Utu Hopiavuori

Tietoturvan automatisointi verkko-sovellusalustapalveluissa

Ansible tietoturvan tukena

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintätekniikka

Opinnäytetyö
28.3.2025

Tiivistelmä

Tekijä(t):	Utu Hopiavuori
Otsikko:	Tietoturvan automatisointi verkko-sovellusalustapalve-luissa
Sivumäärä:	xx sivua + x liitettä
Aika:	28.3.2025
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Tieto- ja viestintätekniikka
Ohjaaja(t):	Osaamisaluejohtaja Janne Salonen

Tiivistelmän tekstiosuus kirjoitetaan mahduttaen se sivulla käytössä olevaan tilaan. Tekstiosuudessa käytetään **Leipäteksti ilman välistystä** -tyyliä.

Avainsanat: Avainsana

Abstract

Author(s):	First Name Last Name
Title:	Title of the Thesis
Number of Pages:	xx pages + x appendices
Date:	28.3.2025
Degree:	Bachelor of Engineering
Degree Programme:	Information and Communication Technology
Instructor(s):	Janne Salonen, Director of school

Tiivistelmän tekstiosuus kirjoitetaan mahduttaen se sivulla käytössä olevaan tilaan. Tekstiosuudessa käytetään **Leipäteksti ilman välistystä** -tyyliä.

Keywords: Keyword

Sisällys

1 Johdanto 1

1.1 Sovellusalustan vakiointi 1
1.2 Ansible konfiguraatioautomaatio-ohjelmistona 2
1.3 Vaihtoehtoiset tietojärjestelmäautomaatioratkaisut 2

2 Työn toteuttaminen 3

2.1 Syntaksitarkistus 4
2.2 Tietojenkäsittely-ympäristön luominen ja konfiguroiminen 4
2.3 Idempotenssi 5
2.4 Generoitujen salasanojen vahvuus 6

3 Opinnäytetyöprojektissa huomioitavaa 7

3.1 Tietosuoja koodiautomaatiossa 7
3.2 Avoimen lähdekoodin lisensointi 7
3.3 Eurooppalainen kyberturvadirektiivi (NIS2) 8

4 Lopuksi 8

Lähteet 9

Liitteet 10

1 Johdanto

Tein palkattoman työharjoittelun yrityksessä, joka nimettäköön tätä opinnäytettä varten nimellä Yritys Oy. Tarkoituksena oli toteuttaa vielä palkallinen harjoittelu, jossa opinnäytettä olisi laajennettu tietoturvatapahtumien valvonnan suuntaan, mutta kyseinen projekti peruuntui taloudellisista syistä. Teen opinnäytteen Metropolia-ammattikorkeakoululle ja opinnäytteen ohjaaja on tietoinen varsinaisesta yrityksestä, jolle työ tehtiin ja jonka edustajan piti olla alun perin mukana opinnäytetyön teknisenä ohjaajana. Koska projektia ei ikinä toteutettu loppuun saakka, päätin tietosuojasyistä käyttää kyseisestä yrityksestä pseudonyymiä. Yrityksen todellisella nimellä tai automatisoidulla palvelualustalla ajettavien sovellusten kauppanimillä ei ole merkitystä työn sisällön kannalta. Alkuperäisen suunnitelman mukaan olisin ollut ko. yritykseen työsuhteessa opinnäyteprojektin aikana.

Opinnäytetyössäni selvitän tarkemmin tuotetun koodin omistajuuteen ja lisensointiin liittyvistä asioista sekä yrityksen immateriaalioikeuteen liittyvistä seikoista, koska mainittu työharjoittelu tehtiin salassapitosopimuksen alaisena.

1.1 Sovellusalustan vakiointi

Sain tehtäväkseni kehittää sovellusalustan vakioimista varten skriptejä tai rakenteista konfiguraatiota, jota voidaan sopivalla tietojenkäsittely-ympäristön automaatioon käytetyllä ohjelmistolla sitten ajaa palvelimille, joiden sovelluspalveluiden avulla on tarkoitus ajaa verkkosovelluksen komponentteja: edusta- ja taustapalveluita.

Skriptien piti toteuttaa seuraavat vaatimukset:

1. Käyttöjärjestelmä on tietoturvapäivitysten osalta ajantasainen
2. Vaadittavat sovellukset on asennettu ja kofiguroitu oikein
   2.1. Apache ja PHP edustapalveluita tarjoavilla palvelimilla
   2.2. Apache, PHP ja MySQL taustapalveluita tarjoavilla palvelimilla

    2.3. Apachen modsecurity-moduulit ovat asennettuina ja konfiguroituina
3. Yrityksen palvelutunnukset lokien keräämistä ja palveluiden mahdollista paikallista vianselvitystä varten on luotu ja niillä on tarkoituksenmukaiset oikeudet ja pääsyt tiedostojärjestelmässä.
    3.1. Käyttäjätunnusten salasanat pitää voida uusia käyttäen skriptejä ja ne pitää voida lähettää sähköpostitse yrityksen omaan sähköpostiosoitteeseen.
4. SSH-palvelun portit pitää satunnaistaa ja ne pitää voida satunnaistaa uudelleen, jos on tarpeen.

Skripteihin oli tarkoitus lisätä tietoturvavalvontaa (SIEM) varten konfiguroidut palvelut, mutta tämän työn kontekstissa kyseiset palvelut kuvataan tarkemmin jatkokehitysmahdollisuuksia käsittelevässä opinnäytetyön luvussa.

Lähes kaikki asiat, joita työssäni hyödynsin löytyvät esimerkkeineen kirjasta *Ansible: Up and running (3rd edition),* (1).

1.2 Ansible konfiguraatioautomaatio-ohjelmistona

Ansible-ohjelmiston dokumentaation mukaan Ansible on avoimen lähdekoodin työkalupaketti, joka käyttää yksinkertaista ja ihmiselle selkeästi luettavissa olevia skriptejä, joita kutsutaan pelikirjoiksi (2).

Ylläpitäjänä määrittelen tietojenkäsittely-ympäristöltä vaadittavan tilan Ansiblen pelikirjoihin ja rooleihin. Ansible on kehitetty varmistamaan, että ympäristö toteuttaa nämä vaatimukset.

Pelikirjat on tavallisesti kirjoitettu käyttäen YAML- tai JSON-rakenteista merkintäkieltä. Tässä työssäni käytin YAML-rakennetta skripteissäni.

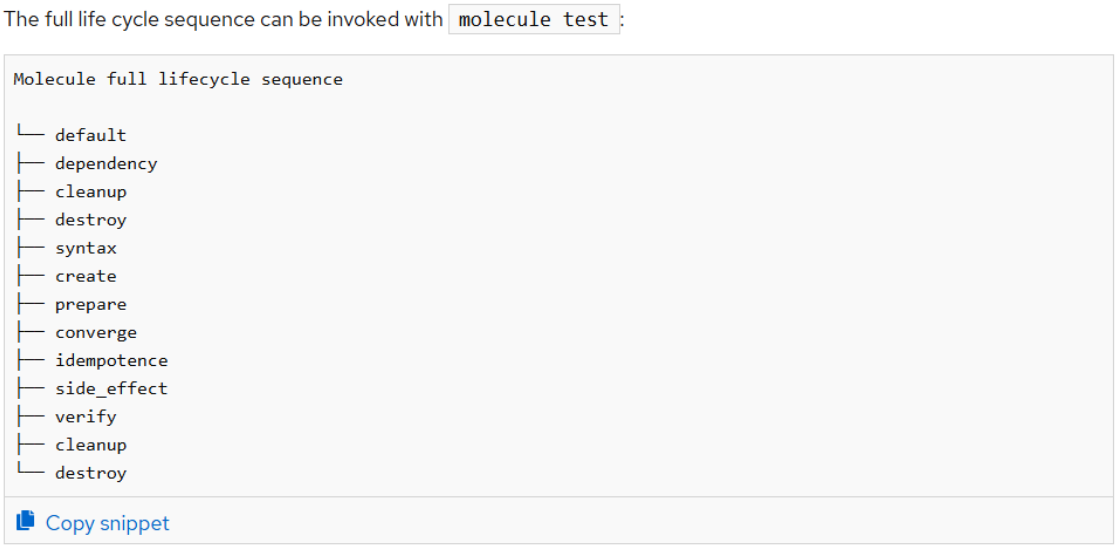
1.3 Vaihtoehtoiset tietojärjestelmäautomaatioratkaisut

Ansiblen merkittävimmät kilpailijat tai vaihtoehtoiset tavat lähestyä tietojärjestelmien automaatiota ja palveluiden orkestrointia ovat Puppet, Chef ja Salt Stack (3).

2 Työn toteuttaminen

Ansible-pelikirjat ja roolit ovat tarkoitetut ajettaviksi standardoidulla tavalla riippumatta siitä, onko konfiguroitava palvelualusta tai palvelin konfiguroitu tai vielä konfiguroimatta. Palvelimen tila on tarkasti määritelty pelikirjojen ja roolien kautta, joten palveluita voidaan testata käyttäen Molecule-testikirjastoa (4).

Työssäni en toteuttanut koko testikirjastossa määriteltyä palvelun elinkaarta (Kuva 1). Kuvaan seuraavissa aliluvuissa tarkemmin testikirjastossa määriteltyjä vaiheita, joiden vaatimukset pyrin työssäni täyttämään.


The full life cycle sequence can be invoked with `molecule test`:

```
Molecule full lifecycle sequence

└── default
├── dependency
├── cleanup
├── destroy
├── syntax
├── create
├── prepare
├── converge
├── idempotence
├── side_effect
├── verify
├── cleanup
└── destroy
```

Copy snippet

Kuva 1: Molecule-testikirjaston koko elinkaari

Koska en työssäni käyttänyt ohjelmistoalustapalvelimien luomisessa automaatiota, loin virtuaalikoneet käsin kloonaamalla täysin konfiguroimatonta perusasennusta, jossa toki oli käyttämäni asennustunnus ja salasana asetettuna. Myöhemmin pelikirjoissa salasanakirjautuminen asennustunnuksella estetään ja Ansible-ohjauskoneen SSH-avaimet lisätään tunnukselle, jolloin ohjelmistoautomaatio toimii käytännössä eri tunnuksilla kuin mahdollinen palvelimien muu konfigurointi ja lokimerkinnät automaation ja ihmiskäyttäjien tekemistä konfiguraatiomuutoksista ovat selvät.

2.1 Syntaksitarkistus

Molecule-prosessin (Kuva 1: Molecule-testikirjaston koko elinkaariKuva 1) mukainen työskentelytapani alkaa pelikirjojen syntaksitarkistuksella, josta on vastuussa ohjelmistokirjasto nimeltään Ansible-lint (5). Kyseinen kirjasto asentuu kiltisti käyttämäni Visual Studio Code -editorin käyttöliittymään.

Ongelmia syntaksitarkistuksessa tuli jonkin verran, koska käytin tiettyjen salassa pidettäväksi tarkoitettujen muuttujien tallentamiseen kryptografisesti vahvaa salausta, jonka sisäistä syntaksia sitten Ansible-lint ei voinut tarkistaa. Tällaisia muuttujia olivat esimerkiksi pääkäyttäjänä ajettavaksi tarkoitettujen komentojen salasanat. Kryptografisesti vahva osuus Ansible-ohjelmistoa on nimeltään Ansible Vault (6).

2.2 Tietojenkäsittely-ympäristön luominen ja konfiguroiminen

Työssäni loin kolme roolia: ”Common”, ”Frontend” ja ”Backend”. Kaikille palvelimille yhteiset konfiguraatiot määrittelin roolin ”Common” alla oleviin pelikirjoihin. Edusta- ja taustapalvelimille erilliset konfiguraatiot loin vastaavien roolien alla oleviin pelikirjoihin.

Koska roolien eriytyminen perustuu esimerkiksi tietokantaohjelmista MySQL:n tarpeeseen tai asioita julkisesti esittävän PHP:n tarpeeseen, niitä sovelluksia ei ”Common”-roolin kautta määritellä. Kyseisen roolin kautta kuitenkin tulee päivitykset asennettuihin paketteihin ja kaikille palvelimille yhteisten palvelutunnusten tai ihmisten käyttöön tarkoitettujen käyttäjätunnusten konfigurointi tehdään myös kaikille yhteisen roolin kautta esimerkiksi tiedostojärjestelmäoikeuksien tai pääkäyttäjäryhmään kuulumisen osalta. Jokaiselle serverille tehdään myös tietoturvavaatimuksista tuleva SSH-palvelinohjelmiston portin satunnaistaminen.

2.3 Idempotenssi

RFC 9110 määrittelee idempotenssin HTTP-käsitteissä seuraavasti: ” A request method is considered "idempotent" if the intended effect on the server of multiple identical requests with that method is the same as the effect for a single such request.” ”Kysely (request) määritellään idempotentiksi silloin, kun sen tavoiteltu vaikutus serverille on sama riippumatta samanlaisten kyselyiden lukumäärästä (7).”

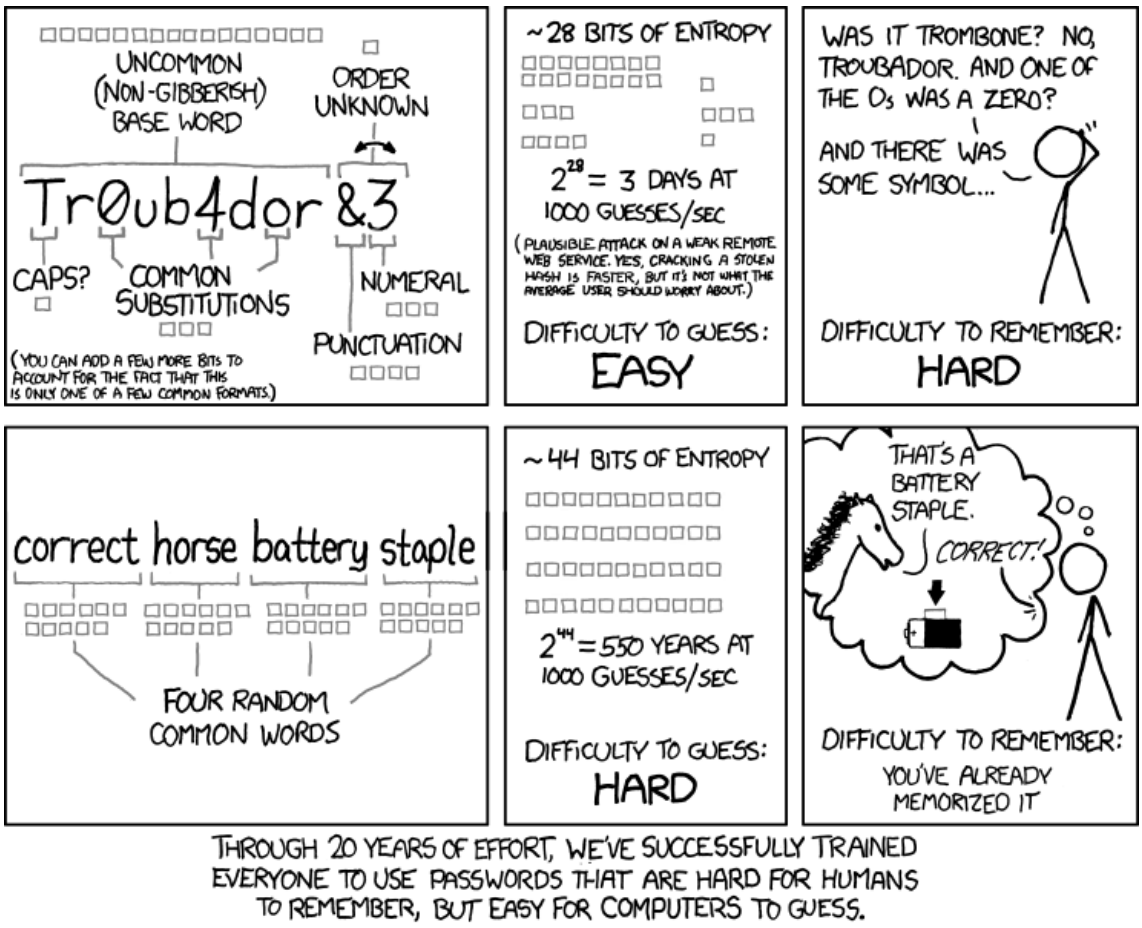
Ansiblessa idempotenttisuus on tärkeä tavoite, kun kirjoitetaan pelikirjoja. Pelikirjoissa määritellään haluttu lopputulos ja pelikirjat suoritetaan aina riippumatta siitä, tarvitseeko jokaista sen tehtävää suorittaa. Jos tehtävä tekee muutoksia kohdepalvelimelle, se raportoi lokiin tehtävän tilaksi ”Changed” (muuttunut) ja jos muutoksia ei tarvitse tehdä, se raportoi lokiin ”OK”. Ansible siis tarkistaa kohdepalvelimien vaatimuksenmukaisuuden aina pelikirjoja suoritettaessa ja se tekee Ansiblesta erinomaisen työkalun tietoturvavaatimusenmukaisuuden varmistamiseen.

Koska yrityksen tietoturvavaatimuksissa oli SSH-palvelimen portin satunnaistaminen, satunnaistamisen toteuttavan pelikirjan tehtävän idempotenttisuus osoittautui haastavaksi varmistaa. Jos satunnaistaminen toteutetaan uudelleen, palvelimelle määritelty SSH-portti muuttuu ja tehtävä ei määritelmän mukaisesti ole idempotenttinen. Toteutin koodiin kysymyksen SSH-portin satunnaistamisen tarpeesta ja jos kysymyksen ohitti, porttia ei muutettu. Kysymystä ei kysytty, jos SSH-palvelin oli palvelimella standardiportissa 22, eli vaatimus "SSH-palvelimen portti on satunnaistettu” toteutuu varmasti pelikirjaa suoritettaessa riippumatta siitä, mitä kysymykseen vastaa.

Pelikirjoissa piti myös luoda yrityksen lokitarkastelijan käyttäjätunnus sekä ihmisylläpitäjien käyttöön tarkoitettu käyttäjätunnus. Näiden salasanat piti pystyä tarvittaessa uusimaan ja lähettämään sähköpostilla yrityksen pääkäyttäjille.

2.4 Generoitujen salasanojen vahvuus

XKCD-sarjakuva (8) kertoo hyvin siitä, kuinka vaikeaa tietokoneen on arvata pitkiä salasanoja, vaikka niiden rakenne olisi helposti selvitettävissä. Moni Linux-jakelu tarjoaa sovelluspakettienhallinnan kautta asennettavaksi xkcdpass-sovelluksen (9), jolle voi antaa halutessaan myös muun kuin oletussanalistan. Kotimaisten kielten tutkimuskeskus (KOTUS) julkaisee vuosittain suomen kielen sanalistan koneluettavassa muodossa ja pystyin työssäni hyödyntämään sen yli sadantuhannen sanan kokoelmasta standardiin ASCII-merkistöön sisältyvät vähän yli 64000 sanaa osana skriptiä, joka koosti kolmesta suomenkielisestä sanasta muodostuvan salalauseen.



Kuva 2: xkcd #936 Password Strength

Salasanan vaikeudeksi arvioin optimitilanteessakin (hyökkääjällä on sekä sanalista että tieto salasanojen rakenteesta) vähintään 60 bittiä. Tällaisen salasanan

murtamiseen miljardi hashia sekunnissa vertaillen menisi laskennallisesti 28 vuotta, joten salasanan vahvuus kirjoittamassani skriptikokoelmassa on vähintäänkin riittävä.

3 Opinnäytetyöprojektissa huomioitavaa

Koska toteutin opinnäytetyön varsinaisen työn osuuden salassapitosopimuksen alaisena, joudun työssäni huomioimaan paitsi yrityksen tietojenkäsittely-ympäristön luottamuksellisuuden, myös koodini lisensoinnin ja eurooppalaisen kyberturvadirektiivin vaatimukset pienen yrityksen toimintaan, joka kuitenkin toimii osana julkishallinnon toimitusketjua ja siten kyberturvadirektiivi asettaa vaatimuksia mm. yrityksen raportoinnille ja tietoturvan perustasolle toimitetuissa palveluissa.

3.1 Tietosuoja koodiautomaatiossa

ToDo: koodirepositoryn anonymisointi ja GIT-repositoryn jakaminen alimoduuleihin.

Kävin yrityksen toimitusjohtajan kanssa keskustelun siitä, kuinka työssäni tuottamaani koodia voi opinnäytetyössä käyttää ja hän kertoi, että jos haluan tuottamaani koodia sisällyttää opinnäytetyöhöni, minun pitää julkaista koodi sillä tavalla, että yrityksen aineetonta omaisuutta olevat asiat eivät koodin kautta paljastu ja lisensoida koodini sillä tavalla, että he voivat hyödyntää mahdollista jatkokehitystäni yrityksen tuotteissa riippumatta siitä, olenko työsuhteessa yritykseen.

3.2 Avoimen lähdekoodin lisensointi

Ansible on lisensoitu GNU General Public License v3.0-lisenssillä, joka edellyttää koodia käytettäessä kaiken siihen liittyvän lisensoimista samalla lisenssillä.

Kyseessä on avoimen lähdekoodin lisenssi ja siten työkalujen lisensoinnilla voi olla vaikutuksia kaupalliseen tarkoitukseen kehitettyjen sovellusten kannalta.

Jos Ansiblen koodia ei muokkaa tai tee koodia, joka muokkaa Ansiblen toimintaa, ei esimerkiksi pelikirjoja tarvitse julkaista samalla lisenssillä. Pelikirjat rinnastuvat konfiguraatiotiedostoihin ja ne määrittävät sitä, miten standardinmukainen Ansible toteuttaa tietojenkäsittely-ympäristön automaation.

3.3 Eurooppalainen kyberturvadirektiivi (NIS2)

Yritys, johon työni tein, on kooltaan sen verran pieni, että kriteerit NIS2-direktiivin tiukassa vaikutuspiirissä eivät täyty. Yrityksen tuotteita on kuitenkin osana julkishallinnon toimitusketjuja ja siten NIS2-direktiivin vaatimukset esimerkiksi tietoturvapoikkeamien raportoinnin osalta tulevat tärkeiksi tarkastella.

4 Lopuksi

Lähteet

1. **Meijer, , Hochstein, and Moser, .** *Ansible: Up and Running, 3rd Edition.* s.l. : O'Reilly Media, Inc, 2022.

2. **Ansible project contributors.** Introduction to Ansible. [Online] 2025. [Cited: 22 02 2025.] https://docs.ansible.com/ansible/latest/getting_started/introduction.html.

3. **Venezia, .** Review: Puppet vs. Chef vs. Ansible vs. Salt. [Online] 21 11 2013. [Cited: 17 02 2025.] https://www.infoworld.com/article/2186089/data-center-review-puppet-vs-chef-vs-ansible-vs-salt.html.

4. **Behl, .** Introducing Ansible Molecule with Ansible Automation Platform. [Online] 13 09 2023. [Cited: 17 02 2025.] https://developers.redhat.com/articles/2023/09/13/introducing-ansible-molecule-ansible-automation-platform.

5. **Ansible project contributors, Red Hat project, Will Thames.** About Ansible Lint. *Ansible Lint Documentation.* [Online] [Viitattu: 27. 02 2025.] https://ansible.readthedocs.io/projects/lint/.

6. **Ansible project contributors.** Ansible Community Documentation. *Protecting sensitive data with Ansible vault.* [Online] 21. 03 2025. [Viitattu: 21. 03 2025.] https://docs.ansible.com/ansible/latest/vault_guide/index.html.

Liitteet