurointi-muutoksia ei tarvita, sillä kertakäyttösalasanan kysely tapahtuu osana tavanomaista SSH-kirjautumistapahtumaa.



Kuva 3: Periaatekuva tunnistamisjärjestelmään liittyvistä laitteista

5.3 Puutteita

Yllä esitetty toteutustapa ei juuri anna mahdollisuutta puuttua yhtäaikaiseen kirjautumiseen perustuvaan hyökkäykseen, sillä sen enempää FreeRADIUS kuin oath-toolkitin PAM-moduuli eivät sisällä tähän liittyvää toiminnallisuutta. Samoin kirjautumisyritysten määrän rajoittaminen keskitetysti on vaikeaa, ja käytännössä se täytyy tehdä jokaisella tunnistamispalvelua käyttävällä asiakaspalvelimella erikseen. Käytetyt valmiit ohjelmistot eivät myöskään tue todellista haaste-vaste -tunnistamista. Näiden ongelmien korjaaminen vaatisi laajempaa jatkokehitystä.

Tässä työssä ei myöskään puututa muiden käyttäjätietojen synkronointiin asiakasjärjestelmien välillä, vaan tunnustiedot pitää levittää asiakasjärjestelmille erikseen tai käyttää erillistä hakemistopalvelinta, kuten LDAP (*Lightweight Directory Access Protocol)*.

Aikaperustaista tunnuskoodia käyttävä järjestelmä rajoittaa hyväksyttyjen kirjautumistapahtumien tiheyttä, koska seuraavaa tunnuskoodia voidaan käyttää vasta sen aktivoituessa ajan myötä. Tämä voi johtaa käytettävyysongelmaan, mikäli käyttäjä haluaisi kirjautua useampaan järjestelmään yhden koodin voimassaoloaikana.

6 Yhteenveto ja johtopäätökset

Työssä käytiin läpi käyttäjäntunnistamisen perusteita, ja tunnistamistilanteen kohdistuvia uhkia/hyökkäyksi, sekä kertakäyttösalasanojen toimivuutta näiden hyökkäysten torjumiseksi. Todettiin että kertakäyttöiset, muuttuvat tunnisteet toimivat hyvänä suojakeinona passiivista salakuuntelua vastaan, mutta eivät toimi erityisen hyvin sellaista aktiivista salakuuntelijaa vastaan, joka voi käyttää kaappaamaansa tunnistetta reaaliajassa, tai joka pystyy muokkaamaan käyttäjän ohjelmistoa. Haaste-vaste -menetelmät toimivat myös reaaliaikaista hyökkäystä vastaan, koska pystyvät selvästi erottamaan samanaikaiset kirjautumisyritykset. Niiden käyttö on kuitenkin hankalampaa kuin ilman tapahtumakohtaista haastetta toimivien menetelmien.

Työn loppuosassa esiteltiin kertakäyttösalasanajärjestelmiä, kuten historiallinen S/Key, markkinaosuudeltaan merkittävät kaupallinen RSA SecurID, sekä avoimet standardit HOTP, TOTP ja OCRA.

Avoimen lähdekoodin komponenteista oath-toolkit -kirjasto tarjoaa valmiin tunnistamismoduulin kertakäyttösalasanoille. Linux-järjestelmien PAM-autentikointirajapintaa ja FreeRADIUS -palvelinta käyttäen voidaan toteuttaa hajautettu tunnistamisjärjestelmä. FreeRADIUS-ohjelmiston PAM-tuen ominaisuuksien vuoksi sekä kiinteän että vaihtuvan salasanan keskitetty tallentaminen ei kuitenkaan ole mahdollista, eivätkä valmiit komponentit myöskään tarjoa erityisen hyviä keinoja rajoittaa keskitetysti kirjautumisyritysten määrää verkko-osoitteen tai käyttäjätunnuksen perusteella.

Lähteet

[1] Facebook -palvelun ohjesivu: “Desktop Help / Security: What's a one-time password and how do I get one?”; haettu 2015-10-20 https://www.facebook.com/help/214309978590084

[2] F-Secure Labs: ”News from the Lab – More on international phishing”, 2005-10-28; haettu 2015-11-24 https://www.f-secure.com/weblog/archives/00000689.html

[3] FreeBSD Handbook, 12.3. One-time Passwords; haettu 2015-10-21 https://www.freebsd.org/doc/handbook/one-time-passwords.html

[4] The FreeRADIUS Project; haettu 2017-02-21 http://freeradius.org/

[5] Googlen palvelujen ohjesivu: “Google 2-Step Verification: Stronger security for your Google Account”; haettu 2015-10-20 https://www.google.com/landing/2step/#tab=how-it-works

[6] M'Raihi D., Bellare M, et al: HOTP: An HMAC-Based One-Time Password Algorithm (RFC 4226), December 2005; saatavilla https://tools.ietf.org/html/rfc4226

[7] Lamport, Leslie: Password authentication with insecure communication (Communications of the ACM, Volume 24 Issue 11; November 1981)

[8] LastPass -palvelun ohjesivu: “User Manual, One Time Passwords”; haettu 2015-10-20 https://helpdesk.lastpass.com/security-options/one-time-passwords/

[9] Lappeenrannan teknillisen yliopiston käyttäjätunnusohjeet; haettu 2015-11-24 https://uni.lut.fi/itohjeet

[10] Mulliner, C at al: SMS-based One-Time Passwords: Attacks and Defense, Technische Universität Berlin, Technical Report, September 2014 https://www.eecs.tu-berlin.de/fileadmin/f4/TechReports/2014/tr_2014-02.pdf

[11] National Cyber Security Centre (UK): The problems with forcing regular password expiry; haettu 2017-05-16 https://www.ncsc.gov.uk/articles/problems-forcing-regular-password-expiry

[12] National Institute of Standards and Technology: NIST Special Publication 800-63B: Digital Identity Guidelines - Authentication and Lifecycle Management; viitattu 2017-06-28 https://pages.nist.gov/800-63-3/sp800-63b.html

[13] Initiative for Open Authentication http://www.openauthentication.org/

[14] Josefsson, Simon: Introducing the OATH Toolkit, 2011; haettu 2017-02-21 https://blog.josefsson.org/2011/01/20/introducing-the-oath-toolkit/

[15] M'Raihi D., Rydell J.: OCRA: OATH Challenge-Response Algorithm (RFC 6287), June 2011; saatavilla https://tools.ietf.org/html/rfc6287

[16] O'Gorman, Lawrence: Comparing passwords, tokens, and biometrics for user authentication (Proceedings of the IEEE; Volume: 91, Issue: 12; December 2003)

[17] McDonald, D., Atkinson R.: One Time Passwords In Everything (OPIE): Experiences with Building and Using Stronger Authentication. Usenix 1995; saatavilla https://www.usenix.org/legacy/publications/library/proceedings/security95/full_papers/mcdonald.pdf

[18] Haller N. et al: A One-Time Password System (RFC 2289), February 1998; saatavilla http://tools.ietf.org/html/rfc2289

[19] Markus Kuhn: OTPW – A one-time password login package, University of Cambridge; haettu 2015-10-22 https://www.cl.cam.ac.uk/~mgk25/otpw.html

[20] Pfleeger C., Pfleeger S.L.: Security in Computing, 4th ed. Prentice Hall 2007

[21] Rigney C. et al: Remote Authentication Dial In User Service (RADIUS) (RFC 2865), June 2000; saatavilla https://tools.ietf.org/html/rfc2865

[22] The Register: "Phishing attack targets one-time passwords", 2005-10-12; haettu 2015-11-24 http://www.theregister.co.uk/2005/10/12/outlaw_phishing/

[23] RSA SecurID; haettu 2015-10-22 http://www.emc.com/security/rsa-securid.htm

[24] Haller N.: The S/KEY One-Time Password System (RFC 1760); saatavilla https://www.ietf.org/rfc/rfc1760.txt

[25] Stallings W., Brown: Computer Security: Principles and Practice, 2nd ed. Pearson 2012

[26] Stavroulakis P, Stamp M: Handbook of Information and Communication Security. Springer 2010; http://link.springer.com/book/10.1007%2F978-3-642-04117-4

[27] stoken - Software Token for Linux/UNIX; haettu 2015-10-22 http://sourceforge.net/p/stoken/wiki/Home/

[28] M'Raihi D., Machani S. et al: TOTP: Time-Based One-Time Password Algorithm (RFC 6238), May 2011; saatavilla https://tools.ietf.org/html/rfc6238

[29] Viestintäviraston "Tietoturva nyt!" -julkaisu, 2014-07-28; https://www.viestintavirasto.fi/kyberturvallisuus/tietoturvanyt/2014/07/ttn201407281137.html

[30] Yan, J et al: The memorability and security of passwords – some empirical results (Technical Report, Number 500, University of Cambridge Computer Laboratory; September 2000)

[31] Yubico AB: Yubikey Hardware; haettu 2015-03-01
https://www.yubico.com/products/yubikey-hardware/

[32] Yubico AB: The YubiKey Manual version 3.3; 2014-09-17; haettu 2015-03-01
https://www.yubico.com/wp-content/uploads/2014/10/YubiKey-Manual-v3.3.pdf

[33] Zhang Y., Monrose F., Reiter M.: The security of modern password expiration: an algorithmic framework and empirical analysis (Proceedings of the 17th ACM Conference on Computer and communications security, 2010)

[34] Menezes A., van Oorschot P., Vanstone S.: Handbook of Applied Cryptography. CRC Press 1997. Sähköinen versio saatavilla: http://cacr.uwaterloo.ca/hac/

Liite A: Ohjelmistokomponenttien konfigurointi

Tässä liitteessä esitetään tarkemmin työn toteutuksessa käytettyjen komponenttien konfigurointi ja olennaisimmat asetukset. Esimerkeissä käytetään seuraavia IP-osoitteita: tunnistamispalvelin 10.0.111.9 ja asiakasjärjestelmä 10.0.111.11 sekä näiden välinen jaettu salaisuus `xXzyqIZq`. Esitetyt tiedostonimet ovat Debian-järjestelmässä käytetyt.

Asiakasjärjestelmien konfigurointi

Konfiguroitavat komponentit: SSH-palvelin (sshd), SSH:n PAM-asetukset, ja pam_radius_auth -moduuli.

Varmistetaan että SSH-palvelin käyttää PAM-moduuleja kirjautumiseen

```
/etc/ssh/sshd_config:
    ChallengeResponseAuthentication yes
    UsePAM yes
```

Konfiguroidaan SSH:n käyttämät PAM-moduulit (jatkorivi merkitty kenoviivalla):

```
/etc/pam.d/sshd:
    auth required pam_unix.so nullok_secure
    auth required pam_radius_auth.so conf=/etc/pam_radius_auth.conf \
           force_prompt prompt=OTP
```

Debian-järjestelmissä voidaan myös viitata järjestelmän yhteisiin asetuksiin ja lisätä vain radius-moduuli:

```
/etc/pam.d/sshd:
    @include common-auth
    auth required pam_radius_auth.so conf=/etc/pam_radius_auth.conf \
           force_prompt prompt=OTP
```

Konfiguroidaan RADIUS-palvelimet joita pam_radius_auth käyttää:

```
/etc/pam_radius_auth.conf:
    # <server IP> <shared secret> <timeout>
    10.0.111.9    xXzyqIZq        3
```

pam_radius_auth -moduuli pystyy käyttämään vikasietoisuuden vuoksi useampaa palvelinta, mutta se ei ole tässä keskitetyssä ratkaisussa mahdollista. Palvelimia kuvaavien rivien sarakkeet ovat palvelimen IP-osoite; asiakkan ja palvelimen välinen jaettu salaisuus; ja aika, jonka RADIUS-asiakas odottaa vastausta sekunteina. Useampaa asiakasjärjestelmää käytettäessä on muistettava luoda jokaiselle erillinen jaettu salaisuus.

Tunnistuspalvelimen konfigurointi

Konfiguroidaan RADIUS-palvelimen käyttämä PAM-moduuli:

```
/etc/pam.d/radiusd:
    auth required pam_oath.so usersfile=/etc/oath/users window=3 digits=6
```

Parametri `usersfile` kertoo oath-moduulin käyttäjätietokannan sijainnin, ja `window` kuinka monta seuraavaa kertakäyttösalasanaa hyväksytään. Tätä käytetään siltä varalta että käyttäjän ja palvelimen kellot eivät pysy täysin samassa ajassa. Käyttäjätietokanta (`/etc/oath/users`) tulee luoda siten, että RADIUS-palvelimella on kirjoitusoikeus sekä tiedostoon että sen sisältävään hakemistoon. Käyttäjätiedosto sisältää yhden käyttäjätunnuksen kullakin rivillä, esim.

```
/etc/oath/users
     HOTP    testuser        -       1234567890abcdef
```

Kentät ovat algoritmityyppi, käyttäjätunnus, viiva, sekä kertakäyttösalasanan jaettu salaisuus (heksanumeroina). pam_oath lisää tiedostoon viimeisen käytetyn tunnuksen ja viimeisen kirjautumisajan sisältäviä merkintöjä. Algoritmityyppi kertoo, onko kyseessä tapahtuma- vai aikapohjainen laskuri:

```
     HOTP      - tapahtumapohjainen laskuri
     HOTP/T30  - aikapohjainen laskuri, 30 sekunnin välein päivittyvä
     HOTP/T60  - aikapohjainen laskuri, 60 sekunnin välein päivittyvä
```

FreeRADIUS -palvelimen asetukset: Määritetään palvelin käyttämään PAM-kirjastoa tunnistamis-menetelmänä kaikissa tilanteissa. `authenticate` -osiossa voi olla myös muita määrityksiä, mutta ne voidaan poistaa.

```
/etc/freeradius/users:
     DEFAULT Auth-Type := PAM

/etc/freeradius/sites-enabled/default:
     authenticate {
          pam
     }
```

Palvelimen tuntemat asiakasjärjestelmät ja niiden jaetut salaisuudet:

```
/etc/freeradius/clients.conf:
     client host1 {
          ipaddr = 10.0.111.11
          secret = xXzyqIZq
          nastype = other
     }
```

Lokitietojen tallentaminen voidaan kytkeä päälle `log` -osiossa. Muut samassa osiossa olevat asetukset voidaan jättää paikalleen.

```
/etc/freeradius/radiusd.conf:
     log {
          auth = yes
     }
```