Opinnäytetyö (AMK)

Tieto- ja viestintätekniikka

2019

Aleksi Vuorinen

HYÖKKÄYSMENETELMIEN HAVAITSEMINEN SYSMONIN AVULLA

TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tieto- ja viestintätekniikka

2019 | 39 sivua

Aleksi Vuorinen

HYÖKKÄYSMENETELMIEN HAVAITSEMINEN SYSMONIN AVULLA

Kohdistettujen hyökkäyksien sekä erilaisten hyökkäyskampanjoiden määrän noustessa yritykset ovat miettineet miten hyökkäyksiin kuluvaa reagointi- ja havaitsemisaikaa voisi lyhentää. Opinnäytetyössä käsiteltiin uhkien havaitsemista hyökkäyksen aikana Windowsin tapahtumalokeista löydetyistä poikkeamista. Työssä keskityttiin erillisen Sysmon-auditointilisäosan tuottamiin tapahtumalokeihin. Ennen hyökkäystä havainnollistettiin Sysmonin toimintaa ja sen ylläpitoa.

Työn aikana simuloidun hyökkäyksen toteutuksessa hyödynnettiin Microsoftin laatimaa prosessikaaviota, joka pohjautuu asiantuntijoiden laatimaan tietoon tunnetuiden hyökkäysten kulusta. Simuloitu hyökkäys kohdistettiin virtualisoituun toimialueympäristöön, joka kuvastaa pienen tai keskisuuren yrityksen toimintamallia. Hyökkäyksestä aiheutuneita tapahtumalokeja tarkasteltiin erillisestä lokienhallintajärjestelmästä, joka asennettiin toimialueen rinnalle virtuaaliympäristöön.

Hyökkäyksestä aiheutuneita tapahtumalokin poikkeamia analysoitiin ja tuloksia verrattiin osittain eri tutkimuksista saatavilla oleviin tuloksiin. Hyökkäystyökalujen sekä tekniikoiden osalta pyrittiin käyttämään tuoreita esimerkkejä, joista löytyi vapaasti saatavilla olevaa dokumentointia. Havaituista poikkeamista koostettiin lopuksi Sysmonin sääntötiedosto, jolla työssä toteutetut hyökkäykset pystytään havaitsemaan.

Opinnäytetyön tavoitteessa parantaa hyökkäysten havainnointikykyä Sysmonin avulla onnistuttiin, vaikka toteutetut hyökkäysesimerkit olivatkin kärjistettyjä, eivätkä esimerkkinä toimineet menetelmät sisältäneet kaikkia mahdollisia hyökkäystyökaluja. Jokaisesta hyökkäysvaiheesta saatiin kerättyä lokitietoja, joita voidaan hyödyntää uhan havaitsemisen indikaattorina. Tavoitteen onnistumisesta huolimatta, lopputuloksilla ei kuitenkaan voida havaita jokaista mahdollista hyökkäystä ja siksi onkin suositeltavaa laajentaa Sysmonin pohjalla toimivia sääntöjä.

ASIASANAT:

tietoturva, lokienhallinta, kyberhyökkäys, virtuaaliympäristö, mimikatz, sysmon

BACHELOR´S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information technology

2019 | 39 pages

Aleksi Vuorinen

DETECTING CYBER ATTACK TECHNIQUES WITH SYSMON

As targeted attacks and the number of different attack campaigns increase, companies have wondered how the response and detection time for these attacks could be reduced. The thesis dealt with the detection of anomalies found from the Windows event logs during the attack. The work focused on the event logs produced by a separate Sysmon audit add-on. The operation of Sysmon and its maintenance were illustrated before the attack.

Process diagram of well-known attacks developed by Microsoft was utilized on the execution of simulated attack. The simulated attack was targeted to a virtualized domain environment that reflects a small or medium-sized business model. The event logs caused by the attack were examined on a separate log management system installed alongside the virtualized domain environment.

The event log anomalies caused by the attack were analyzed and the found results were partly compared with existing online studies. Recent examples of freely available documentation of attack tools and techniques were utilized during the simulation. The anomalies observed were compiled in to a Sysmon rule file which detects attacks used in the simulation.

The goal of the thesis to improve detection of attacks using Sysmon was successful, even though exemplary attacks were exaggerated, and the exemplary attack methods did not include all the possible attack tools available. From each attack phase anomalies were found which can be used as an indicator of attack. Despite the success of the goal, the end results cannot be used to detect every possible attack and it is therefore advisable to expand and develop rules which Sysmon relies on.

KEYWORDS:

cybersecurity, log management, cyberattack, virtual environment, sysmon, mimikatz

SISÄLTÖ

KUVAT

TAULUKOT

LYHENTEET

Sysmon	System Monitor, järjestelmä- ja prosessitason lokien keräämiseen tarkoitettu lisäosa.
ELK STACK	Elastic, Logstash ja Kibana -kokonaisuus lokienhallintaan.
TTP	Techniques, tactics, and procedures. Tekniikat, taktiikat ja menettelyt.
FTP	File Transfer Protocol. TCP-protokollaa käyttävä tiedonsiirtomenetelmä kahden tietokoneen välille.
SFTP	SSH File Transfer Protocol. SSH-protokollaa käyttävä tiedonsiirtomenetelmä.
GPO	Group Policy Object. Hallitsee toimialueen työasemia ja palvelimia.
LSASS	Local Security Authority Subsystem Service, Windowsin palvelu, jonka vastuulla todentaminen on.
NTLM	New Technology Local Area Network Manager. Windowsin todennusprotokolla.
TGT	Ticket Granting Ticket. Lipun myöntävä lippu, jota käytetään Kerberos-protokollassa.

1 JOHDANTO

Viime aikoina uutisotsikoihin nousseet tietomurrot ja hyökkäysyritykset ovat herättäneet paljon keskustelua yritysmaailmassa siitä, miten niitä voitaisiin havaita paremmin ja mahdollisesti ehkäistä. Suurimpien tietomurtojen takana ovat olleet usein kyberhyökkäyksiin erikoistuneet rikollisjärjestöt, joista osa on spekuloitu olevan eri valtioiden tukemia tai rahoittamia. Tämän kaltaisia hyökkäyksiä kutsutaan kohdistetuiksi hyökkäyksiksi. Tyypillisesti hyökkäyksien motiivi perustuu taloudellisen hyödyn tavoittelemiseen, mutta nykyään kyberhyökkäykset ovat osana myös nykyaikaista sodankäyntiä. Nykyaikaisen sodankäynnin motiivina ei ole suoraan hyökätä, vaan tavoitteena saattaa olla esimerkiksi tiedustelu, poliittinen vaikuttaminen tai jalansijan saaminen tärkeistä kohteista.

Suojautuminen kohdistetuilta hyökkäyksiltä, jotka ovat tarkoin suunniteltuja, saattaa olla hankalaa. Edistyneillä hyökkäysmenetelmillä pystytään mahdollisesti ohittamaan yritysten tiukoin määritellyt tietoturvaprotokollat. Vuonna 2018 Symantecin (2018) teettämän tutkimuksen mukaan suosituin hyökkäysvektori on ollut sähköposti 71-prosenttisesti. Sähköpostin kautta tulevilla hyökkäyksillä pyritään saamaan käyttäjä mahdollisesti vierailemaan haitallisella sivustolla tai esimerkiksi avaamaan makroilla muokattu Office- dokumentti, joka jättää takaoven käyttäjän koneelle. Haitallisen sivuston kautta kalastellut käyttäjätunnukset tai koneelle asentunut takaovi mahdollistaa hyökkääjän liikkumisen syvemmälle yrityksen järjestelmiin. Hyökkäysvektoreita on monia, mutta sähköposti on niistä suosituin.

Suuri ongelma yrityksissä on uhkien havainnointikyky ja niihin reagoiminen. Verizonin (2018) raportin mukaan tartunnan jälkeen noin kaksi kolmasosa uhista huomataan vasta yli kuukauden jälkeen.

Opinnäytetyön tarkoituksena on tutkia, miten uhkien havaitsemista voisi nopeuttaa työasemien, sekä palvelinten lokitiedoista löydetyistä poikkeamista. Lokitiedot kerätään keskitettyyn lokienhallintajärjestelmään, joka käy läpi lokeja sille annettujen sääntöjen perusteella ja hälyttää säännön osuessa havaittuun mahdolliseen poikkeamaan. Työn aikana demonstroidaan erilaisia hyökkäysmenetelmiä virtuaaliympäristössä, johon kyseinen lokienhallintajärjestelmä on implementoitu. Hyökkäyksien jälkeen kohteena olleen työaseman tai palvelimen lokidataa analysoidaan ja pyritään etsimään hyökkäyksistä aiheutuneet poikkeamat. Löydetyistä poikkeamista voidaan luoda uusia sääntöjä,

jotka auttavat havaitsemaan mahdollisen hyökkäyksen ja sen avulla nopeuttaa niihin reagoimista.

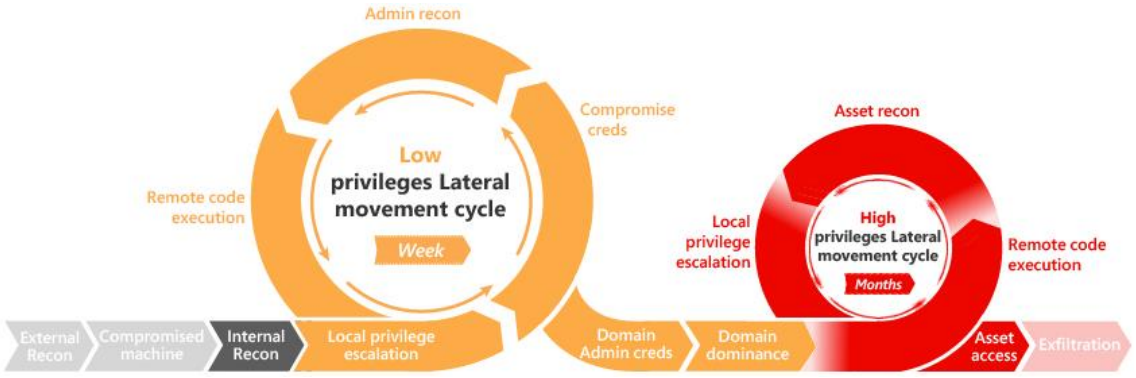
Uhkien havaitsemista tutkitaan Sysinternalsin kehittämällä Sysmon-auditointilaajennuksella. Windowsin natiivin auditoinnin tuottama lokidata on osittain hyvää, jos halutaan esimerkiksi seurata käyttäjien todentamista tai tärkeiden käyttäjäryhmien muutoksia toimialueessa. Auditoinnista puuttuu kuitenkin tarkka prosessi- ja järjestelmätason seuranta, joka on tärkeä osa hyökkäyksien ja mahdollisten haittaohjelmien havaitsemiseen. Sysmonin avulla tämänkaltainen auditointi on mahdollista.

Virtuaalinen testiympäristö, jota vastaan hyökätään, koostuu toimialuepalvelimesta ja työasemista. Toimialuepalvelin on asennettu vuoden 2016 Windows-palvelimelle. Työasemilla on jo valtaosalla käytössä oleva Windows 10 -käyttöjärjestelmä. Testiympäristön tarkoitus on kuvastaa pienen tai keskisuuren yrityksen mukaista ympäristöä, jossa tietokoneita ja käyttäjiä on hallittava toimialuepalvelimelta. Keskitetyssä lokienhallintajärjestelmässä käytetään Elasticin tarjoamaa ELK Stack -kokonaisuutta, johon on yhdistetty erillinen palvelin lokien keräämistä ja analysointia varten.

2 HYÖKKÄYKSEN KULKU

2.1 Microsoft ATA Kill Chain

Microsoftin Global Incident Response and Recovery (GIRR) -ryhmä ja Enterprise Threat Detection Service ovat luoneet havaintoihinsa perustuvan prosessikaavion kohdennetun hyökkäyksen kulusta (Kuva 1). Kaaviossa kuvataan tyypillinen hyökkäyksen kulku, mukaan lukien tekniikat, taktiikat ja menettelyt, eli TTP:t (techniques, tactics, and procedures), joita hyökkääjät käyttävät organisaation verkkoihin ja järjestelmiin hyökätessä. Kaavio koostuu kahdesta päävaiheesta, joista ensimmäinen on korkeampien käyttäjätunnuksien sisäistä tiedustelua sekä levittäytymistä ja toinen on lyhyemmällä aikavälillä tapahtuva vaihe, jossa hyökkääjä on saanut halutut korkeamman käyttövaltuuden omaavat tunnukset haltuunsa. (Microsoft 2016.)



Kuva 1. Microsoft ATA Kill Chain (Microsoft 2016).

2.2 Ulkoinen tiedustelu

Ulkoisessa tiedusteluvaiheessa hyökkääjä etsii mahdollisimman paljon julkisesti saatavilla olevaa tietoa kohteestaan. Tietoja hankitaan esimerkiksi kohteen IP-osoitealueesta, liiketoiminnasta, teknologiasta, työntekijöistä ja johtajista. Tämän vaiheen tavoitteena on haravoida riittävästi tietoa, jonka avulla saadaan nostettua kohdennetun hyökkäyksen onnistumismahdollisuuksia. Tiedustelua on kahdenlaista, aktiivista ja passiivista.

Aktiivinen tiedustelu on hyökkäys, jossa hyökkääjä on yhteydessä kohdistettuun verkkoon skannaamalla sen avoimia palveluita ja haavoittuvuuksia. Passiivinen tiedustelu on pyrkimys saada tietoa kohdennetusta tietojärjestelmästä ja työntekijöistä ilman yhteyden luomista. Tämä voidaan tehdä esimerkiksi keräämällä tietoa käyttäjistä ja teknologiasta yhtiön verkkosivuilta. Verkkosivut voivat sisältää sähköposteja, puhelinnumeroita tai käyttäjien sosiaalisen median tilejä (Velazquez 2015). Avoimien hakukoneiden avulla pystytään myös hakemaan hyvin laajasti tietoa yrityksestä, työntekijöistä ja sen käyttämästä teknologiasta.

2.3 Tartunta

Tiedustelussa kerätyillä tiedoilla päätetään, mitä hyökkäysvektoria halutaan käyttää. Hyökkäysvektorina voi olla esimerkiksi koodin etäsuorittamisen mahdollistava haavoittuvuus julkiverkon palvelussa tai työntekijälle lähetetty kalastelusähköposti. Sähköposti voi sisältää haitallisen liitetiedoston, joka asentaa uhrin tietokoneeseen etäyhteyden mahdollistavan haittaohjelman.

Hyökkääjien tiedetään myös tartuttaneen kolmannen osapuolen sivustoja, joissa kohdeyrityksen työntekijöiden tiedetään vierailevan. Esimerkiksi vuonna 2017 elokuun ja syyskuun välisenä aikana tunnetun levytilan siivoustyökalun CCleanerin latauspalvelimen jakama asennustiedosto sisälsi haittaohjelman. CCleaner.exe-tiedoston luvaton muokkaus johti kaksivaiheisen takaportin asentamiseen, joka mahdollisti pääsyn sovelluksen ladanneen käyttäjän järjestelmään (Yung 2017.) Kyseinen hyökkäysstrategia tunnetaan nimellä Watering Hole.

2.4 Sisäinen tiedustelu ja leviäminen

Kun hyökkääjällä on jalansija organisaation verkossa, hän alkaa kerätä tietoja, joita ei ole ollut aiemmin saatavilla ulkopuolelta. Tähän kuuluu esimerkiksi sisäverkon ja sen laitteiden kartoittaminen ja avoimien verkkolevyjen etsiminen. Hyökkääjä alkaa myös käyttämään vapaasti saatavilla olevia, mutta erittäin tehokkaita työkaluja, kuten Mimikatzia ja Windows Credentials Editoria (WCE). Työkaluilla pyritään hyväksikäyttämään käyttöjärjestelmän muistissa olevia kirjautumistietoja ja niiden avulla saamaan haltuun

korkeamman oikeuden omaavia toimialueen käyttäjiä. Hyökkääjän tarkoitus on levittäytyä mahdollisimman laajalle ympäristöön, jotta tiedonkeruu helpottuu ja kiinni jäädessä uhkaa olisi vaikea poistaa.

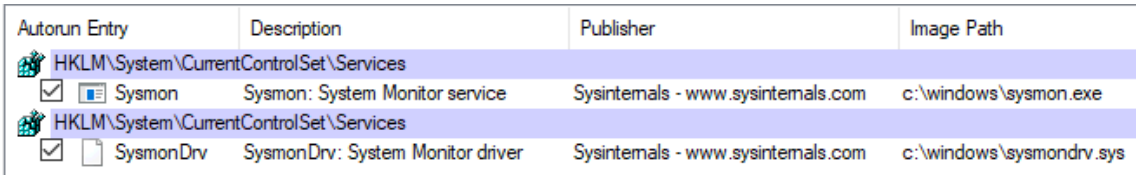
2.5 Hallinta ja tavoite

Hyökkääjän lopullinen tavoite on yleensä saada toimialueen ylläpitäjän oikeudet haltuunsa, joiden avulla on mahdollista päästä käsiksi kaikista kriittisimpään dataan. Halutun datan löytyessä, hänen on pakattava ja pystyttävä lähettämään se verkosta ilman, että sitä havaitaan tai estetään. Tämä tapahtuu tyypillisesti salaamalla tiedot ja siirtämällä ne käyttäen esimerkiksi hyväksyttyjä FTP- tai SFTP-protokollia. Jos tavoitteena ei ole tärkeän datan ryöstäminen, niin toimialueen ylläpito-oikeuksilla pystyy tekemään käytännössä mitä vain yrityksen verkossa.

3 SYSMON

3.1 Asennus

System Monitor, eli Sysmon on Sysinternalsin kehittämä lisäosa Windowsin auditoinnille, joka kerää lokidataa prosessi- ja järjestelmätason toiminnasta Windowsin tapahtumalokiin. Sysmon asentaa laiteohjaimen koneen HKLM\SYSTEM\CurrentControlSet\Services-rekisteriin. Laiteohjain pysyy siis päällä keräten lokitapahtumia läpi koneen käynnistymisen ajan, vaikka käyttäjä ei kirjautuisikaan sisään (Kuva 2).



Kuva 2. Sysmonin services-rekisteri.

Sysmonin asennuspaketti on ladattavissa Microsoftin sivuilta Sysinternals-osiosta. Latauksen jälkeen asennus tapahtuu hyvin yksinkertaisella komennolla komentoriviltä, joka on käynnistetty administrator-oikeuksilla. Komennossa määritellään myös sääntötiedosto, jota halutaan käyttää (Kuva 3). Isommissa toimialueissa on myös mahdollista suorittaa asennus keskitetysti useammalle työasemalle GPO:n avulla.

```
C:\Users\Ale\Desktop>sysmon -accepteula -i c:\users\ale\desktop\konfiguraatio.xml

System Monitor v9.01 - System activity monitor
Copyright (C) 2014-2019 Mark Russinovich and Thomas Garnier
Sysinternals - www.sysinternals.com

Loading configuration file with schema version 4.20
Configuration file validated.
Sysmon installed.
SysmonDrv installed.
Starting SysmonDrv.
SysmonDrv started.
Starting Sysmon..
Sysmon started.
```

Kuva 3. Sysmonin asennuskomento.

3.2 Tapahtumakategoriat ja havainnointikyky

Sysmon kerää lokitapahtumia yhteensä 21 eri kategoriasta. Seuraavissa kappaleissa on lueteltu muutamia esimerkkejä tärkeimmistä tapahtumista ja niiden havainnointikyvyistä.

Process Create -tapahtumasta voidaan havaita haitallisten ohjelmien aiheuttamat alkuinfektiot. Process Create -tapahtumasta käy ilmi, kuka käyttäjä sen on käynnistänyt, mikä sovellus sen on käynnistänyt ja millä komennolla se on käynnistetty. Tapahtuma tallentaa myös prosessin allekirjoituksen salauksen sekä MD5, että SHA256-muodossa. Näitä arvoja voidaan verrata jo ennalta tunnettujen prosessien arvoihin varmistaakseen prosessin aitous.

Network connection -tapahtuman avulla voidaan analysoida tietokoneen verkkoliikennettä. Verkkoliikennettä monitoroimalla nähdään mahdolliset yhteydenotot haitallisiin osoitteisiin ja hyökkääjän liikkuminen toimialueessa. Tapahtumasta käy ilmi yhteyden luoja (prosessi ja käyttäjä), sekä lähde- ja kohdeosoitteet portteineen.

CreateRemoteThread-tapahtumasta havaitaan, jos prosessi injektoi koodia toiseen prosessiin. Haittaohjelmat käyttävät yleensä tätä tekniikkaa esimerkiksi toiminnan naamioimiseen alkutartunnan jälkeen. Lokitiedoista on nähtävissä lähde- ja kohdeprosessi.

Process accessed -tapahtuma kerätään, kun prosessi käyttää tai lukee toisen prosessin resursseja onnistuneesti. Tapahtuman tiedot kertovat lähde- ja kohdeprosessin, sekä saadut oikeudet kohdistuneeseen prosessiin.

Registry events -tapahtumia on kolme ja ne seuraavat tietokoneen rekisteriin tehtyjä muutoksia. Tapahtumilla pystytään esimerkiksi havaitsemaan muutokset autorun-rekisteripolkuun, joka määrittää mitkä sovellukset tai palvelut käynnistyvät tietokoneen käynnistyessä. Haittaohjelmat käyttävät tätä tekniikkaa esimerkiksi persistenssin takaoven luomiseen.

Alla olevassa taulukossa on esitetty lista kaikista Sysmonin tapahtumakategorioista ja niiden tapahtumanumeroista (Taulukko 1).

Taulukko 1. Sysmonin tapahtumakategoriat.

Tapahtumakategoria	Tapahtumanumero
Process Create	1
File creation time	2
Network connection detected	3
Sysmon service state changed	4
Process terminated	5
Driver Loaded	6
Image loaded	7
CreateRemoteThread detected	8
RawAccessRead detected	9
Process accessed	10
File created	11
Registry object added or deleted	12
Registry value set	13
Registry object renamed	14
File stream created	15
Sysmon configuration changed	16
Named pipe created	17
Named pipe connected	18
WmiEventFilter activity detected	19
WmiEventConsumer activity detected	20
WmiEventConsumerToFilter activity detected	21

3.3 Konfiguraatio

Tapahtumalokiin tallentuvia tapahtumia on mahdollista suodattaa erikseen määriteltävällä sääntötiedostolla. On suotavaa harkita, tarvitseeko valvoa kaikkia tapahtumia, koska liiallinen lokien kerääminen hidastaa analysointia ja vie ylimääräistä levytilaa. Sääntötiedoston voi määrittää Sysmonin asennuksen yhteydessä tai jälkeenpäin. Suodattamisessa pitää myös miettiä, onko esimerkiksi tietyn prosessin tapahtumien valvominen tietoturvan kannalta olennaista. Kannattaako tallentaa kaikki tapahtumat, vai pelkästään osa tapahtumista? Mitä jää huomaamatta, jos ei tallenneta tiettyä osa-aluetta?

Sääntötiedosto on XML-formaatissa, joten sitä on helppo lukea ja muokata. Rakenne muodostu Sysmonin lokikategorioista, jotka esitettiin edellisessä luvussa. Raaka Sääntötiedosto ilman sääntöjä koostuu pelkästään include- ja exclude-osioista. Tämän päälle on tarkoitus rakentaa halutut säännöt lokien suodattamista varten. Kategorialle annettaessa pelkästään exclude-arvo, tallentaa Sysmon kaikki lokitapahtumat sen osalta. Pelkällä include-arvolla on puolestaan vastakkainen vaikutus, eikä tapahtumia tallenneta.

HashAlgorithms-rivillä määritellään missä muodossa prosessien allekirjoitusten salaukset tallennetaan. Kärjistetty esimerkki sääntötiedostosta, joka tallentaa pelkästään ProcessCreate -tapahtumat (Kuva 4).

```
<Sysmon schemaversion="4.20">
    <HashAlgorithms>md5,imphash,sha256</HashAlgorithms>
    <EventFiltering>
        <ProcessCreate onmatch="exclude"/>
        <FileCreateTime onmatch="include"/>
        <NetworkConnect onmatch="include"/>
        <DriverLoad onmatch="include"/>
        <ImageLoad onmatch="include"/>
        <CreateRemoteThread onmatch="include"/>
        <RawAccessRead onmatch="include"/>
    </EventFiltering>
</Sysmon>
```

Kuva 4. Kärjistetty esimerkki sääntötiedostosta.

Luodaan 5 esimerkkiä raa'an sääntötiedoston pohjalle:

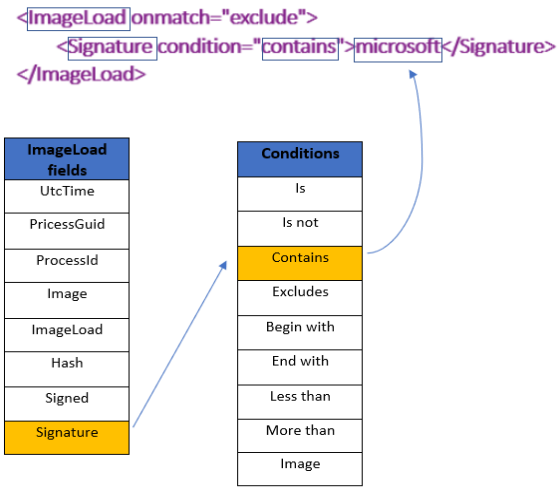
1. Sisällytä vain tapahtumat, jossa komentorivi sisältää sanan ”whoami”
2. Hylkää kaikki tapahtumat, joissa aiheuttajana on F-Securen ohjelma
3. Sisällytä vain Google Chromen generoimat NetworkConnect-tapahtumat
4. Hylkää ImageLoad-tapahtumat, jotka ovat Microsoftin allekirjoittamia. Kaikki muut tallennetaan
5. Sisällytä vain CreateRemoteThread-tapahtumat, joissa TargetImage on winlogon.exe tai lsass.exe

Tiedostoon on lisätty kommentit helpottaakseen hahmottamista. Annetuilla ehdoilla sääntötiedosto näyttää seuraavalta (Kuva 5).

```
<Sysmon schemaversion="4.20">
    <HashAlgorithms>md5,imphash,sha256</HashAlgorithms>
    <EventFiltering>
        <!--Log only process creations where command line contains whoami (event id 1)-->
1.      <ProcessCreate onmatch="include">
            <CommandLine name="System owner discovery" condition="contains">whoami</CommandLine>
        </ProcessCreate>
        <!--Exclude events generated by F-Secure-->
2.      <ProcessCreate onmatch="exclude">
            <ParentImage condition="is">C:\Program Files (x86)\F-Secure\SAFE\fsscan.exe</ParentImage>
        </ProcessCreate>
        <!--Don't log file creation time changes (event id 2)-->
        <FileCreateTime onmatch="include"/>
        <!--Log only Google Chrome network activity (event id 3)-->
3.      <NetworkConnect onmatch="include">
            <Image condition="contains">chrome.exe</Image>
        </NetworkConnect>
        <!--Don't log drivers loaded (event id 6)-->
        <DriverLoad onmatch="include"/>
        <!--Log all images loaded except Microsoft-signed (event id 7)-->
4.      <ImageLoad onmatch="exclude">
            <Signature condition="contains">microsoft</Signature>
        </ImageLoad>
        <!--Log only thread injections into winlogon and lsass (event id 8)-->
5.      <CreateRemoteThread onmatch="include">
            <TargetImage condition="image">lsass.exe</TargetImage>
            <TargetImage condition="image">winlogon.exe</TargetImage>
        </CreateRemoteThread>
        <!--Don't log rawaccessread (event id 9)-->
        <RawAccessRead onmatch="include"/>
    </EventFiltering>
</Sysmon>
```

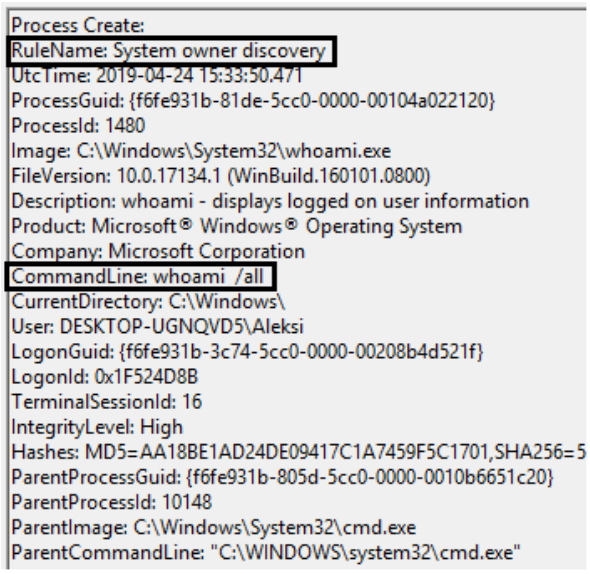
Kuva 5. Esimerkkisäännöt.

Säännöissä arvo on jokin alue lokitapahtumasta, jolle annetaan ehto. Esimerkiksi neljännessä kohdassa, jossa suodatetaan pois Microsoftin allekirjoittamat järjestelmäkuvat, on alueena ”signature”, ehtona ”contains” ja arvona ”microsoft”. Ehdot ovat jokaisen tapahtumakategorian osalta samat (Kuva 6).



Kuva 6. Säännön luomisen rakenne.

Kohdassa yksi luotiin sääntö, joka tallentaa ainoastaan processcreate-tapahtumat, joiden komentorivi sisältää sanan ”whoami”. Tapahtumiin on mahdollista lisätä omia kommentteja, jotka määritellään sääntötiedostossa. Nämä kommentit näkyvät siis tietokoneen tapahtumalokeissa ja ne helpottavat tapahtumien analysointia lokienhallintajärjestelmässä. Kommenteilla kannattaa kertoa lyhyesti ja ytimekkäästi mihin mahdolliseen hyökkäystapaan kyseinen tapahtuma saattaa liittyä. (Rodriguez 2018) Jos komentoriville syötetty ”whoami”-komento ei ole käyttäjän itse tai asianmukaisen sovelluksen syöttämä, saattaa se olla mahdollisen hyökkääjän tapa tiedustella mitä oikeuksia tartutetulla käyttäjällä on toimialueessa. Kommentiksi on määritetty ”System owner discovery” (Kuva 7).

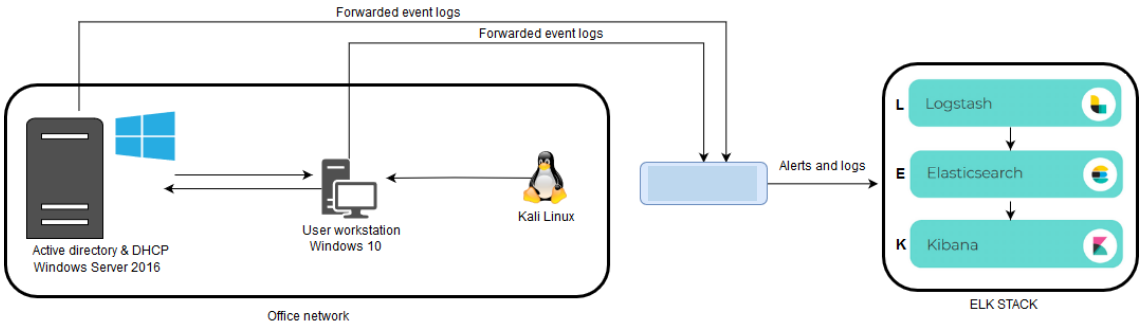


Kuva 7. Esimerkki Sysmonin lokitapahtumasta.

4 HYÖKKÄYKSEN TOTEUTUS

4.1 Testiympäristö

Testiympäristö koostuu virtualisoidusta toimialueesta, johon kuuluu Windows Server 2016 toimialuepalvelin ja Windows 10 työasema. Virtualisointi on toteutettu Oracle Virtualboxilla. Toimialuepalvelin mahdollistaa käyttäjien, sekä tietokoneiden hallinnan keskitetysti isommassa ympäristössä. Palvelin käsittelee käyttäjätietoja ja mahdollistaa kirjautumisen muille toimialueen päätelaitteille. Testauksen helpottamiseksi hyökkäykset suoritetaan kuvitteelliseen toimistoverkkoon sijoitetusta virtuaalikoneesta, johon on asennettu Kali Linux. Toimialuepalvelimen ja työaseman tapahtumalokit tuodaan ELK STACK -kokonaisuudella toteutettuun lokienhallintajärjestelmään. Lokienhallintajärjestelmän rinnalle on asennettu erillinen palvelin, jonka tarkoituksena on vastaanottaa tapahtumalokit palvelimelle ja työasemalle asennetun ohjelman avulla. Erillinen palvelin siirtää tapahtumalokit lokienhallintajärjestelmään, jossa tapahtumat normalisoidaan luettavaan muotoon. Normalisoitujen tapahtumien tarkastelu toteutuu Kibanan selainpohjaisen näkymän kautta (Kuva 8).



Kuva 8. Testiympäristö.

Virtualisoidun toimialueen koko ja toteutus on kärjistetty testausten helpottamiseksi, mutta toimintaperiaatteeltaan se on lähes sama kuin pienissä tai keskisuurissa yrityksissä. Todellisuudessa toimialue koostuisi useammasta työasemasta, tiukoin määritellyistä käyttäjäryhmistä ja epänormaalin liikenteen estävistä palomuurisäännöistä. Vaikka

opinnäytetyössä toteutetuilla hyökkäysmenetelmillä ei välttämättä pääsisi palomuurien tai virustorjuntojen läpi, niin useamman hyökkäystyökalun aiheuttamat lokitapahtumat järjestelmissä eivät muutu. Työn tarkoituksena on tarkastella tapahtumalokeista löytyviä poikkeamia eikä keksiä uutta keinoa ohittaa palomuuri tai virustorjuntaohjelma. Loppuen lopuksi hyökkääjät käyttävät kuitenkin lähestulkoon samoja avoimesti jaettavia työkaluja päästyään yrityksen järjestelmään. Suurin este hyökkäysten toteuttamiselle on ohittaa tietoturvaprotokollat.

4.2 Käytetyt hyökkäystyökalut

Virtualisoidussa hyökkäysalustassa käytetään Debianiin pohjautuvaa jakelupakettia Kali Linuxia. Kali Linux on penetraatiotestaukseen tarkoitettu alusta, jossa tulee mukana noin 600 eri työkalua. Työkaluilla pystytään esimerkiksi keräämään tietoa sekä havaitsemaan haavoittuvuuksia ja käyttämään niitä hyväksi (Kali Linux 2019). Hyökkäyksen viimeisimmässä vaiheessa käytetään esiasennettua Metasploit Framework -laajennusta ja sen PsExec-moduulilla luodaan etäyhteys toimialuepalvelimeen.

Hyökkäysalustaan on asennettu erikseen Pythoniin pohjautuva PoshC2-palvelin. PoshC2 on Nettitude Labsin kehittämä ”command and control” -palvelin, jolla voidaan luoda erilaisia tartuntalähteitä ja komentaa hallintaan saatuja koneita. (Nettitude 2019.)

Metasploit, sekä PoshC2 ovat implementoineet työkaluihinsa erilaisia avoimeen lähdekoodiin perustuvia hyökkäystyökaluja, joista yksi tunnetuimmista on Mimikatz. Mimikatz on Benjamin Delpyn kehittämä hyökkäystyökalu, jonka avulla pystytään lukemaan ja hyväksikäyttämään käyttäjien kirjautumistietoja Windowsin LSASS-prosessista (Local Security Authority Subsystem Service). Kun käyttäjä kirjautuu tietokoneelle, tallentuvat hänen tunnistetietonsa LSASS-prosessin muistiin. Muistiin tallentuneiden tunnistetietojen avulla mahdollistetaan käyttäjän kertakirjautuminen, jolloin ei tarvitse syöttää salasanaa uudestaan esimerkiksi tiedostojakoihin tai sähköpostiin mentäessä (Adsecurity 2018). Tartuttamisen jälkeen hyökkäyksessä käytetään kyseistä Mimikatz-työkalua.

4.3 Toteutetun hyökkäyksen vaiheet

Hyökkäyksen kulussa pyritään jäljittelemään Microsoft ATA Kill Chainin prosessikaavioita. Ulkoinen tiedustelu on jätetty työstä pois, koska sitä ei voida havaita laitteiden paikallisista tapahtumalokeista. Ulkoisen tiedustelun pystyy osittain havaitsemaan palomuurin lokidatan perusteella. Ensimmäisessä vaiheessa suunnitellaan takaoven avaava haitallinen tiedosto, jolla saadaan kohdelaite haltuun. Haitallisena tiedostona käytetään makroja sisältävää Word-dokumenttia. Kolmannessa vaiheessa tiedustellaan ympäristöä Windowsin natiiveja komentoja sisältävällä tiedusteluskriptalla. Neljännessä vaiheessa tartutetusta laitteesta löytyy ylläpitokäyttäjän tunnistetiedot, joita hyväksikäyttämällä voidaan luoda etäyhteys toimialuepalvelimelle. Viimeisessä vaiheessa toimialuepalvelimelle päästäessä saadaan kaikki toimialueen käyttäjät haltuun (Kuva 9).

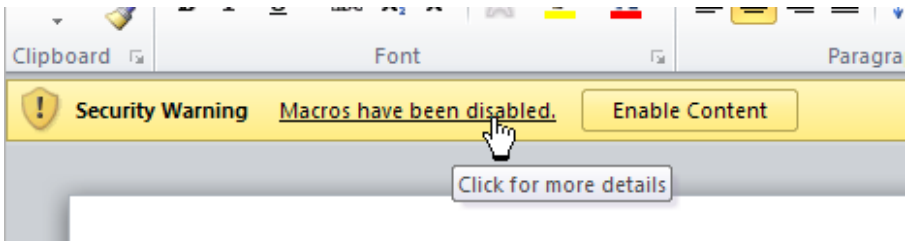
Kuva 9. Toteutetun hyökkäyksen vaiheet.

5 HYÖKKÄYS JA LOKIEN ANALYSOINTI

5.1 Haitallinen Office-dokumentti

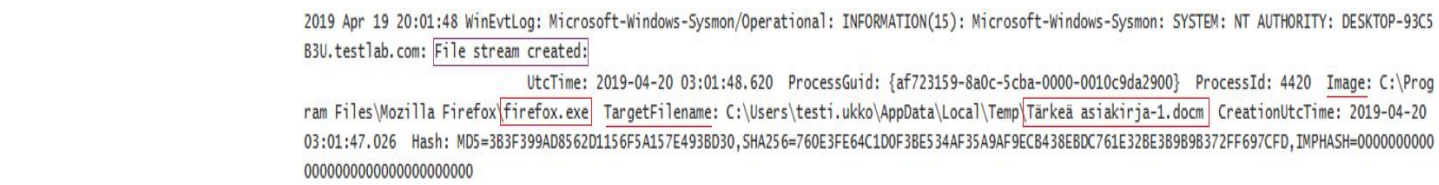
Lähes jokainen meistä on joskus lukenut tai muokannut Microsoftin tarjoamia Office-dokumentteja ja ne ovat hyödyllisiä meille kaikille. Tämä on osa syy, miksi olemme erittäin todennäköisesti avaamassa sähköpostin liitetiedostona vastaanotetun dokumentin, vaikka se ei olisikaan tunnetulta lähettäjältä. Hyökkääjät kuitenkin hyväksikäyttävät ihmisten mielenkiintoa lähettämällä heille makroilla muokattuja dokumentteja, jotka avatessaan suorittavat haitallisia komentoja. Yritysten tietoturvaprotokollat ovat hyvin usein estäneet makrojen suorittamisen dokumenteissa, mutta poikkeuksia löytyy, missä makrojen ajaminen on hyväksytty automaattisesti dokumentin avatessa. Haitallisia makrodokumentteja käytetään edelleen laajasti, sekä yksityishenkilöihin kohdistuvissa massahyökkäyksissä, että yrityksiin kohdennetuissa korkeamman tason hyökkäyksissä. Yksi tunnettu esimerkki on Sofacy-kampanja, jossa hyökkääjien kohteena oli eri maiden ulkoministeriöt (Hayun 2018).

PoshC2-palvelimen mukana tulee laaja valikoima erilaisia tartutusmenetelmiä, jotka mahdollistavat takaoven avaamisen kohdekäyttäjän koneelle. Tässä esimerkissä käytetään valmiiksi luotua makroa, jonka voi upottaa Word-, Excel -tai Powerpoint -dokumenttiin. Makro on luotu siten, että se suorittaa itsensä automaattisesti, jos käyttäjän asetukset sallivat sen. Muussa tapauksessa käyttäjän on hyväksyttävä makron suoritus erikseen painamalla Enable Content -valintaa (Kuva 10).



Kuva 10. Enable Content -valinta.

Haitallinen dokumentti ladattiin Outlookin selainpohjaisesta sähköpostista, jolloin lähdesovelluksena on firefox.exe. Normaalin Word-tiedoston tiedostopääte on docx, mutta makroja sisältävä dokumentti päättyy puolestaan docm-päätteeseen. File stream created -tapahtumasta voidaan havaita, että käyttäjä on ladannut mahdollisesti haitallisen tiedoston (Kuva 11). File stream, eli datavirta on tietoliikenteessä sarja digitaalisesti ohjelmoituja koherentteja signaaleita (paketti dataa tai IP-paketti), jotka laite lähettää tai vastaanottaa (ITS 1996). Selaimet ja sähköpostit käyttävät datavirtaa merkitsemään internetistä tai ulkoisista lähteistä ladattuja tiedostoja.



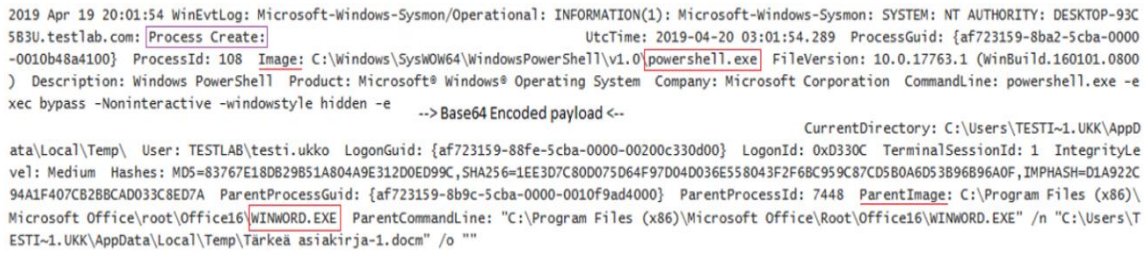
Kuva 11. File stream created.

Muut makroja käyttävät Microsoft Officen sovellukset ja niiden tiedostopäätteet on lueteltu taulukossa (Taulukko 2).

Taulukko 2. Microsoft Officen makroja käyttävät sovellukset.

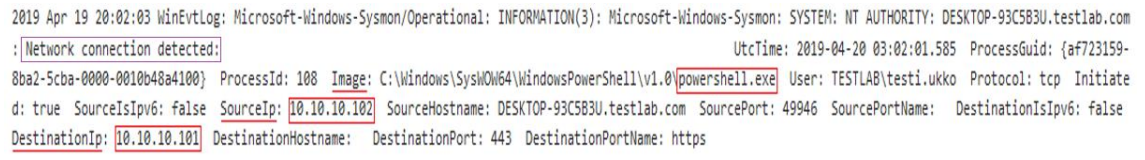
Tiedostotyyppi	Tiedostopääte
Excel Macro-Enabled Workbook	.xlsm
Excel Macro-Enabled Add-In	.xlam
PowerPoint Macro-Enabled template	.potm
PowerPoint Open XML Macro-Enabled Presentation	.pptm
PowerPoint Macro-Enabled slide	.sldm

Käyttäjän avattua makroilla muokatun dokumentin, syntyy process create -tapahtuma. Prosessin avaamisesta syntyvästä tapahtumasta käy ilmi, että Word-ohjelma on avannut Windows Powershellin ja suorittanut sitä kautta dokumenttiin upotetun takaoven lataavan komennon. Lokin commandline-kohdassa näkyvä latauskomento on suoritettu parametreilla execution policy bypass, noninteractive, windowstyle hidden ja encodedcommand. Määritetyt parametrit antavat automaattisesti luvan suorittaa komentoja ja piilottavat avautuvan komentorivi-ikkunan käyttäjältä. Encodedcommand-parametri, joka näkyy pelkkänä e -kirjaimena kääntää koko komennon sisällön Base64-muotoon (Kuva 12).



Kuva 12. Process Create.

Haitallisen komennon suoritettua takaovi on asentunut onnistuneesti ja powershell luo yhteyden PoshC2-palvelimeen, jonka IP-osoite on 10.10.10.101. Takaovi tarkistaa yhteyden hallintapalvelimeen 5 sekunnin välein. Network connection detected -tapahtumista havaitaan, että powershell.exe sovellus luo yhteydenotot kyseiseen IP-osoitteeseen (Kuva 13).



Kuva 13. Network connection detected.

5.2 Prosessin injektointi

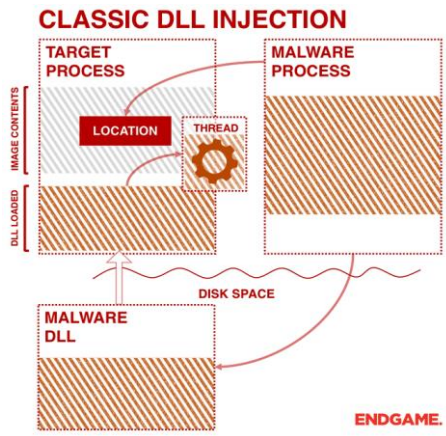
Prosessin injektoimisella tarkoitetaan haitallisen koodin syöttämistä toiseen prosessiin. Tämän avulla voidaan ajaa takaoven kautta syötettyjä komentoja injektoidun prosessin kautta. Uuden prosessin haltuunotto voi mahdollistaa pääsyn prosessin muistiin, järjestelmän tai verkon resursseihin sekä mahdollisesti suurempiin oikeuksiin. Naamioituminen aktiiviseksi ja natiiviksi prosessiksi saattaa myös hidastaa hyökkäyksen havaitsemista. (MITRE 2019.)

Hyökkäyksen tartuntavaiheen jälkeen migratoidaan alkuperäinen takaovi uuteen prosessiin. CreateRemoteThread-tapahtumasta havaitaan, että powershell.exe luo uuden säikeen netsh.exe -tiedostoon (Kuva 14).

```
2019 Apr 20 12:04:16 WinEvtLog: Microsoft-Windows-Sysmon/Operational: INFORMATION(8): Microsoft-Windows-Sysmon: SYSTEM: NT AUTHORITY: DESKTOP-93C5B3U.t
estlab.com: CreateRemoteThread detected:                                    UtcTime: 2019-04-20 19:04:16.109  SourceProcessGuid: {af723159-6c
89-5cbb-0000-0010791a7d00}  SourceProcessId: 6996  SourceImage: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe  TargetProcessGuid: {af723159
-6d30-5cbb-0000-0010bef97e00}  TargetProcessId: 2924  TargetImage: C:\Windows\System32\netsh.exe  NewThreadId: 1628  StartAddress: 0x0000000000010000
StartModule:   StartFunction:
```

Kuva 14. CreateRemoteThread detected.

Haittaohjelma kirjoittaa polun sen dynaamiseen linkkikirjastoon (DLL) toisen prosessin näennäismuistiin (disk space) ja varmistaa, että etäinen prosessi lataa sen luomalla säikeen (thread) kohdeprosessiin (Kuva 15) (Hosseini 2017).

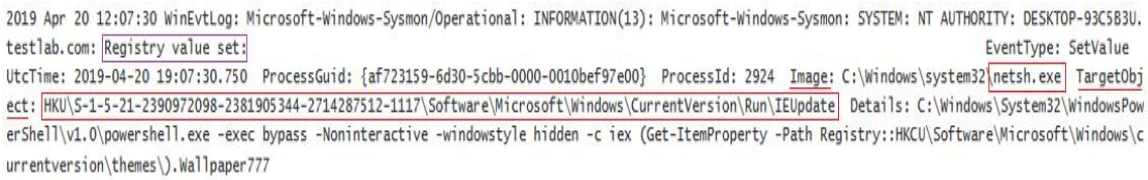


Kuva 15. Remote Thread (Hosseini 2017).

5.3 Persistenssi

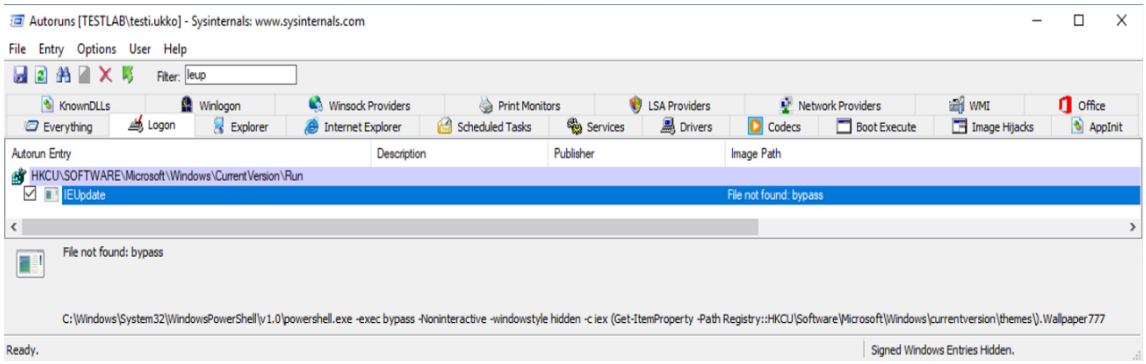
Persistenssi on toiminnan tai ilmiön jatkuminen, vaikka sen aiheuttanut asia on lakannut vaikuttamasta tai se kohtaa esteitä. Hyökkääjien yhtenä tavoitteena on luoda persistenssi takaovi tartutettuun järjestelmään, jotta yhteys kohteeseen säilyisi mahdollisimman pitkään. Persistenssin avulla mahdollistetaan se, että esimerkiksi verkkoyhteyden katketessa, koneen uudelleen käynnistyessä tai sovelluksen sammuessa, tartutettu kone ei häviä kontrollista. Hyökkääjien yhtenä suosimana tekniikkana on ollut asentaa käynnistystiedosto Windowsin rekisteriin, joka avaa haitallisen sovelluksen aina käyttäjän kirjautuessa koneelle. (MITRE 2019.)

PoshC2-palvelimen avulla voidaan asentaa niin sanottu autorun-persistenssi. Persistenssin asennettua lokitapahtumasta havaitaan, että IEUpdate -niminen rekisteriavain on lisätty käyttäjän Software\Microsoft\Windows\currentversion\run\ -rekisterijuureen. PoshC2 asentaa myös erillisen Wallpaper777-nimisen avaimen toiseen juureen, joka sisältää varsinaisen takaoven latauskomennon. Joka kerta sisäänkirjautuessa, IEUpdate käynnistää powershellin ja suorittaa latauskomennon sisältävän avaimen käyttäjän huomaamatta (Kuva 16).



Kuva 16. Registry value set.

Persistenssin asentumisen voi tarkastaa Sysinternalsin kehittämällä autoruns-ohjelmalla (Kuva 17).



Kuva 17. Autoruns-ohjelma.

5.4 Sisäinen tiedustelu

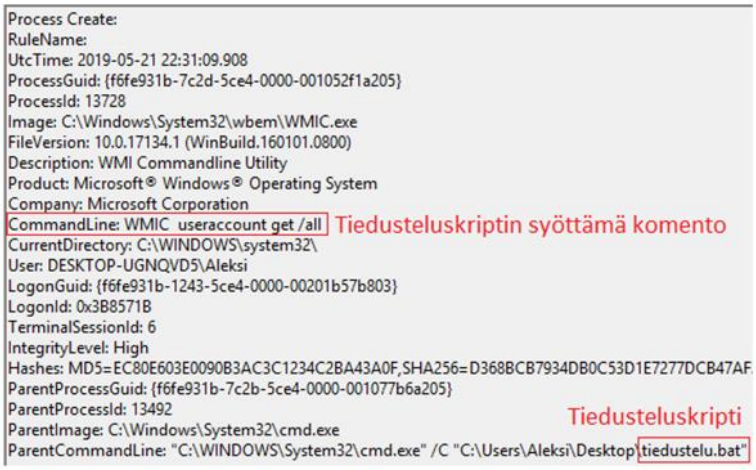
Sisäisessä tiedustelussa hyökkääjä haluaa tietää mahdollisimman paljon yrityksen ympäristöstä, jotta hän voi suunnitella oikean reitin tavoitteeseen. Tiedustelussa käytetään yleensä jo valmiina saatavilla olevia ohjelmia, kuten Nmap tai Bloodhound. Nämä ohjelmat kuitenkin nojautuvat enemmänkin verkkopakettien analysointiin, eikä niitä voida havaita Sysmonin lokitapahtumista. Hyökkääjät suosivat myös Windowsin natiiveja työkaluja, jotka eivät laukaise hälytyksiä virustorjuntaohjelmissa tai aiheuta epäilyksiä lokitapahtumia tarkastellessa. Windowsin natiivit tiedustelukomennot voidaan kerätä yhdeksi skriptiksi ja ajaa ne tartutetulla koneella saadakseen lisää tietoa toimialueesta ja tartutetusta käyttäjästä (Kuva 18).

```
@echo off

::Luo kansio c:\temp
mkdir c:\temp
::Käyttäjän nimi, oikeudet ja ryhmät
whoami /all>>"c:\temp\tiedustelu.txt"
::IP-asetukset
ipconfig /all>>"c:\temp\tiedustelu.txt"
::Aktiiviset TCP-yhteydet
netstat -nao>>"c:\temp\tiedustelu.txt"
::Käynnissä olevat palvelut
net start>>"c:\temp\tiedustelu.txt"
::Käynnissä olevat prosessit
WMIC PROCESS list>>"c:\temp\tiedustelu.txt"
::Järjestelmän käyttäjätilit
WMIC useraccount get /all>>"c:\temp\tiedustelu.txt"
::Järjestelmän pääkäyttäjät
net localgroup administrators>>"c:\temp\tiedustelu.txt"
::Laitteet nykyisessä toimialueessa
net view>>"c:\temp\tiedustelu.txt"
::Toimialueet verkossa
net view /domain>>"c:\temp\tiedustelu.txt"
::Globaalit ryhmät toimialueessa
net group /domain>>"c:\temp\tiedustelu.txt"
::"Domain users" -ryhmän jäsenet
net group "domain users" /domain>>"c:\temp\tiedustelu.txt"
::"Domain admins" -ryhmän jäsenet
net group "domain admins" /domain>>"c:\temp\tiedustelu.txt"
::"Domain controllers" -ryhmän jäsenet
net group "domain controllers" /domain>>"c:\temp\tiedustelu.txt"
::"Exchange domain servers" -ryhmän jäsenet
net group "exchange domain servers" /domain>>"c:\temp\tiedustelu.txt"
::"Exchange servers" -ryhmän jäsenet
net group "exchange servers /domain>>"c:\temp\tiedustelu.txt"
::"Domain computers" -ryhmän jäsenet
net group "domain computers" /domain>>"c:\temp\tiedustelu.txt"
```

Kuva 18. Tiedusteluskripti.

Kun tiedusteluskripti käynnistetään, niin se kerää kaikki saadut tiedot omaan tekstitiedostoon, jonka hyökkääjä voi ladata. Jokaisesta skriptin suorittamasta komennosta syntyy Process Create -lokitapahtuma, jossa nähdään syötetty komento CommandLine-kohdassa (Kuva 19).

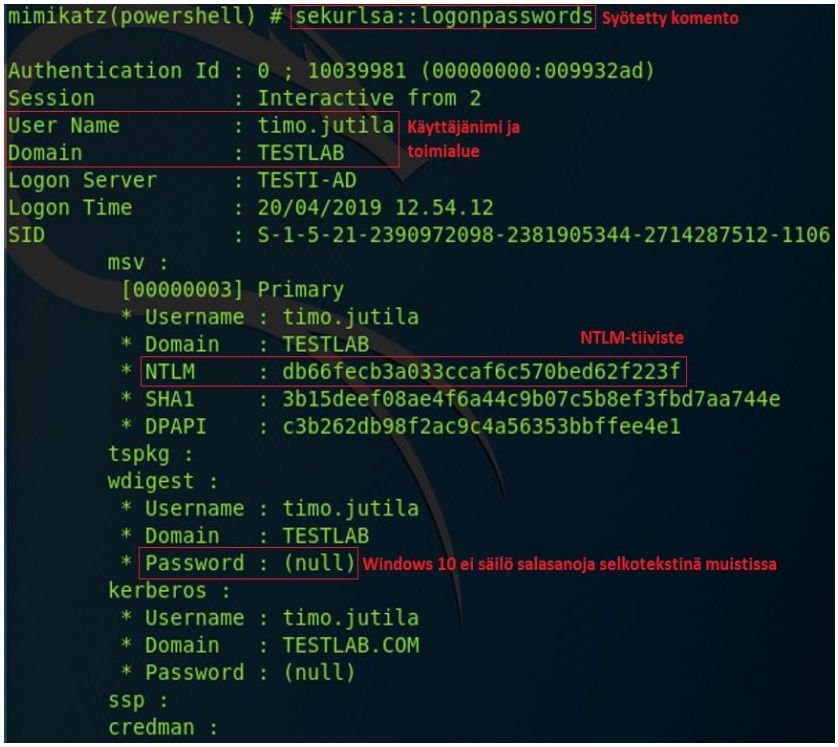


Kuva 19. Skriptin syöttämä komento.

5.5 Kirjautumistietojen etsiminen

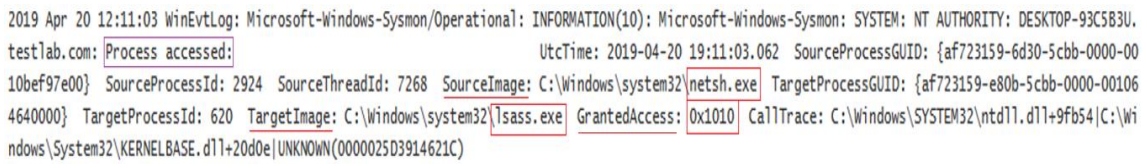
Kirjautumistietojen etsiminen on hyökkäysvaihe, jossa hyökkääjä pyrkii löytämään järjestelmistä käyttäjien salasanoja selkotekstin tai tiivisteen muodossa. Löydettyjä kirjautumistietoja voidaan hyväksikäyttää eri hyökkäystyökaluilla, joiden avulla voidaan liikkua uusiin järjestelmiin ja mahdollisesti löytää tietoa, johon normaalilla käyttäjällä ei ole lukuoikeuksia (MITRE 2019). Kirjautumistietoja pystyy lukemaan suoraan tietokoneen välimuistista esimerkiksi Mimikatzin avulla. Mimikatz on todella suosittu ja tehokas työkalu, koska se perustuu avoimeen lähdekoodiin. Jokainen hyökkääjä voi siis itse muokata sen lähdekoodia ja tämän avulla ohittaa mahdollisesti virustorjunnan tekemät tarkistukset. Myös monet hyökkäysalustat, kuten tässä työssä käytetty PoshC2-palvelin, ovat integroineet Mimikatzin työkaluihinsa.

PoshC2 -palvelimen kautta syötetyllä Mimikatzin komennolla saadaan listattua kaikki koneessa saatavilla olevat kirjautumistiedot. Tähän siis sisältyy kaikkien niiden käyttäjien kirjautumistiedot, jotka ovat kirjautuneena koneelle komennon syöttämishetkellä. Tartutetun kohdekoneen pääkäyttäjän nimi oli ”testi.ukko”. Koneen muistista kuitenkin löytyi käyttäjän ”timo.jutila” kirjautumistiedot, joita voimme hyväksikäyttää seuraavassa hyökkäysvaiheessa. Hyökkäyksen kannalta olennaiset kirjautumistiedot, jotka Mimikatz lukee lsass.exe-prosessin välimuistista ovat käyttäjänimi, toimialue ja NTLM-tiiviste. Windows 7 ja aikaisempien käyttöjärjestelmien välimuistista on mahdollista lukea myös salasanat selkotekstin muodossa. Hyökkäyksen kohdekoneen käyttöjärjestelmänä toimii kuitenkin Windows 10 ja tämän takia salasanaa ei ole mahdollista lukea selkotekstinä (Kuva 20).



Kuva 20. Kirjautumistietojen luku muistista.

Mimikatzin lukiessa kirjautumistiedot lsassin välimuistista ilmenee Process accessed -tapahtuma. Tapahtumasta havaitaan, että netsh.exe (lähdeprosessi) on lukenut onnistuneesti lsass.exe:n (kohdeprosessi) välimuistin oikeuksilla 0x1010 (GrantedAccess) (Kuva 21).



Kuva 21. Process accessed (0x1010).

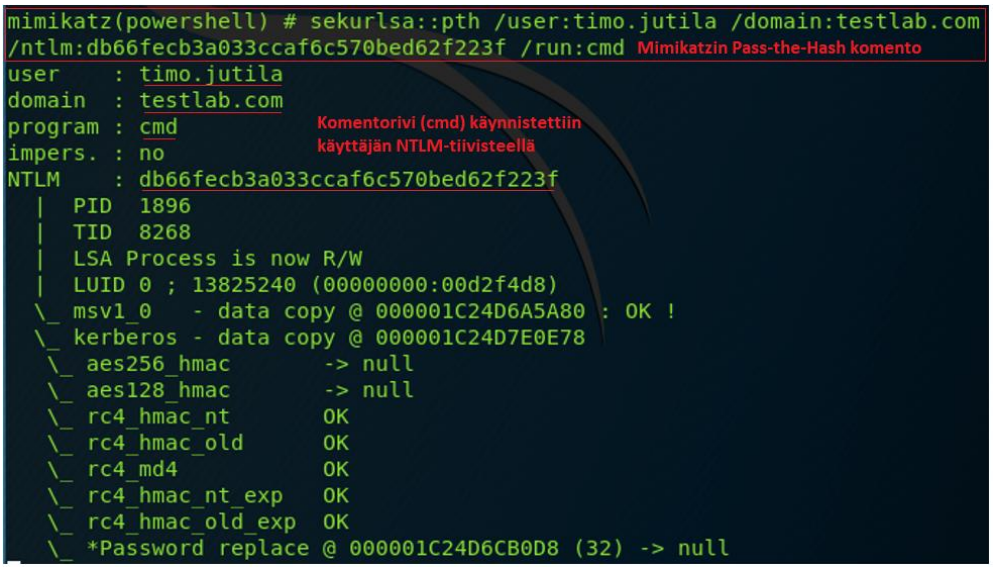
Windowsin tietoturvamalli mahdollistaa prosessien välisten pääsyjen kontrolloimisen. Prosessit tarvitsevat luvan suorittaakseen eri operaatioita keskenään. Lupia on yhteensä 14 kappaletta, joista Mimikatz tarvitsee vain kahta. GrantedAccess-arvo saadaan näistä kahden tarvittavan luvan arvon summasta (Kuva 22) (Rodriguez 2017).

GrantedAcces	Permissions
0x0010	PROCESS_VM_READ
↓	+
0x1000	PROCESS_QUERY_LIMITED_INFORMATION
= 0x1010	

Kuva 22. GrantedAccess-arvo.

5.6 Kirjautumistietojen hyväksikäyttö

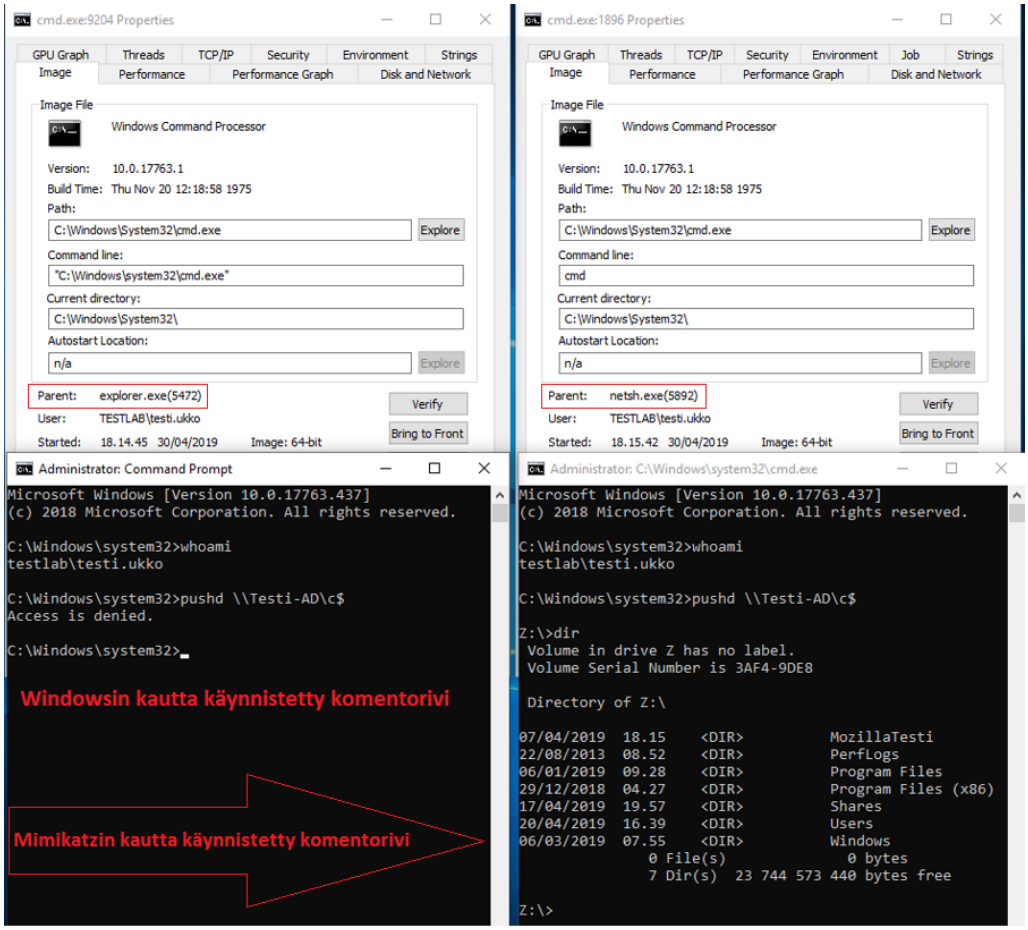
Edellisessä vaiheessa löydetyistä kirjautumistiedoista voidaan hyväksikäyttää käyttäjän ”timo.jutila” NTLM-tiivistettä. Käynnistetään Mimikatzin Pass-the-Hash (pth) -komennolla komentorivi (Kuva 23).



Kuva 23. Pass-the-Hash.

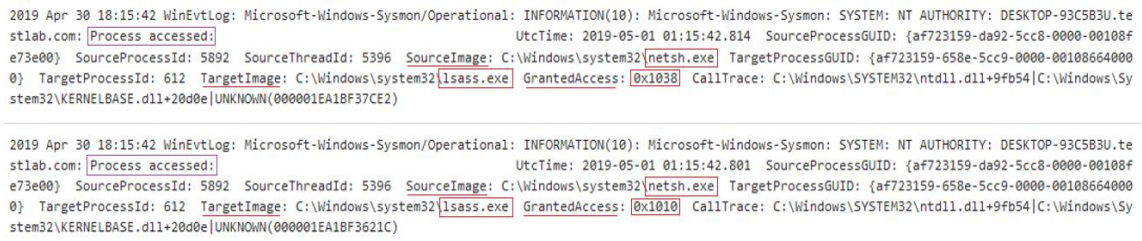
Uusi komentorivi käynnistyy komennossa määritetyn käyttäjän oikeuksilla. Koska käyttäjällä timo.jutila on toimialueen ylläpitäjän oikeudet, niin komentorivillä on mahdollista luoda yhteys toimialuepalvelimeen tai mahdollisesti muihin laitteisiin. Tiivistettä voi myös hyväksikäyttää missä tahansa toimialueen sisällä. Kuvassa on rinnakkain kaksi komen-

toriviä, joista vasen on käynnistetty työpöydän ja oikea Mimikatzin kautta. Taustalla näkyvästä tehtävienhallinnasta nähdään myös, että oikeanpuoleisen komentorivin on käynnistänyt takaovena toimiva sovellus netsh.exe (Kuva 24).



Kuva 24. Käynnistetty komentorivi.

Kun komentorivi käynnistettiin Mimikatzin kautta, niin tapahtumalokiin tulee jälleen Process accessed -tapahtuma. Tällä kertaa tapahtumia on kuitenkin kaksi, joista alempi on täysin sama kuin edellisessä hyökkäyksessä, mutta ylemmässä tapahtumassa takaovemme on saanut puolestaan GrantedAccess 0x1038 -arvon. Tätä voidaan käyttää indikaattorina Pass-the-Hash hyökkäyksen onnistumisesta (Kuva 25).



Kuva 25. Process accessed (0x1038).

Suorittaakseen Pass-the-Hash hyökkäyksen, Mimikatz tarvitsee siis taulukossa 3 ilmoitetut oikeudet, joiden yhteenlaskettu arvo on 0x1038 (Taulukko 3).

Taulukko 3. GrantedAccess-arvo.

GrantedAccess	Permissions
0x1000	PROCESS_QUERY_LIMITED_INFORMATION
0x0010	PROCESS_VM_READ
0x0020	PROCESS_VM_WRITE
0x0008	PROCESS_VM_OPERATION

5.7 Ympäristön hallinta

Tavoiteltaviin resursseihin pääsemistä helpottaa yleensä koko toimialueen käyttäjien hallinta. Hallitakseen koko toimialuetta hyökkääjän on ensiksi saatava aktiivinen etäyhteys toimialuepalvelimeen. Etäyhteyden luomiseksi hyödynnetään Metasploittiin integroitua PsExec-moduulia, jossa voimme hyödyntää kirjautumistietojen etsimisvaiheessa löydettyä käyttäjän NTLM-tiivistettä (Offensive Security 2019). Etäyhteyden saatua suoritetaan hashdump-komento, joka lukee kaikkien toimialueen käyttäjien NTLM-tiivisteet palvelimen muistista (Kuva 26).

```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:ef2664677c70506df9a5658e195ca162:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:0ce5b45a8d74ef727c6a49a26a216355:::
timo.jutila:1106:aad3b435b51404eeaad3b435b51404ee:db66fecb3a033ccaf6c570bed62f223f:::
patrik.laine:1108:aad3b435b51404eeaad3b435b51404ee:20305b4198f476ef0c6dfb40d1ed412e:::
sebastian.aho:1115:aad3b435b51404eeaad3b435b51404ee:69ee725deba8d5e47bf859a3c4c068ca:::
mikko.rantanen:1116:aad3b435b51404eeaad3b435b51404ee:acaea7b56cccf0724583dc200f3face9:::
```

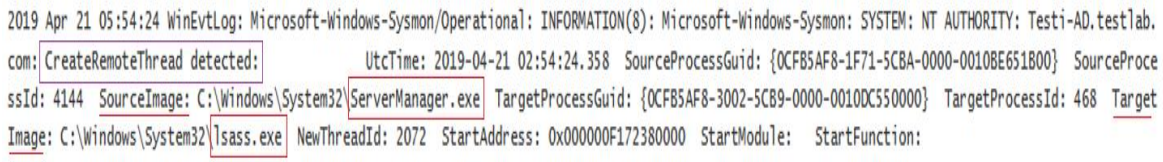
Kuva 26. Toimialueen käyttäjien NTLM-tiivisteet.

Toimialuepalvelimelta löydettyjä NTLM-tiivisteitä voidaan käyttää uusissa Pass-the-Hash hyökkäyksissä. Käyttäjistä tärkein on kuitenkin kuvassa näkyvä krbtgt-käyttäjä. Krbtgt -käyttäjä vastaa toimialueen kerberos TGT-tikettien salaamisesta käyttäjien ja palvelimen välillä. Käyttämällä krbtgt-käyttäjän NTLM-tiivistettä, voimme luoda Mimikatzilla Golden Ticketin, eli väärennetyn TGT-tiketin, jota käyttämällä on mahdollista päästä käsiksi jokaiseen toimialueessa olevaan resurssiin. Väärennetty TGT-tiketti voidaan luoda mille tahansa käyttäjälle vaikka se ei kuuluisikaan toimialueeseen. Naamioidakseen toimintaansa hyökkääjien taktiikkana on yleensä luoda kyseinen tiketti käyttäjälle, joka kuuluu toimialueeseen ja kenellä pitäisikin olla oikeudet tavoiteltavaan resurssiin. Seuraavassa esimerkissä suoritetaan Mimikatzilla Golden Ticket -hyökkäys, jossa väärennetty TGT-tiketti annetaan toimialueeseen kuulumattomalle Kultainentiketti-nimiselle käyttäjälle. Suojaustunnisteena käytetään krbtgt-käyttäjän suojaustunnistetta ja se on määritetty S-parametrissa (Kuva 27).

```
meterpreter > golden_ticket_create -d testlab.com -u Kultainentiketti -s S-1-5-21-2390972098-238
1905344-2714287512 -k 0ce5b45a8d74ef727c6a49a26a216355 -t /root/Downloads/Kultainentiketti.tck
[+] Golden Kerberos ticket written to /root/Downloads/Kultainentiketti.tck
meterpreter > kerberos_ticket_use /root/Downloads/Kultainentiketti.tck
[*] Using Kerberos ticket stored in /root/Downloads/Kultainentiketti.tck, 1884 bytes ...
[+] Kerberos ticket applied successfully.
meterpreter > kerberos_ticket_list
[+] Kerberos tickets found in the current session.
[00000000] - 0x00000017 - rc4_hmac_nt
   Start/End/MaxRenew: 21.4.2019 6:09:59 ; 18.4.2029 14:09:59 ; 18.4.2029 14:09:59
   Server Name       : krbtgt/testlab.com @ testlab.com
   Client Name       : Kultainentiketti @ testlab.com
   Flags 40e00000    : pre_authent ; initial ; renewable ; forwardable ;
```

Kuva 27. Golden Ticket -hyökkäys.

Väärennetyn TGT-tiketin luomisvaiheessa ei ilmentynyt Sysmonin lokitapahtumia. NTLM-tiivisteiden lukeminen toimialuepalvelimella aiheutti kuitenkin tapahtuman, jossa takaovena toimiva isäntäprosessi injektoi haitallista koodia tunnistetietoja käsittelevään lsass -prosessiin (Kuva 28).



Kuva 28. CreateRemoteThread-tapahtuma.

5.8 Havainnoista luotu sääntötiedosto

Työn aikana suoritettujen hyökkäysten perusteella voidaan laatia Sysmonin sääntötiedosto, joka perustuu lokitapahtumista tehtyihin havaintoihin (Kuva 29). Tulkinnan helpottamiseksi sääntötiedostoon on kommentoitu hyökkäysten eri vaiheet. Säännöt ovat rakennettu siten, että ne havaitsevat jokaisen työssä toteutetun hyökkäyksen. Tapahtumakategoriat, joita hyökkäyksen aikana ei ilmennyt, on kokonaan poissuljettu säännöistä. Sääntöjen include-kohdat ovat myös todella suppeat, eivätkä ne kerää tapahtumia annettujen ehtojen ulkopuolelta. Kyseistä sääntötiedostoa ei suositella käytettäväksi tuotannossa, ellei siihen sisällytä lisää ehtoja jokaisen tapahtumakategorian osalta. Sääntötiedoston tarkoituksena on toimia esimerkkinä siitä, miten Sysmonin havainnointikykyä voidaan kehittää laatimalla uusia sääntöjä.

```
<Sysmon schemaversion="4.20">
    <HashAlgorithms>md5,imphash,sha256</HashAlgorithms>
    <EventFiltering>
        <ProcessCreate onmatch="include">
            <!--Haitallinen Office-dokumentti-->
            <ParentImage name="Possible program launched by Office macros" condition="contains">WINWORD.EXE</ParentImage>
            <ParentImage name="Possible program launched by Office macros" condition="contains">POWERPNT.EXE</ParentImage>
            <ParentImage name="Possible program launched by Office macros" condition="contains">EXCEL.EXE</ParentImage>
            <!--Tiedustelukomennot-->
            <CommandLine name="System owner discovery" condition="contains">whoami</CommandLine>
            <CommandLine name="IPCONFIG discovery" condition="contains">ipconfig</CommandLine>
            <CommandLine name="System Network Connections Discovery" condition="contains">netstat</CommandLine>
            <Image name="NET.EXE discovery" condition="contains">net</Image>
            <Image name="WMI Querying" condition="image">wmic.exe</Image>
        </ProcessCreate>
        <FileCreateTime onmatch="include"/>
        <NetworkConnect onmatch="include">
            <!--Takaoven yhteydenotto hyökkääjän koneeseen-->
            <Image name="Powershell network connection" condition="contains">powershell.exe</Image>
        </NetworkConnect>
        <ProcessTerminate onmatch="include"/>
        <DriverLoad onmatch="include"/>
        <ImageLoad onmatch="include"/>
        <CreateRemoteThread onmatch="include">
            <!--Prosessin injektointi-->
            <SourceImage name="Possible powershell code injection" condition="contains">powershell.exe</SourceImage>
            <!--Tunnistetietojen lukeminen toimialuepalvelimelta-->
            <TargetImage name="Possible Hashdump activity" condition="contains">lsass.exe</TargetImage>
        </CreateRemoteThread>
        <RawAccessRead onmatch="include"/>
        <ProcessAccess onmatch="include">
            <!--Tunnistetietojen luku muistista-->
            <GrantedAccess name="Mimikatz logonpasswords lsass access" condition="is">0x1010</GrantedAccess>
            <!--Pass-the-Hash hyökkäys-->
            <GrantedAccess name="Mimikatz Pass-the-Hash lsass access" condition="is">0x1038</GrantedAccess>
        </ProcessAccess>
        <FileCreate onmatch="include"/>
        <RegistryEvent onmatch="include">
            <!--Persistenssi-->
            <TargetObject name="Registry Autorun Keys" condition="contains">\CurrentVersion\Run</TargetObject>
        </RegistryEvent>
        <FileCreateStreamHash onmatch="include">
            <!--Ladatut Office macro -dokumentit-->
            <TargetFilename name="Word macro file downloaded" condition="end with">.docm</TargetFilename>
            <TargetFilename name="Excel macro file downloaded" condition="end with">.xlsm</TargetFilename>
            <TargetFilename name="Excel macro file downloaded" condition="end with">.xlam</TargetFilename>
            <TargetFilename name="PowerPoint macro file downloaded" condition="end with">.potm</TargetFilename>
            <TargetFilename name="PowerPoint macro file downloaded" condition="end with">.pptm</TargetFilename>
            <TargetFilename name="PowerPoint macro file downloaded" condition="end with">.sldm</TargetFilename>
        </FileCreateStreamHash>
        <PipeEvent onmatch="include"/>
        <WmiEvent onmatch="include"/>
    </EventFiltering>
</Sysmon>
```

Kuva 29. Hyökkäyksen perusteella koottu sääntötiedosto.

6 POHDINTA JA TULOKSET

Yrityksiin kohdistuvat hyökkäykset ovat jokainen omalla tavallaan erilaisia ja siksi niistä on hankalaa esittää yksiselitteistä hyökkäysketjua. Microsoftin luoma ATA Killchain kuitenkin auttoi osittain työssä tehtyjen hyökkäysten eri vaiheiden suunnittelussa ja toteutuksessa. Kaikkia hyökkäyksen prosessikaavion vaiheita ei esitetty työssä, koska virtualisoitu hyökkäysympäristö vie liikaa tietokoneen resursseja. Esimerkiksi yhden päätelaitteen sijaan niitä olisi voinut olla enemmän, jotta sisäinen tiedustelu- ja leviämisvaihe olisi saatu toteutettua paremmin.

Hyökkäyksessä käytetyt työkalut ja menetelmät kattavat vain murto-osan niistä, joita hyökkääjien tiedetään käyttävän. Käytetyt työkalut ja menetelmät ovat kuitenkin erittäin suosittuja ja hyökkääjät saattavat käyttää niitä jossain vaiheessa hyökkäystään. Työn tarkoituksena ei ollut löytää jokaista mahdollista hyökkäystekniikan indikaattoria, vaan antaa yleiskuva Sysmonin toiminnasta ja demonstroida, miten sen lokitiedoista voidaan havaita yleisimmin käytettyjä hyökkäysmenetelmiä.

Sysmonin havainnointikykyä on helppo kehittää myös ilman fyysistä testausta. Internetissä on tarjolla todella paljon dokumentointia eri tietomurroista, hakkeriryhmistä ja haittaohjelmista. Nämä dokumentoinnit sisältävät paljon teknistä analyysiä hyökkäysten kulusta ja toiminnasta, joiden pohjalta voidaan luoda uusia sääntöjä laajentaakseen hyökkäyspinta-alan havainnointikykyä.

Myös tässä työssä lähteenä käytetty MITRE ATT&CK -tietosivusto, joka toimii ikään kuin kyberuhkien Wikipediana, auttoi hyökkäysten analysoinnissa ja tarjosi kattavan määrän lähteitä. Jatkokehitysideana havainnointikyvyn parantamiseksi on luoda MITRE:n pohjalta hyökkääjien taktiikoihin ja tekniikoihin perustuva matriisi, jota voidaan käyttää apuna Sysmonin sääntöjen kehittämisessä.

Hyökkäyksissä käytetyt moduulit olivat PoshC2, Metasploit ja mimikatz. Alkutartunnan vektoriksi valittiin edelleen hyökkääjien suosiossa oleva sähköpostin kautta vastaanotettu haitallinen Microsoft Office –dokumentti. Jalansijan saatua järjestelmästä, varmistettiin sen pysyvyys persistenssillä takaovella. Toimintaa naamioitiin injektoimalla takaovi toiseen sovellukseen. Sisäisen tiedustelun tuloksena saatiin haltuun toimialueen ylläpitäjän käyttäjä, jolla vaarannettiin koko toimialue.

Jokaisesta hyökkäysvaiheesta ilmeni poikkeamia Sysmonin tapahtumalokeissa. Työn tavoitteena oli lyhentää hyökkäysten havaitsemiseen kuluvaa aikaa. Sysmonia ja työssä laadittua sääntötiedostoa käyttämällä havaittaisiin tyypillinen hyökkäys jo ensi minuuteilla käyttäjän avatessa haitallisen Office-dokumentin. Koska hyökkäys saatettiin alusta loppuun ja sen toteutuksessa noudatettiin Microsoftin laatimaa prosessikaaviota tunnetuista hyökkäyksistä, voidaan myös osittain havaita hyökkäysmenetelmiä muissakin kuin alkutartunnan vaiheissa.

Sysmonin havainnointikykyä testatessa on hyvä muistaa, että indikaattoreiden pitää olla toistettavissa. Hyökkäykset on toistettava ja tuloksia on verrattava keskenään, jotta voidaan olla varmoja löydetyn poikkeaman luotettavuudesta. Sysmon implementoituna lokienhallintajärjestelmään ei ole myöskään ainut ratkaisu uhkien estämiseen ja on muistettava, että suurin rooli tietoturvassa on ajan tasalla olevat, sekä hyvin suunnitellut tietojärjestelmät. Työntekijöiden kouluttamisella voidaan kanssa pitkälti ehkäistä syntyviä uhkia. Keskitetty lokienhallinta ja Sysmon ovat kuitenkin tärkeä osa koko tietoturvakokonaisuutta, jonka avulla voidaan havaita mahdollisesti syntyneitä uhkia.

LÄHTEET

Adsecurity 2018. Unofficial Guide to Mimikatz & Command Reference. Viitattu 26.4.2019. https://adsecurity.org/?page_id=1821

Hayun, L. 2018. Office Documents Can Be Dangerous (But We'll Continue to Use Them Anyway). Viitattu 27.4.2019. https://unit42.paloaltonetworks.com/unit42-threat-brief-office-documents-can-dangerous-well-continue-use-anyway/

Hosseini, A. 2017. Ten Process Injection Techniques: A Technical Survey of Common and Trending Process Injection Techniques. Viitattu 28.4.2019. https://www.endgame.com/blog/technical-blog/ten-process-injection-techniques-technical-survey-common-and-trending-process

ITS 1996. data stream. Viitattu 27.4.2019 https://www.its.bldrdoc.gov/fs-1037/dir-010/_1451.htm

Kali Linux 2019. Kali Linux Tools Listing. Viitattu 25.4.2019. https://tools.kali.org/tools-listing

Microsoft 2016. Disrupting the kill chain. Viitattu 26.4.2019. https://www.microsoft.com/security/blog/2016/11/28/disrupting-the-kill-chain/

MITRE 2019. Credential Dumping. Viitattu 29.4.2019. https://attack.mitre.org/techniques/T1003/

MITRE 2019. Process Injection. Viitattu 28.4.2019. https://attack.mitre.org/techniques/T1055/

MITRE 2019. Registry Run Keys / Startup Folder. Viitattu 28.4.2019. https://attack.mitre.org/techniques/T1060/

Nettitude 2019. Python server for PoshC2. Viitattu 26.4.2019. https://labs.nettitude.com/blog/python-server-for-poshc2/

Rodriguez, R. 2018. Categorizing and Enriching Security Events in an ELK with the Help of Sysmon and ATT&CK. Viitattu 25.4.2019. https://cyberwardog.blogspot.com/2018/07/categorizing-and-enriching-security.html

Rodriguez, R. 2017. Hunting for In-Memory Mimikatz with Sysmon and ELK – PART II. Viitattu 30.4.2019. https://cyberwardog.blogspot.com/2017/03/chronicles-of-threat-hunter-hunting-for_22.html

Rodriguez, R. 2017. Hunting for In-Memory Mimikatz with Sysmon and ELK – PART III. Viitattu 30.4.2019. https://cyberwardog.blogspot.com/2017/04/chronicles-of-threat-hunter-hunting-for.html

Symantec 2018. Internet Security Threat Report. Viitattu 6.4.2019. https://www.symantec.com/content/dam/symantec/docs/reports/istr-23-2018-en.pdf

Velazquez, C. 2019. Detecting and Preventing Attacks Earlier in the Kill Chain. Viitattu 26.4.2019. https://www.sans.org/reading-room/whitepapers/infosec/detecting-preventing-attacks-earlier-kill-chain-36230

Verizon 2018. Data Breach Investigations Report. Viitattu 6.4.2019. https://enterprise.verizon.com/resources/reports/DBIR_2018_Report_execsummary.pdf

Yung, P. 2017. Security Notification for CCleaner v5.33.6162 and CCleaner Cloud v1.07.3191 for 32-bit Windows users. Viitattu 26.4.2019. https://www.cclearner.com/news/blog/2017/9/18/security-notification-for-ccleaner-v5336162-and-ccleaner-cloud-v1073191-for-32-bit-windows-users