Tampereen ammattikorkeakoulu

C2-infrastruktuurin rakennus ja tietoturvatestaus

C2-Framework and Penetration Testing

Jaakko Oja

OPINNÄYTETYÖ
Syyskuu 2025

Tietotekniikka
Tietoliikennetekniikka ja tietoverkot

TIIVISTELMÄ

Tampereen Ammattikorkeakoulu
Tietotekniikka
Tietoliikennetekniikka ja tietoverkot

Oja Jaakko:
C2-infrastruktuurin rakennus ja tietoturvatestaus
C2-Framework and Penetration Testing

Opinnäytetyö 33 sivua, joista liitteitä 2 sivua
Elokuu 2025

Opinnäytetyön tavoitteena oli rakentaa toimiva C2-infrastruktuuri, perehtyä tarkemmin tämän toimintaan ja suoritella erilaisia hyökkäyssimulaatioita virtuaalikoneilla. Tavoitteena oli myös tutkia, millä tavalla C2:n aiheuttama tietoliikenne näyttäytyy esimerkiksi Wireshark-työkalua apuna käyttäen.

C2-infrastruktuurit ovat keskeinen osa modernia kyberturvallisuutta, sillä niiden avulla voidaan ymmärtää, miten hyökkääjät hallitsevat ja ohjaavat vaarantuneita järjestelmiä. Rakentamalla oman infrastruktuurin ja suorittamalla hallittuja hyökkäyssimulaatioita työ tarjosi käytännön näkökulman siihen, miten komentokanavat toimivat ja millaista tietoliikennettä ne tuottavat. Tämän kautta saatiin arvokasta tietoa siitä, miten tällainen toiminta voidaan havaita ja estää jo varhaisessa vaiheessa.

Lopputuloksena syntyi toimiva C2-infrastruktuuri käyttäen Mythiciä ja sen eri moduuleja. Työssä suunniteltiin ja toteutettiin C2-järjestelmän arkkitehtuuri alusta alkaen virallisen ohjeistuksen mukaisesti, minkä jälkeen suoritettiin erilaisia hyökkäyksiä Windows- ja Linux-koneille virtuaalimaailmassa. Näiden hyökkäysten aikana kerättiin ja analysoitiin syntyvää tietoliikennettä, mikä mahdollisti C2-liikenteen toimintaperiaatteiden syvällisen ymmärtämisen.

Viimeisimpänä tehtiin pienimuotoista havainto- ja suojautumisraportointia, eli kirjattiin ylös erilaisia asioita, miten C2-infrastruktuuria vastaan olisi hyvä suojautua. Työssä myös pohdittiin tietoturvan merkitystä omassa osiossaan.

Asiasanat: eettinen hakkerointi, tietoturva, hakkeri, hyökkäyskuorma, hyökkäyssimulaatio, penetraatiotestaus

ABSTRACT

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

Oja Jaakko,
C2-infrastruktuurin rakennus ja tietoturvatestaus
C2-Framework and Penetration Testing

Bachelor's thesis 35 pages, appendices 2 pages
August 2025

The aim of this thesis was to build a functional C2 infrastructure, examine its operation, and carry out attack simulations in virtual machines. Another objective was to study how the network traffic generated by the C2 infrastructure appears when analyzed using tools such as Wireshark.

C2 infrastructures are an essential part of modern cybersecurity, making it possible to understand how attackers control compromised systems. By building a dedicated infrastructure and performing attack simulations, this study provided a practical perspective on how command channels operate and what kind of network traffic they generate. Through this process, valuable information was obtained on how such activity can be detected and prevented early.

A functional C2 infrastructure was created using Mythic and its modules. The thesis involved designing the C2 system architecture according to official documentation, after which attacks were conducted on Windows and Linux virtual machines. The resulting network traffic was collected and analyzed, enabling a deeper understanding of C2 communication principles.

Finally, a small-scale observation and defense report was compiled, recording ways to protect against C2 infrastructures. The thesis also included a section reflecting on the importance of cybersecurity.

Key words: ethical hacking, cybersecurity, hacker, payload, penetration testing

SISÄLLYS

LYHENTEET JA TERMIT

TAMK	Tampereen ammattikorkeakoulu
op	opintopiste
C2	Command and Control
VM	Virtuaalikone
Hyökkäyskuorma	Mythicissä kahden komponentin, agentin ja C2-profiilin yhdistelmä
SSL	Englanninkielinen termi “Secure Sockets Layer”, eli salausprotokolla
MITRE ATT&CK	Maailmanlaajuisesti käytetty tietoturvaviitekehys
DDOS	Palvelunestohyökkäys
Proxy	Välityspalvelin
Wireshark	Verkkoliikenteen analysointityökalu
APT	”Advanced persistent threat ” eli kehittynyt jatkuva uhka
Kontti	virtuaalinen ajoympäristö
GUI	Käyttöliittymä
Obfuskointi	Koodin tai tiedoston tarkoituksellinen monimutkaistaminen
EDR	Engl. “Endpoint Detection and Response”, suomeksi päätelaitteiden havaitsemis- ja reagointiratkaisut

1. JOHDANTO

Opinnäytetyön tarkoituksena oli perehtyä tarkemmin C2-servereihin, eli tuttavallisemmin engl. "Command and Control" -järjestelmiin: selvittää niiden rakennetta, tyypillisiä hyökkäyskuvioita ja toimintamalleja sekä tutkia keinoja suojautua niitä vastaan. Työssä käydään myös läpi erilaisia red teaming -tekniikoita C2-palvelimen ollessa toimintavalmiina, ja pyritään toimimaan mahdollisimman huomaamattomasti.

Tässä opinnäytetyössä C2-ohjelmaksi valittiin Mythic, joka itsessään on käytännöllinen vaihtoehto rakentaa alusta alkaen virallisen wikiohjeistuksen mukaisesti. Mythic tarjoaa myös helpon käyttöliittymän, joka myös on mieluisa käyttää visuaalisesti ja auttaa selventämään kommunikointia palvelimen ja kohdekoneen välillä.

Työssä tarkkailtiin verkossa tapahtuvaa tietoliikennettä kohdekoneen ja Mythicin välillä käyttämällä yleisimpiä työkaluja mm. Wiresharkkia joka mahdollistaa tietoliikenteen tarkkailun hyvinkin tarkalla tasolla. Lopuksi käsitellään tietoturvan merkitystä.

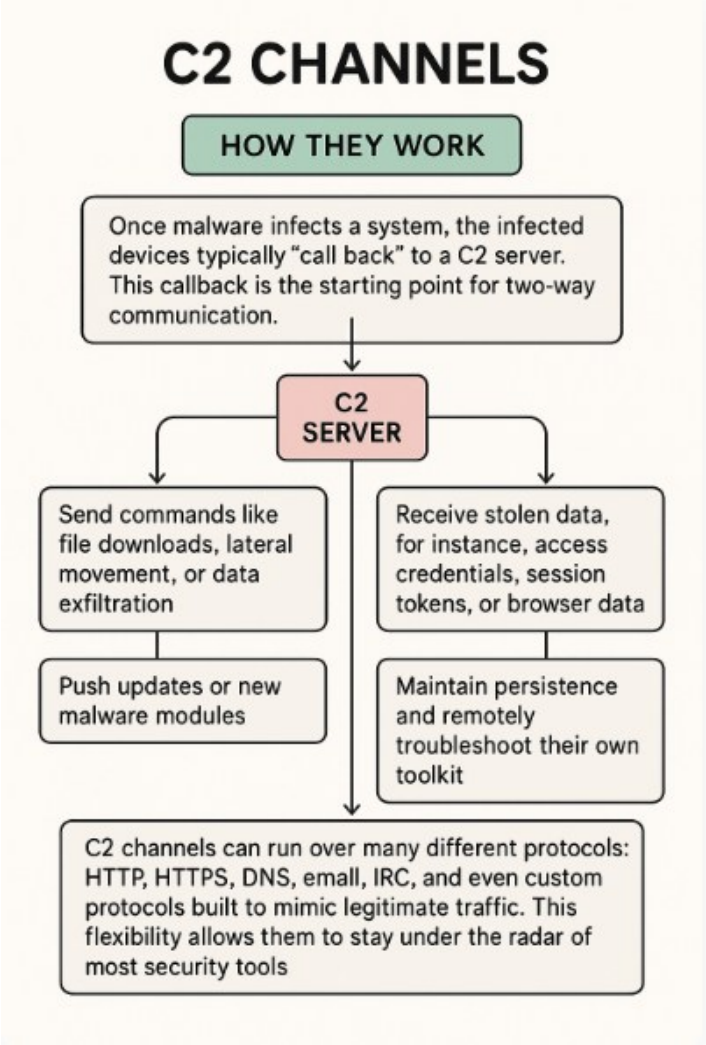
2. C2-INFRASTRUKTUURI

Opinnäytetyössä käsitellään C2:sta eli “Command and Control”-infrastruktuuria yhden ohjelmiston kautta. Halutaan kumminkin selventää vähän termiä *C2* ja mitä kaikkea tämän taakse kätkeytyy.

2.1. C2-järjestelmien rakenne ja toimintatavat

C2 eli englanniksi “Command and Control” tarkoittaa kyberturvallisuudessa hyökkääjän käyttämää komentokanavaa ja hallintainfrastruktuuria. Tämän avulla voidaan ohjata ja hyödyntää vaarantuneita järjestelmiä etäyhteyden kautta usein käyttämällä salattua yhteyttä. (MITRE ATT&CK 2018).

Kohdejärjestelmä voi vaarantua monin eri tavoin, esimerkiksi tietojenkalastelusähköpostien kautta, lataamalla C2:sen tartuttamia tiedostoja tai klikkaamalla internetissä haitallisia linkkejä/sivustoja, jotka asentavat järjestelmään haittaohjelmia. Kun kohteena oleva järjestelmä on vaarantunut, hyökkääjä saa siihen yhteyden ja voi käyttää C2-serveriä antaakseen komentoja tai toteuttaa muita haluttuja hyökkäyksen tavoitteisiin liittyviä toimia, esimerkiksi varastaa arkaluonteisia tietoja (MITRE ATT&CK 2018). Kuvassa 1 on havainnollistettu C2-kanavien toimintaa.



KUVA 1. C2-kanavat ja miten ne toimivat (Lähde: Hunt 2025).

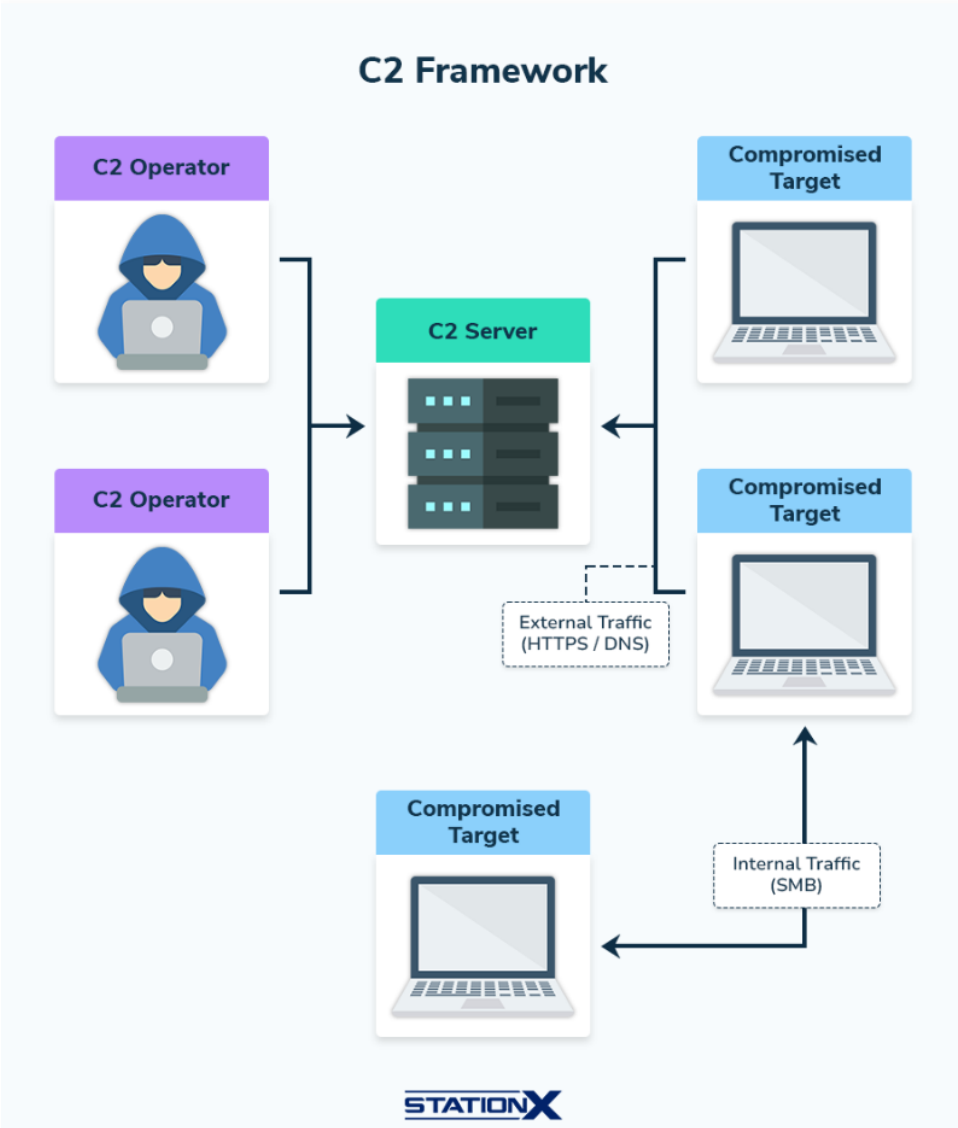
2.2. Käyttötarkoitukset ja hyödyt

Näiden järjestelmien avulla voidaan muun muassa:

- Lähettää komentoja haittaohjelmaa käyttämällä saastutettuihin laitteisiin,
- Kerätä tietoja kohteena olevista järjestelmistä,
- Koordinoida ja suorittaa laajoja hyökkäyksiä, esimerkiksi palvelunesto-hyökkäyksiä (engl. Distributed Denial of Service eli lyhyemmin DDOS).

C2-infrastruktuurit on tarkoitettu olemaan *client – server* malli, jolloin tarvittaessa useammat käyttäjät (operaattorit) voivat yhdistyä keskitetylle C2-serverille selkeän käyttöliittymän kautta. Tämä mahdollistaa useamman operaattorin

työskentelyn samanaikaisesti saman kohteen parissa kuten kuvassa 2 on esitetty (Lähde: StationX 2024).



KUVA 2. StationX-sivustolta otettu kuva, joka havainnollistaa C2-infrastruktuuria ja tiedonsiirtoa operaattorin ja kohteen välillä (Lähde: StationX 2024).

Arkkitehtuurin etuina ovat muun muassa se, että kaikki data saadaan tarvittaessa yhdestä keskitetystä lähteestä jokaiselle operaattorille ja että ulospäin näkyy vain yksi julkinen IP-osoite tai verkkotunnus (domain). Tämä yksinkertaistaa hallintaa ja helpottaa C2-infrastruktuurista lähtevän liikenteen naamiointia.

Muita hyötyjä hyökkäysnäkökulmasta on monia:

- Huomaamattomuus: voivat salata tiedonsiirron SSL-sertifikaateilla, naamioida haitallisen liikenteen lailliseksi ja piilottaa hyökkäävän puolen IP-osoitteet C2-välityspalvelimilla.

- Joustavuus: Agenttien konfigurointi ja uusien hyökkäystekniikoiden kehittäminen tarpeen mukaan.
- Skaalautuvuus: Yksi C2-serveri voi hallita satoja kaapattuja kohdejärjestelmiä. Pilvipalveluiden ansiosta tämä luku voi olla vieläkin isompi.
- Pysyvyys: Kaapatussa kohdejärjestelmässä pysyminen säilyy myös järjestelmän uudelleenkäynnistyksen tai suojaustoimien jälkeen.

Tämä arkkitehtuurinen toteutus varmistaa sen, että C2-infrastruktuuri on hyvin stabiili. Samalla se parantaa myös tietoturvaa, sillä erilliset osat voidaan suojata ja valvoa tarkemmin (NIST 2020).

2.3. Käyttäjät ja tunnetut infrastruktuurit

C2-infrastruktuurit ovat yleisessä käytössä muun muassa seuraavilla ryhmittymillä: kyberrikolliset ja hakkeriryhmät, APT-ryhmät sekä hyökkäävän tiimin (engl. “red teaming”), eli penetraatiotestaajat (MITRE ATT&CK 2018).

Tunnettuja ja yleisimmin käytössä olevia C2-infrastruktuureita ovat muun muassa Havoc, Cobalt Strike, Sliver, PowerShell Empire ja Mythic (Cobalt Strike n.d.; BishopFox/Sliver 2025; Mythic 2025; Havoc 2024; Empire 2020)

2.4. C2-infrastruktuurien vertailu

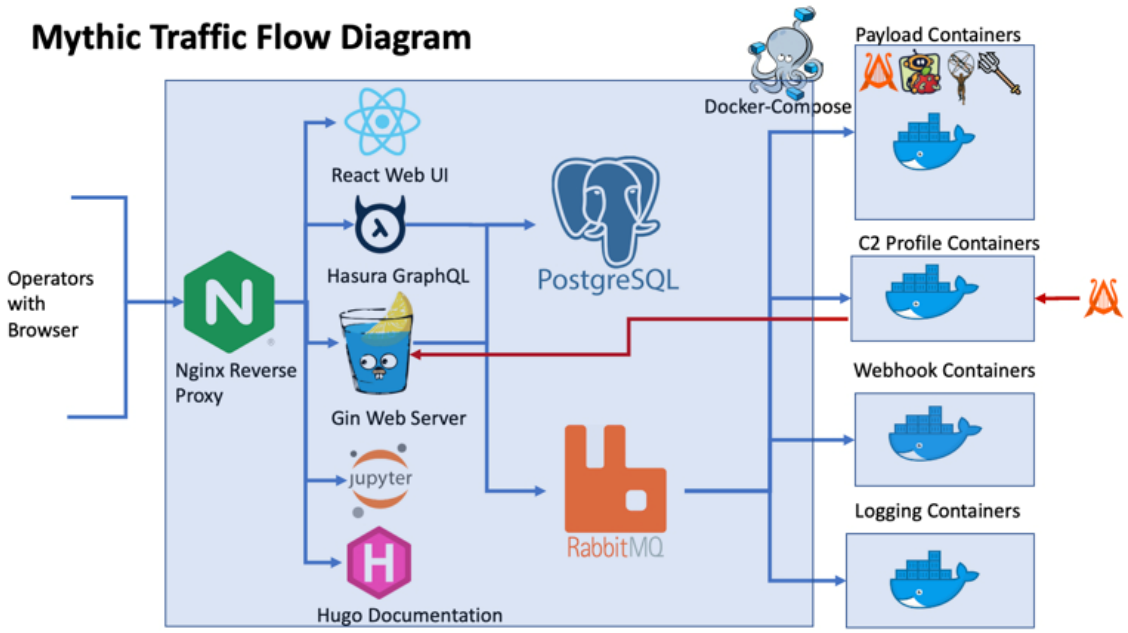
C2-infrastruktuurien vertailuun voidaan hyödyntää C2 Matrix -nimistä projektia, jonka yhteydessä on julkaistu laaja Google Sheets -taulukko. Se esittelee eri C2-alustoja ja niiden ominaisuuksia kattavasti vertailumuodossa (C2 Matrix 2024). C2-infrastruktuurille ei ole olemassa yksiselitteisesti parasta, huonointa, oikeaa tai väärää ratkaisua – tärkeintä on valita tarkoitukseen ja tavoitteeseen parhaiten sopiva vaihtoehto.

3. MYTHIC C2

3.1. Mikä on Mythic?

Mythic on modulaarinen, monikäyttäjäinen komentojärjestelmäalusta, joka on suunniteltu hyökkääviin operaatioihin tai “red-teaming”-penetraatiotestauksiin. Sen tavoitteina on operaattorien työn helpotus, agenttien ylläpito ja mukautusmahdollisuudet halutulla tavalla.

Mythic hyödyntää verkkopohjaista käyttöliittymää, joka on kirjoitettu React-ohjelmointikielellä, sekä Docker-kontteja koko taustajärjestelmän sujuvaan hallintaan. Sen palvelin on kirjoitettu Go-ohjelmointikielellä ja palvelin käsittelee aika pitkälti kaikki verkkopyynnöt GraphQL-rajapinnan ja WebSocket-yhteyksien kautta. Serveri myös hallinnoi yhteydet PostgreSQL-tietokantaan ja kommunikoi käynnissä olevien Docker-konttien kanssa RabbitMQ-välityksellä. Tämänkaltainen iso ja modulaarinen arkkitehtuuri mahdollistaa sujuvan yhteydenpidon konttien välillä, sekä myös sen, että eri Mythicin komponentit voidaan sujuvasti sijoitella esimerkiksi eri koneille, oli ne sitten virtuaalisia tai fyysisiä. Kuvassa 3 on selkeämpi kaavio Mythicin modulaarisesta kaaviosta.



KUVA 3. Mythicin tietoliikennevirtauskaavio (Lähde. Mythic 2025)

3.1.1. Perustelut Mythicin käytölle

Mythic on ennenkaikkea työkalu, ei haittaohjelma (englanniksi ”malware”). Se tarjoaa kattavan, nykyaikaisen ja älykkään tavan suorittaa erilaisia hyökkäyssimulaatioita. Sen vahvuutena on juuri tiedon ja datan yhdistäminen, seuranta ja analysointi osana jokapäiväistä työskentelynä.

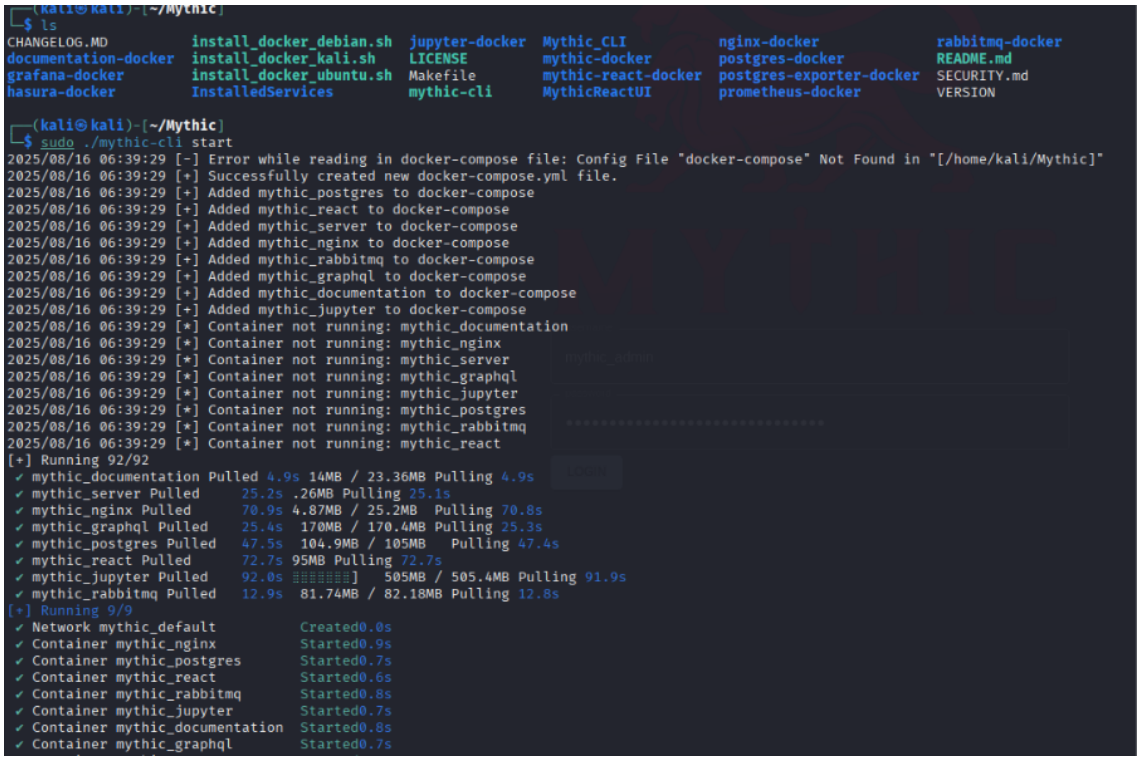
Tämä työkalu auttaa myös operatiivisen turvallisuuden hallinnassa, tuttavallisemmin OpSec. Mythic integroi MITRE ATT&CK-kehystä osana työskentelyä. Jokainen annettu komento voidaan yhdistää ATT&CK-tekniikoihin ja operaattori voi nähdä mitä tekniikoita komennot kattavat ja mitä on esimerkiksi käytetty.

Mythicin etuna on sen käyttöliittymä, joka mahdollistaa koko operaatioiden ja simulaatioiden läpikäynnin yhdessä paikassa. Tämä tuo helpotusta tilanteissa, joissa on esimerkiksi iso tiimi työskentelemässä saman toimeksiannon kanssa: kaikki komennot, kommentit ja raportoinnit ovat sujuvasti yhdessä paikassa.

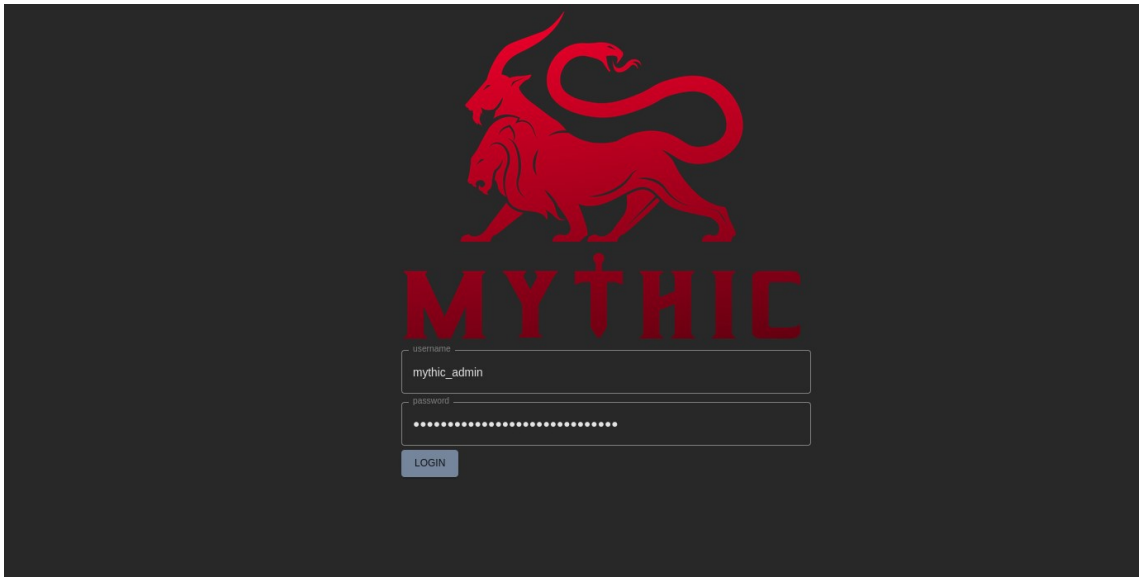
3.2. Asennusympäristöt ja tarvittavat ominaisuudet

Kali Linux on sopiva jakelu penetraatiotestaukseen ja eettiseen hakkerointiin. Se sisältää runsaasti työkaluja, jotka kattavat monenlaisia hyökkäys- ja puolustustekniikoita(Kali Linux 2025). Tämä laaja työkalupaketti oli syynä siihen, miksi Mythic haluttiin asentaa juuri Kalille. Asennus suoritettiin Mythicin virallisen dokumentaation mukaisesti Kalin virtuaalikoneelle.

Kuten aiemmin on todettu Mythicin käyttävän hyvin pitkälti Dockerin kontteja käynnistyäkseen ja toimiakseen oikein, oli asennuksessa ensiarvoisen tärkeää saada nämä kontit käynnistymään oikein. Asennusvaiheessa huomattiin, että jos Mythiciä ei sulje oikein, sen käyttämät Docker-kontit menevät jumiin. Kuvassa 4 tämä käynnistymisprosessi on käynnissä ja kuvassa 5 on Mythicin Web-käyttöliittymä.



KUVA 4. Tässä kuvakaappauksessa Mythicille annettu käynnistyskäsky ja nähdään kuinka Docker käynnistää tarvittavat kontit (Kuva: Jaakko Oja).



KUVA 5. Kuvakaappaus Mythicin Web-käyttöliittymän (GUI, eli englanniksi “graphical user interface”) kirjautumisikkunasta. Tunnukset löytyvät komennolla *cat .env*, johon Mythic ensimmäisellä käynnistyskerralla luo tunnukset (Kuva: Jaakko Oja).

3.2.1. Mythicin agentit ja C2-profiilit

C2-serverit eivät ole yksinään käyttökelpoisia ilman niihin liitettyjä agentteja tai C2-profiileja. Mutta mitä nämä sitten ovat?

Mythicillä (ja yleisesti kaikilla C2-servereillä) agentit toimivat operaatioiden käytännön toteuttajina kohdejärjestelmissä. Jokaisella agentilla on omat ominaisuutensa, tuetut käyttöjärjestelmät ja viestintäkanavat (Mythic 2025; MITRE ATT&CK 2018)

C2-profiilit määrittävät, miten agentit kommunikoivat Mythicin kanssa. Jokainen C2-profiili pyörii omassa Docker-kontissa, joko Python tai Go-palveluna. Ne yhdistyvät Mythicin käyttämään RabbitMQ-viestinvälittäjään vastaanottaakseen tehtäviä. Profiileilla on kaksi pääosaa:

- Palvelinkoodi – Ajetaan Docker-kontissa ja muuntaa profiilin viestintätavan (esimerkiksi Slack, WebSocket, http tai https) Mythicin käyttämiksi REST-rajapinnoiksi.
- Agenttikoodi – Ajetaan kohdekoneella agentin yhteydenoton yhteydessä ja toteuttaa C2-profiilin viestinnän

Mythicissä ei automaattisesti näitä agentteja tai C2-profiileja ole vaan ne pitää itse ladata. Githubissa on olemassa kattava lista näistä ja näiden ominaisuuksista, joita voi sitten itse latailla käyttötarkoituksille sopiviksi. Tähän työhön valittiin Windows-ympäristölle sopivat agentit.

3.2.2. Hyökkäyskuorma

Mythicissä hyökkäyskuormaa kutsutaan englanninkielisellä termillä “payload” eli tarkemmin sanottuna edellä olevien agenttien ja C2-profiilien kokonaisuutta. Hyökkäyskuorma rakennetaan joko käyttöliittymässä tai komentoriviltä, ja se paketoidaan suoritettavaksi tiedostoksi. Esimerkiksi näitä paketoituja tiedostoja voi rakentaa muun muassa: .exe-, .dll- ja .js-muotoon.

4. HYÖKKÄYSSIMULAATIOT

4.1. Aloitus

Opinnäytetyön olennaisena osana oli C2-serverin rakentaminen ja sitä kautta suoritettava tietoturvatestaus. Haluttiin simuloida erilaisia hyökkäyksiä, mitä C2 pystyy suorittamaan, mutta virtuaalimaailmassa. Simuloinnissa suoritetaan tavallisimpia hyökkäyksiä, joita on todella helppo tehdä Mythicillä.

Hyökkäyssimulaatiossa käytettiin VMWare Workstationissa kolmea virtuaalikonetta:

- Kali Linux, jolle Mythic on asennettu ja konfiguroitu toimivaksi
- Windows 10 Pro
- Parrot

Hyökkäyksiä testattiin ensimmäisenä Windows-ympäristöön, sillä se on yleisin käytössä oleva käyttöjärjestelmä organisaatioissa ja siten todennäköisin kohde todellisissa hyökkäystilanteissa (Core Security 2024).

Hyökkäyksiä testattiin myös Linux-ympäristöön ja erityisesti Parrot Securityn ylläpitämään Parrot-käyttöjärjestelmään. Tähän sopisi moni muukin Linux-pohjainen käyttöjärjestelmä, mutta valittiin Parrot tähän, koska se on tietoturvatestaukseen suunnattu jakelu.

Korostettakoon, että suoritetut hyökkäyssimulaatiot on toteutettu suljetuissa virtuaaliympäristöissä. Kuvattuja tekniikoita tai hyökkäyksiä **ei tule suorittaa** missään fyysisessä ympäristössä, sillä nämä rikkovat Suomen lakia ja toiseksi nämä pystyvät tekemään hyvin paljon vahinkoa.

4.2. Kohdejärjestelmä

Opinnäytetyötä varten tarvittiin kohdekoneisto, joka on tässä työssä sekä Windows- että Linux-virtuaalikone. TAMKilla on useampi Windows-virtuaalilevytiedosto ja näistä ladattiin yksi käyttöön. Tämän asennusprosessi oli selkeä ja yksinkertainen. Tarkempia asennustoimenpiteitä ei tarvittu, vaan tiedosto ladattiin suoraan VMware Workstationille ja siihen asetettiin tarvittavat verkkoasetukset. Parrotin virtuaalikoneelle sopiva .iso -tiedosto ladattiin heidän omasta verkko-osoitteestansa.

4.3. Apollo ja HTTP

Hyökkäyssimulaatio aloitettiin aivan yksinkertaisella tunkeutumisella Windows-siin. Simulaatiota varten ladattiin Apollo-agentti ja http-pohjainen C2-profiili Mythicin Githubista sopivilla komennoilla.

Apollo on erityisesti Windows-järjestelmiin tehty agentti (Apollo 2025; Mythic 2025). Sen tarkoitus on suoritella jälkihyökkäystoimintoja, eli sellaisia operaatioita, jotka tapahtuvat sen jälkeen, kun järjestelmään on päästy sisään. Tämä agentti sopii erinomaisesti tähän ensimmäiseen hyökkäyssimulaatioon.

Apollon keskeisiä ominaisuuksia on muun muassa:

- .NET- ja PowerShell-koodien binäärin suoritus
- Tiedostojen hallinta ja prosessin manipulointi
- Koodin injektointi muihin prosesseihin
- Peer-to-peer-yhteys (SMB, TCP, Webshell)

C2-profiiliksi valikoitui http, joka käyttää tavallisia GET- ja POST-pyyntöjä viestien välittämiseen agentin ja käytössä olevan Mythic-serverin välillä. Tämä profiili on helppo naamioida normaaliksi verkkoliikenteeksi, toimii palomuurien ja välityspalvelimien läpi, sekä tukee SSL-salausta. Lyhyesti tämä http toimii tavallaan viestintäkanavana, jonka kautta käytössä oleva Apollo-agentti saa tehtävänsä ja palauttaa tulokset takaisin Mythic-serverille operaattorin analysoitavaksi.

4.3.1. Ensimmäinen simulointi

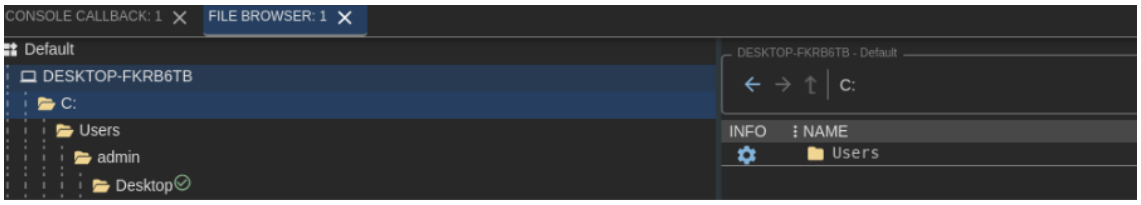
Simulointi varten aloitettiin hyökkäyskuorman rakennus. Tämä onnistui Mythicin Web-käyttöliittymästä helposti valitsemalla haluttu käyttöjärjestelmä ja muut konfiguraatiot. Huomattakoon, että tässä hyökkäyskuorman rakennuksessa komentopalvelimen osoitteeksi on asetettava hyökkäyskoneen oma IP-osoite. Muuten hyökkäyskuorma ei rakennu oikein ja Apollon käyttämä yhteys jää syntymättä. Rakennusvaiheessa määritettiin Apollon agentille lyhyt pingausväli (suomeksi esimerkiksi yhteydenotto- tai takaisinsoittointervalli, englanninkielinen termi “callback interval”). Tämä siittä syystä, että silloin kun agentti on käynnissä kohdejärjestelmässä, se on nopeampi reagoimaan annettuihin komentoihin tai tehtäviin. Rakennusvaiheessa määriteltiin agentille halutut komennot, joita se pystyisi suorittamaan kohdejärjestelmässä. Valittiin tässä kohtaa simuloinnin vuoksi jokainen saatavilla oleva komento.

Hyökkäyskuorman rakennuksen jälkeen syntynyt .exe-tiedosto ladattiin Windows-kohdejärjestelmään SSH-yhteydellä Kalilta.
Oletettavaa olikin, että Windowsin oma Defender-ohjelmisto blokkaa tiedoston välittömästi latauksen jälkeen. Koska kyseessä on pelkkä simulointi, blokattu tiedosto vapautettiin Windowsin työpöydälle. Tiedosto suoritettiin ja se aukaisi tavallisen tyhjän komentoikkunan.

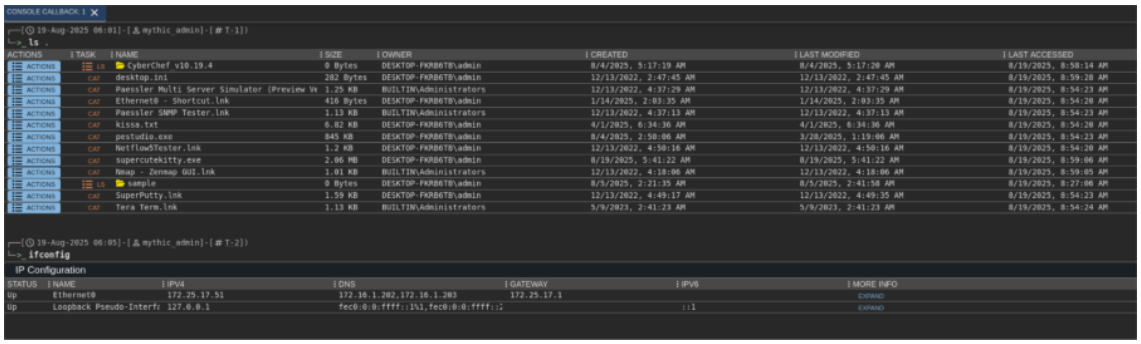
Mythicin käyttöliittymä näytti saman tien tiedoston suorituksen jälkeen aktiivisen yhteydenotto agentilta. Ikkunasta näkyy myös juokseva kirjautumisaika (englanniksi “check-in time”), jolloin Apollo ottaa määritellysti yhteyttä Mythiciin. Tässäkohtaa simulointia kohdejärjestelmään oli saatu onnistunut tunkeutuminen ja luotu hyökkäyskuorma toimi miten oli suunniteltu, kuten voidaan kuvasta 6. todeta.

KUVA 6. Mythicin käyttöliittymän näkymä aktiiviset yhteydenotot agentilta! Tästä voidaan nähdä kohdejärjestelmän IP-osoite, isäntä, käyttäjänimi, verkkotunnus ja prosessin numero (PID). (Kuva: Jaakko Oja)

Onnistuneen tunkeutumisen jälkeen Apollolle annettin erilaisia komentoja. Tavallisilla komennoilla, esimerkiksi “ifconfig”:lla pystyttiin tarkastelemaan kohdejärjestelmän verkkoyhteyksiä (Liite 1). Mythic antaa myös näppärästi tiedostoselausikkunan, jonka avulla kohdejärjestelmän tiedostoja voidaan tarkastella eri kansioissa, kuten kuvista 7 ja 8 huomataan. Huomionarvoista tässä on myös se, että Mythicin operaattori voi tässä kohtaa tunkeutumista luoda myös omia kansioita kohdejärjestelmään.



KUVA 7. Mythicin tiedostoselausikkunassa voidaan selata kohdejärjestelmän kansioita (Kuva: Jaakko Oja)



KUVA 8. Agentille annettu “ls”-komento, jolloin nähdään mitä kohdejärjestelmän latauskansiossa on. Myöhemmin haluttu nähdä verkko-osoitteet sopivalla komennolla (Kuva: Jaakko Oja)

Keskeinen tavoite hyökkääjillä on pysyvyyden varmistaminen tunkeutumisen jälkeen eli halutaan varmistaa pääsy kohdejärjestelmään myös myöhemmin. Simulaatiossa hyökkääjä olisi voinut ladata kohdejärjestelmään esimerkiksi takaportteja sisältäneitä tiedostoja. Näin ei kuitenkaan toimittu, vaan jätettiin syntynyt http-yhteys aktiiviseksi mahdollista myöhempää hyödyntämistä varten.

Apollo-agentti pystyy myös suorittamaan erilaisia injektointiin liittyviä toimintoja, esimerkiksi:

- Shellcode-injektio: Raa'an shellkoodin injektointi haluttuun prosessiin
- PowerShell-injektio: PowerShell-komentojen suoritus
- Token-manipulaatio: Identiteettivarkaus ja toisen käyttäjän esittäminen

Vaihtoehtoisesti jos hyökkääjän tavoitteena ei ole palaaminen kohdejärjestelmään, olisi hän voinut ladata kohdejärjestelmästä haluttuja tiedostoja Mythicin serverille. Tätä simulaatiota varten ei kohdejärjestelmään lähdetty tekemään erikseen mitään erilaisia "arkaluonteisia" tiedostoja tai eri käyttövaltuuksilla olevia käyttäjiä, vaan tarkoituksena oli demonstroida tavallista C2-yhteyttä, hyökkäyskuorman rakennusta ja kohdejärjestelmän saastutusta.

4.4. C2-välityspalvelin

Ammattihakkerit ja rikollisjärjestöt yrittävät parhaansa mukaan piilottaa kaikki C2-serverit ja näissä kulkevan liikenteen. Avuksi tähän on olemassa C2-välityspalvelimia. Nämä kuuntelevat saapuvia yhteyksiä ja välittävät ne eteenpäin C2-palvelimelle. Pääasiallisia tehtäviä näillä palvelimilla on suojata koko serverin sijaintia, olemassaoloa ja identiteettiä, ettei kukaan näitä löydä tai pääse kuuntelemaan niissä tapahtuvaa liikennettä.
Tällaisia työkaluja hyödyntämällä vaikeutetaan esimerkiksi kyberuhkien torjunnan ja jäljityksen mahdollisuuksia yhdistää yhteyksiä takaisin C2-palvelimelle (MITRE ATT&CK 2020; C2 Matrix 2021).

4.5. Apollo ja HTTPS

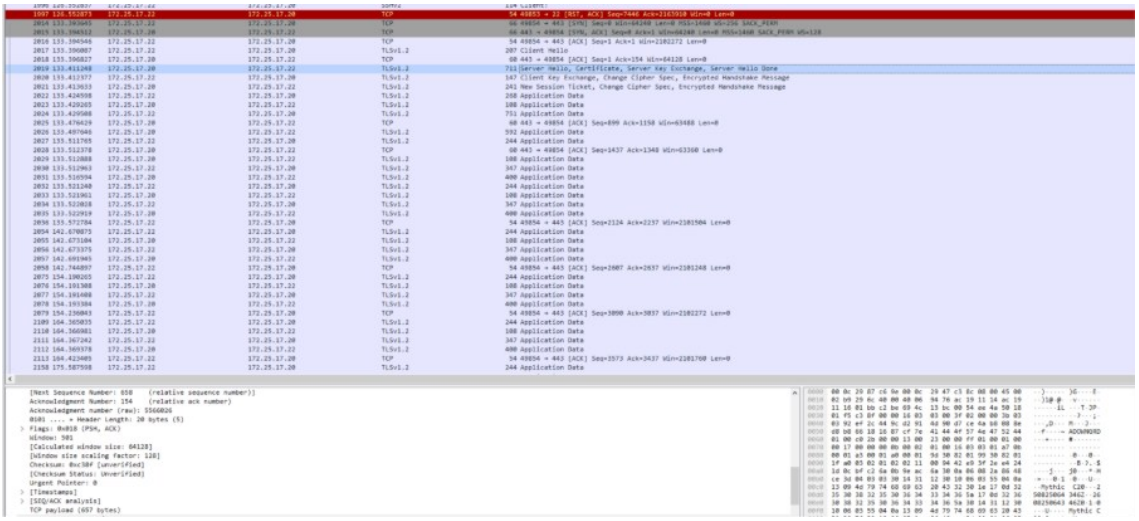
4.5.1. Toinen simulointi

Ensimmäisessä simuloinnissa hyökkäyskuorma siirrettiin kohdejärjestelmään SSH-yhteydellä Kalilta, mutta Windowsin oma Defender-virustorjuntaohjelmisto

esti tämän. Myös pelkän http:n käyttäminen Apollon kommunikoinnissa aiheuttaa erilaisia haasteita, esimerkiksi kohdejärjestelmää puolustava osapuoli pystyy helposti urkkimaan C2-liikenteen, sillä http kulkee täysin salaamattomana.

Haluttu saavuttaa yhden askeleen parempi salaus hyökkäyskuormalle ja C2-liikenteelle käyttämällä Apollo-agentin ja https C2-profiilin yhdistelmää. Https tarjoaa tiedonsiirrolle salauksen, mikä vaikeuttaa sen tunnistamista ja on vähemmän altis tunnistautumiselle.

Toisessa simuloinnissa hyökkäyskuorman rakennus meni pitkälti samalla kaavalla kuin ensimmäisessä simuloinnissa, erona oli, että Mythicin käyttöliittymässä tehtiin muutos http C2 -profiiliin vaihtamalla portti 80 porttiin 443 sekä aktivoimalla SSL-vaihtoehto. Tämä havainnollistettu paremmin visuaalisesti kuvassa 9 Wireshark-näkymänä.



KUVA 9. Https-liikenteen kaappaus Wiresharkissa (Kuva: Jaakko Oja)

4.6. Obfuskointi ja Linux

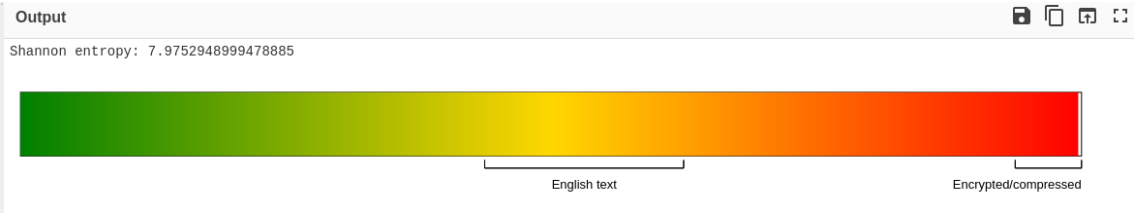
Obfuskointi eli lyhyesti tiedoston tai ohjelmakoodin tahallista monimutkaistamista, niin että sitä on vaikea ihmisen ymmärtää. Tähän simulointiin käytettiin obfuskointia uteliaisuuden takia, mutta myös sen takia että käytetty työkalu, Bincrypter, on ominaisuuksiltaan sopiva tähän tarkoitukseen.

Bincrypter on Bash -kielellä suunniteltu työkalu, joka on suunniteltu salaamaan, pakkaamaan ja obfuskoimaan Linuxin ELF-binääritiedostoja ja shell-skriptejä. Sen tarkoituksena on suojata haluttua tiedostoa analyyseilta ja mahdollisilta tietoturvauhilta. Mikä tekee tästä työkalusta myös erittäin käytetyn, on sen kyky suorittaa obfuskoitu tiedosto (alkuperäinen binääri) suoraan kohdejärjestelmän muistissa, jolloin se ei jätä jälkiä tiedostojärjestelmään eikä tiedostoa tallenneta levylle (Sandfly Security 2025).

Työssä rakennettiin Linuxille sopiva hyökkäyskuorma Poseidon-agentilla ja http-yhteydellä. Tähän olisi sopinut myös salattu yhteys, mutta haluttiin nähdä tietoliikennettä vähän paremmin. Rakennuksen Bincrypterillä tehtiin salaus salasanalla “funnycats”, jolloin lopputuloksena alkuperäisestä ELF-tiedostosta tuli shell-skripti, joka sisältää salatun binäärin ja purkaa sen ajon aikana. Tiedosto ei tässä vaiheessa enää näyttänyt ELF-binääriltä, ja sen nimi muutettiin vähemmän tunnistettavaan muotoon “5up3r5ecret.bin”, josta terminaalinäkymä kuvassa 10. Onnistunut salaus tarkistettiin CyberChef-nimisellä ohjelmistolla, joka näytti kuvassa 11 Shannonin entropia-arvoksi lukuarvoa 7,97.

KUVA 10. Bincrypter on hyvä ja tehokas työkalu ELF-tiedostojen salaamiseen. Tässä kuvakaappauksessa alkuperäinen “supersecret.bin”-niminen hyökkäyskuorma on salattu salasanalla. (Kuva: Jaakko Oja)

Salauksen jälkeen vietiin valmis tiedosto Parrottiin “wget”-komennolla Kalilta. Huomionarvoista tässä kohtaa oli, että Parrot ei ilmoittanut mistään haittaohjelmasta latauksen aikana, toisin kuin Windowsin oma Defender-virustorjuntaohjelmisto. Latauksen jälkeen tiedosto suoritettiin Parrotissa komennolla “./5up3r5ecret.bin”.



KUVA 11. CyberChef-ohjelmisto näyttää obfuskoidulle tiedostolle lukua 7.97, mikä taasen tarkoittaa vahvaa obfuskaatiota, eikä siinä ole helposti tunnistettavaa rakennetta.

Suorituksen jälkeen tiedosto teki sen, mitä pitikin, avaamalla aktiivisen C2-kanavan takaisin Mythicin serverille. Kohdejärjestelmänä olevaan Parrottiin ei luotu tässä kohtaa mitään erillisiä tiedostoja tai käyttäjiä vaan haluttiin tutkia Bincrypterin ominaisuutta, jolla alkuperäinen binääri suoritellaan suoraan RAM-muistissa. Tätä tekniikkaa kutsutaan englanninkielisellä termillä “fileless execution”.

Tämä hyökkäystekniikka onnistui erinomaisesti! Bincrypter käytti memfd_create()-tekniikkaa eli tiedosto ladattiin, purettiin ja suoritettiin ilman pysyvää tallennusta levylle. (Liite 2) Tämän tekniikan etuna on esimerkiksi hyvin vaikea jäljitettävyys haittaohjelma-analyysissä. Tämä on myös tehokas tapa kiertää virustorjuntaohjelmistot, että myös suorittaa simuloituja hyökkäyksiä ilman pysyviä jälkiä.

Tällä hyökkäyssimulaatiolla saavutettiin tilanne, jota kuvataan englanninkielisellä termillä ”evasion”, eli hyökkäys vältti havaitsemisen. Suoritettu tekniikka on myös keino väistellä perinteisiä suojausmekanismeja, esimerkiksi EDR-ratkaisut (engl. “Endpoint Detection and Response“), jotka seuraavat tiedostojärjestelmän muutoksia. Bincrypterin käyttö osoitti, kuinka tehokkaasti haitallinen toiminta, tässä C2-kanavan muodostuminen, voidaan naamioida normaalilta vaikuttavaksi prosessiksi.

5. HAVAITSEMINEN JA SUOJAUTUMINEN

5.1. Mythicin luoma tietoliikenne

Keskeisenä osana opinnäytetyötä on Mythicin luoma tietoliikenne ja tämän havainnointi erilaisin työkaluin. Työssä haluttiin tarkastella C2-kanavan salausta, minkälaisessa muodossa liikenne on, sekä mahdolliset tunnistettavat piirteet.

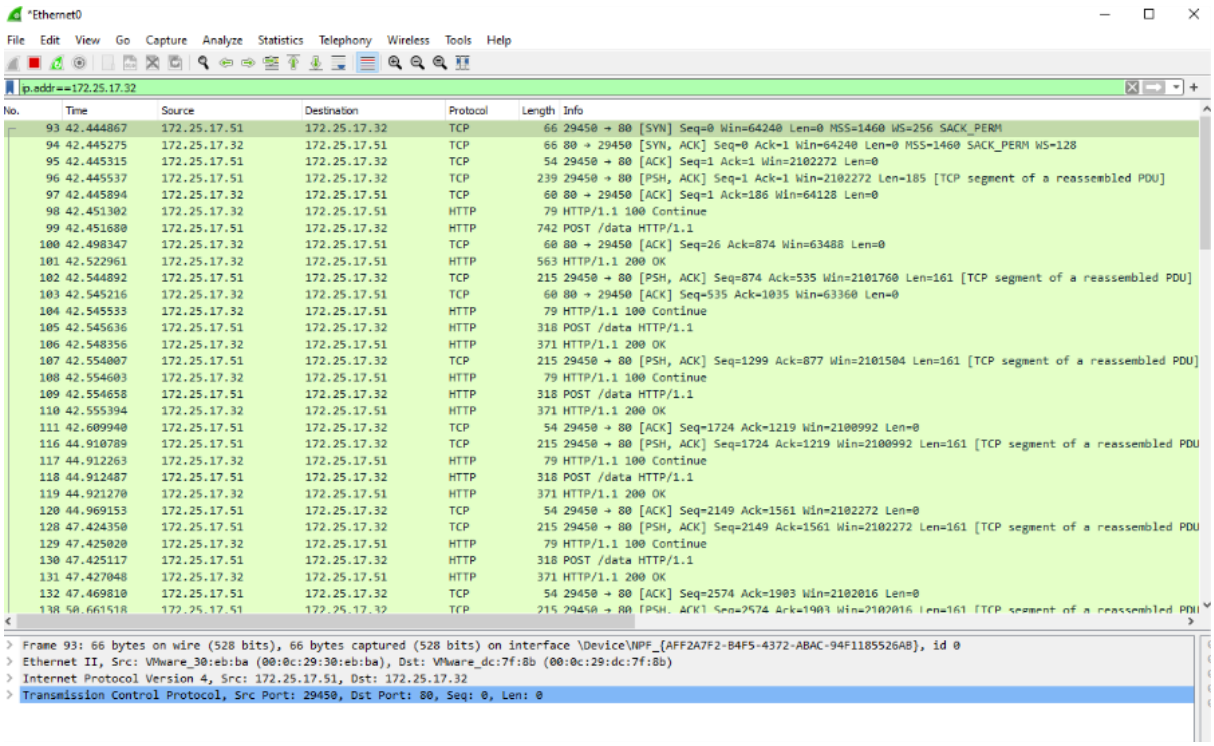
Havainnointiin käytettiin Wiresharkkia, jonka avulla pyrittiin tunnistamaan poikkeavuuksia normaalista verkkoliikenteestä. Wireshark on verkkoliikenteen analysointiin tarkoitettu työkalu, joka mahdollistaa pakettien tarkastelun sekä reaaliaikaisesti että myös tallennetusta kaappauksesta.

5.2. Havaitseminen

C2-yhteydet aiheuttavat paljon tietoliikennettä, kuten kuvasta 12 voidaan todeta. Tämän liikenteen havainnointi aikaisessa vaiheessa on hyvinkin keskeistä vahinkojen estämiseksi. Varhaisella havaitsemisella estetään esimerkiksi arkaluontoisten tietojen leviäminen ulkopuolisille tekijöille.

Ensimmäisiä indikaattoreita voi olla esimerkiksi:

- Poikkeavuudet verkkoliikenteessä: Jos verkkoliikenteessä näkyy odottamattomia piikkejä, erityisesti hiljaisina käyttöaikoina (esimerkiksi öisin), niin tämä voi olla merkki aktiivisesta ja olemassa olevasta C2-yhteydestä.
- Epätavalliset yhteydet, jotka suuntautuvat erityisesti ulospäin: Tämä on yksi yleisin ja varmin merkki aktiivisesti C2-yhteydestä ja tällöin järjestelmä on vaarantunut. Tämänkaltainen ulospäin suuntautuva verkkoliikenne tyypillisesti on naamioituna normaalin liikenteen sekaan.
- Verkkotunnus- ja DNS-kyselykuviot: Toistuvat DNS-kyselyt samaan verkkotunnukseen tai nopeasti vaihtuvaan voivat viitata siihen, että hyökkäävä osapuoli käyttää verkkotunnusten generointialgoritmeja (DGA) kommunikoidakseen C2-serverin kanssa (Hunt.io 2025).



KUVA 12. Wireshark näyttää tässä kuvakaappauksessa aktiivisen http-liikenteen C2-serverin ja Windows-kohdejärjestelmän välillä ensimmäisen simuloinnin aikana (Kuva: Jaakko Oja 2025)

5.3. Suojautuminen

C2-liikenteeseen suojautuminen on haaste nykypäivän modernissa kyberturvallisuudessa, mutta ehdottomasti tarpeellinen. Hyökkääjät käyttävät yhä kehittyneempiä menetelmiä esimerkiksi naamioimalla komentokanavat tavalliseen verkkoliikenteeseen ja hyödyntävät usein salattuja tietoliikenneyhteyksiä.

Suojautuminen edellyttää kattavia ja monikerroksisia tietoturvastrategioita, jotka yhdistävät erilaisia suojausasetuksia, havaitsemismekanismeja sekä ennakoivia toimenpiteitä. Lisäksi organisaatioiden tulisi panostaa henkilöstön koulutukseen ja jatkuvaan valvontaan, sillä ihmisten tekemät virheet ja sosiaalinen manipulointi ovat edelleen merkittäviä hyökkäysreittejä C2-infrastruktuurien muodostumisessa (ENISA 2021; MITRE ATT&CK 2018).

5.3.1. SIEM

Lyhenne tulee englanninkielisestä sanasta “Security Information and Event Management”, mutta suomeksi käytetään yleisesti termiä tietoturvatapahtumien hallintaratkaisut. SIEM tulisi konfiguroida havaitsemaan C2-toimintaan liittyvät poikkeamat, esimerkiksi:

- Yhteydet tunnetuille C2-IP-osoitteille
- Hyvin isot tiedonsiirrot tuntemattomiin kohteisiin
- Uudet ja tuntemattomat ulkoiset yhteydet sisäisistä järjestelmistä

5.3.2. DNS-tunnelointi ja vaihtoehtoiset C2-kanavat

DNS-tunnelointi on hyökkääjillä yleinen tekniikka, jota käytetään, jos halutaan kiertää perinteiset suojausmekanismit ja halutaan siirtää tietoa ulos verkosta (Palo Alto Networks). Tämänkaltaisen tunneloinnin estämiseksi organisaatioiden tulisi:

- Valvoa poikkeavia DNS-kyselyitä, esimerkiksi tiheät kyselyt tuntemattomiin verkkotunnuksiin
- Estää tunnetut ja haitalliset DNS-resolverit
- Käyttää DNS-suojausratkaisuja, esimerkiksi verkkotunnuksen luokittelijoita. Hyviä vaihtoehtoja on muun muassa Cisco Umbrella, Quad9 ja Cloudflare Gateway, jotka suodattavat DNS-kyselyjä.

Muitakin vaihtoehtoisia C2-kanavia on olemassa, jotka ovat hyökkääjillä aktiivisessa käytössä:

- ICMP-protokollat, esimerkiksi ping-pohjaiset
- Sähköpostipohjaiset C2-kanavat
- Lailliset pilvipalvelut: Jotkut C2-infrastruktuurit käyttävät esimerkiksi Googlen Driveä, Slackia, Telegramia tai Discordia komentojen suorittamiseen.

Näiltä vaihtoehtoisilta kanavilta suojautumisessa ensiarvoisen tärkeää on käyttää syväpakettitarkistuksia, englanniksi “deep packet inspection” (DPI).

5.3.3. Nopean ja salatun liikenteen estäminen

C2-liikenne halutaan hyökkääjän näkökulmasta salatuksi ja mahdollisimman nopeaksi, jolloin sen havaitseminen on vaikeampaa. Blokkaamalla liikenne, jota ei voida tarkastella ja käyttämällä SSL/TLS-tarkastuksia palomuureissa voidaan paljastaa C2-liikenne.

5.3.4. Sallitut kohteet verkkoliikenteessä

Merkittävä tietoturvariski syntyy, kun ulospäin suuntautuvalle tietoliikenteelle annetaan laajat tai rajoittamattomat oikeudet muodostaa yhteyksiä Internettiin. Tämänkaltainen toiminta voi mahdollistaa haitallisten tai tuntemattomien kohteiden tavoittamiseen, josta jatkumona syntyy esimerkiksi tietovuotoja, haittaohjelmien latauksia tai C2-kanavien syntymisiä.

Riskien hallitsemiseksi organisaatioiden tulisi ottaa käyttöön pääsylistoja, joissa on ennalta hyväksyttyjä verkkokohteita. Tämä auttaa rajaamaan tietoliikenteen vain luotettuihin, ennalta tiedettyihin kohteisiin ja vähentää merkittävästi altistumista haitallisille verkkoyhteyksille.

5.3.5. IPS

Lyhenne muodostuu englanninkielisestä termistä “Intrusion Prevention Systems “, mutta suomeksi lyhenne kääntyy tunkeutumisen estojärjestelmä. Tämänkaltainen estojärjestelmä tulisi olla käytössä palomuureissa ja erilaisissa verkkoturvaratkaisuissa. Myös uhkia puolustavan osapuolen olisi hyvä säännöllisesti päivittää uhkatiedustelusyötteitä.

5.3.6. EDR

Osa kehittyneistä C2-infrastruktuureista, esimerkiksi Cobalt Strike tai Mythic, voi olla vaikeasti havaittavissa pelkästään verkon tasolla. EDR-lyhenne tulee englanninkielisestä termistä “Endpoint Detection and Response“, ja sama termi suomeksi kääntyy päätelaitteiden havaitsemis- ja reagointiratkaisuihin. Tämä on keskeinen osa syväsuojautumisstrategiaa, erityisesti kehittyneimpiä C2-infrastruktuureja vastaan (MITRE ATT&CK 2018; CrowdSrike 2025; ExtraHop 2023).

Näitä vastaan organisaatioiden tulisi ottaa käyttöön erilaisia käyttäytymispohjaisia sääntöjä, jotka voivat valvoa esimerkiksi:

- Epäilyttävää Powershell tai WMI-toimintaa
- Poikkeavaa muistin injektointia tai suoritusmenetelmiä

6. POHDINTA

Opinnäytetyössä käsiteltiin yleisellä tasolla C2-infrastruktuureja, mitä ne sisältävät, mitä ne pystyvät tekemään ja miten niitä vastaan kannattaa yritysten suojautua. Tällä jo pienellä käsittelyllä voidaan todeta, ettei tietoturva ole enää valinnainen asia – se on kriittinen osa yritysten digitaalista arkea. Yritykset ja organisaatiot, koosta riippumatta, ovat jatkuvasti alttiita erilaisille kyberuhille. C2-infrastruktuurit osoittavat, kuinka kehittyneitä ja myös järjestelmällisiä hyökkäykset voivat olla.

Tietoturva ei ole pelkästään tekninen ongelma, vaan myös inhimillinen. Sosiaalinen manipulointi, englanniksi “social engineering “, on kyberuhka, jonka tavoitteena on vaikuttaa käyttäjän toimintaan psykologisesti. Sosiaalisen manipuloinnin hyökkäykset ilmenevät monin eri tavoin, esimerkiksi huijaussähköpostien muodossa mutta muita tapoja on esimerkiksi tietojenkalasteluhyökkäykset, joka on yksi yleisin ihmisen manipulointiin perustuva huijaus (F-Secure 2024). C2-infrastruktuurit hyödyntävät tätä tehokkaasti, sillä huolimattomuus ja tiedon puute ovat usein tehokkaampia kuin tekniset haavoittuvuudet.

Yritysten näkökulmasta tietoturvan jatkuva kehittäminen ja varautuminen on osa tehokasta digipuolustusta, mutta myös ennakointia sen pahan päivän varalle. Kun tiedetään, miten hyökkääjät toimivat, voidaan kehittää tehokkaampia havainto- ja torjuntajärjestelmiä. Avainsanana tähänkin on tietoturvaosaamisen kehittäminen, koulutus ja erilaiset uhkamallinnukset, sillä hyvä tietoturva on osa digitaalista luottamusta.

C2-infrastruktuurien tutkiminen ja simulointi ei ole pelkästään tehokkaiden hyökkäystekniikoiden ymmärtämistä, vaan se tarjoaa eri näkökulmia, miten tietoturvaa voidaan kehittää realistisella tasolla. Kun puolustava osapuoli oppii ymmärtämään näitä hyökkääjien käyttämiä työkaluja, voidaan ennakoida tulevia uhkia tehokkaammin.

Opinnäytetyön aikana hyvinkin nopealla katsauksella selvisi, että tekninen osaaminen ei yksin riitä. Tietoturva vaatii kokonaisvaltaista lähestymistapaa, jossa yhdistyvät ihmisten toiminta sekä teknologia.

Loppukaneettina todettakoon, että tietoturva pelkästään ei ole systemaattisesti yksittäinen ratkaisu, vaan jatkuva prosessi. C2-infrastruktuurien kaltaiset järjestelmät osoittavat sen, kuinka monimutkainen ja järjestelmällinen kyberuhkien maailma on – ja kuinka tärkeää on ymmärtää sitä, jotta voidaan rakentaa turvallinen digiympäristö.

LÄHTEET

MITRE ATT&CK.2018. Command and Control, Tactic TA0011. Verkkosivu. Luotu 17.10.2018; Viimeksi muokattu 25.4.2025. Viitattu 04.11.2025. https://attack.mitre.org/tactics/TA0011/
Hunt. 2025. C2 channels. Verkkosivu. Viitattu 21.08.2025. https://hunt.io/glossary/c2-channels
StationX. 2024. What is a C2 Framework? Verkkosivu. Viitattu 21.08.2025. https://www.stationx.net/what-is-a-c2-framework/
Kali Linux. 2025. What is Kali Linux? Verkkosivu. Viitattu 19.11.2025. https://www.kali.org/docs/introduction/what-is-kali-linux/
Core Security. 2024. 2024 Penetration Testing Report. Verkkosivu. Viitattu 19.11.2025. https://www.coresecurity.com/resources/guides/2024-pen-testing-survey-report/
NIST. 2020. SP 800-207: Zero Trust Architecture. Gaithersburg, MD: National Insitute of Standards and Technology. Viitattu 04.11.2025. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf
Cobalt Strike. N.d. Cobalt Strike – Adversary Simulation and Red Team Operations. Verkkosivu. Viitattu 04.11.2025. https://cobaltstrike.com/
BishopFox / Sliver. 2025. Sliver – Adversary Emulation Framework. Verkkosivu. Viitattu 04.11.2025. https://github.com/BishopFox/sliver/
Havoc. 2024. Havoc Framework. Verkkosivu. Viitattu 04.11.2025. https://havocframework.com/
Empire (PowerShell Empire). 2020. Empire – PowerShell and Python post-exploitation framework. Verkkosivu. Viitattu 04.11.2025. https://github.com/EmpireProject/Empire/
C2 Matrix. 2024. The C2 Matrix. Verkkosivu. Viitattu 04.11.2025. https://howto.thec2matrix.com/
Mythic. 2025. Mythic Documentation. Verkkosivu. Viitattu 21.08.2025. https://docs.mythic-c2.net/
https://github.com/its-a-feature/Mythic
C2 Redirectors Part.1 | Red Team Leaders

C2 Matrix. 2021. Redirectors/Relays. Verkkosivu. Viitattu 28.08.2025.

https://howto.thec2matrix.com/attack-infrastructure/redirectors

MITRE ATT&CK.2020. T1090: Proxy. Verkkosivu. Luotu 14.03.2020. Viitattu

28.08.2025. https://attack.mitre.org/techniques/T1090/

Palo Alto Networks. What is DNS Tunneling? Verkkosivu. Viitattu 19.11.2025.

https://www.paloaltonetworks.com.au/cyberpedia/what-is-dns-tunneling?/

How to create your own mythic agent in C - Red Team SNCF

Apollo. 2025. Apollo Agent Documentation. Verkkosivu. Viitattu 04.11.2025.

https://docs.mythic-c2.net/agents/apollo/

Sandfly Security. 2025. Detecting Bincrypter Linux Malware Obfuscation. Verkkosivu. Viitattu 28.08.2025. https://sandflysecurity.com/blog/detecting-bincrypter-linux-malware-obfuscation

Kali Linux Tutorials. 2025. Bincrypter: Enhancing Linux Binary Security through Runtime. Verkkosivu. Viitattu 28.08.2025. https://kalilinuxtutorials.com/bincrypter/

MythicC2Profiles/http | DeepWiki

What are C2 Frameworks? Types and Examples

What Are C2 Frameworks? (+Free 2025 Setup Guide)

C2 Architecture - Pull the Strings, Run the Show

Command and Control (C&C) Attacks

Mitä on käyttäjän manipulointi eli social engineering? | F-Secure

PrimusRedir - Fowarding the world to your C2 | . .\Primusinterp

Hiding C2 Infrastructure behind i2p by "abusing" open i2p relays/in-proxies.

https://deepwiki.com/MythicAgents/Apollo/6.2-code-injection-techniques

https://hunt.io/glossary/detect-c2

Crowdstrike. 2025. What is Endpoint Detection and Response (EDR)? Verkkosivu. Viitattu 01.10.2025. https://www.crowdstrike.com/en-us/cybersecurity-101/endpoint-security/endpoint-detection-and-response-edr/

ExtraHop. 2023. Detecting Cobalt Strike with ExtraHop Reveal(x) Verkkosivu. Viitattu 01.10.2025. https://www.extrahop.com/blog/detecting-cobalt-strike-with-extrahop-reveal-x

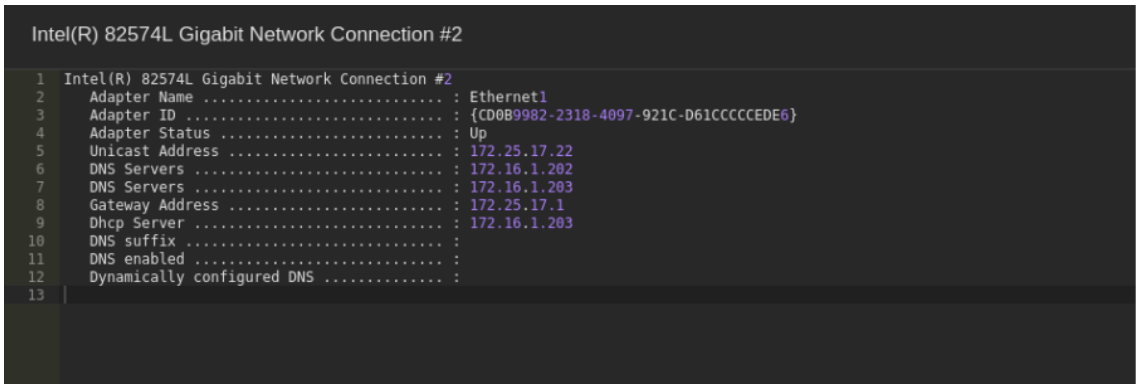
https://osintteam.blog/preventing-command-and-control-c2-traffic-a-multi-layered-approach-39627e7f2477

ENISA. 2021. ENISA Threat Landscape 2021. European Union Agency for Cybersecurity. Verkkoraportti. Viitattu 4.11.2025. https://www.enisa.europa.eu/publications/enisa-threat-landscape-2021

F-Secure. 2024. Mitä on käyttäjän manipulointi? Verkkosivu. 18.03.2024. Viitattu 01.10.2025. https://www.f-secure.com/fi/articles/what-is-social-engineering

LIITTEET

Liite 1. IP-verkkoyhteyksiä tarkasteltu Mythicin agentilla.



Liite 2. Todiste onnistuneesta hyökkäystekniikasta