TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Kerli Saarniit 212346IADB

Turvatestide raportite automatiseerimine

Bakalaureusetöö

Juhendaja: Einar Kivisalu
MSc

Tallinn 2025

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Kerli Saarniit

06.01.2025

Annotatsioon

Käesoleva lõputöö eesmärgid on analüüsida turvatestide raportite automatiseerimise lahendust ja luua toimiv ning praktiline lahendus, mis automatiseerib turvatestide raportite koostamise ning haldamise protsessi.

Lahenduse eesmärk on vähendada manuaalset tööd, kiirendada tulemuste kättesaadavust ja minimeerida vigu andmete koondamisel ning edastamisel. Lahendusega luuakse rollipõhised raportid, mis on kõigile asjaosalistele ligipääsetavad ühest kohast. Automatiseerimine tagab kiiret, täpset ja ajakohast ülevaadet turvariskidest. Automatiseeritud lahendusega optimeeritakse organisatsiooni ressursikasutust ja suurendatakse protsesside usaldusväärsust.

Lõputöös analüüsitakse probleemi, tutvustatakse olemasolevat lahendust ning kirjeldatakse loodava lahenduse eesmärke. Tutvustakse lahendusega seotud tööriistasid ja tehnoloogiaid. Tuuakse välja lahenduse nõuded, tehnoloogilised valikud ja tehakse ülevaade raportite kavandamise protsessist. Kirjeldatakse lahenduse teostust ja hinnatakse tulemusi, eesmärkide ning nõuete täitmist. Tehakse ülevaade planeeritavatest edasiarendustest.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 28 leheküljel, 7 peatükki, 9 joonist, 2 tabelit.

Abstract

Automation of Security Testing Reports

The aim of this thesis is to analyze a solution for automating security test reports and to develop a functional and practical solution that automates the process of generating and managing such reports.

The purpose of the solution is to reduce manual effort, speed up the availability of results, and minimize errors in compiling and sharing data. The solution will provide role-based reports that are accessible to all relevant stakeholders from a single location. Automation ensures a quick, accurate, and up-to-date overview of security risks. The automated solution optimizes resource usage in the organization and increases the reliability of processes.

The thesis analyzes the problem, introduces the existing solution, and describes the objectives of the proposed solution. It introduces the tools and technologies related to the solution. Requirements for the solution, technological choices, and the report design process are outlined. The implementation of the solution is described, and the results, goals, and fulfillment of requirements are evaluated. An overview of planned future developments is also provided.

The thesis is in Estonian and contains 28 pages of text, 7 chapters, 9 figures, 2 tables.

Lühendite ja mõistete sõnastik

API	*Application Programming Interface,* rakendusliides
ASVS	*Application Security Verification Standard,* rakenduste turvalisuse kontrollimise standard
CI/CD	*Continuous integration and Continuous delivery,* pidev integratsioon ja pidev edastamine
cURL	*Client Uniform Resource Locator,* käsurealiidesega utiliit andmevahetuseks
HTML	*Hypertext Markup Language*, hüperteksti märkimise keel
HTTP	*Hypertext Transfer Protocol*, hüperteksti edastuse protokoll
ID	*identifier*, konkreetset objekti kontekstis üheselt tähistav atribuut
JSON	*JavaScript Object Notation,* JavaScripti alamhulgal põhinev andmevahetusvorming
JQL	*Jira Query Language*, Jira päringukeel
OWASP	*Open Web Application Security Project,* avatud veebirakenduste turvalisuse projekt
PDF	*Portable Document Format,* digitaalne dokumendivorming
REST	*Representational State Transfer,* rakendusliidese arhitektuuri stiil
SQL	*Structured Query Language*, struktureeritud päringute keel
UI	*User Interface,* kasutajaliides
URI	*Uniform Resource Identifier,* ühtne ressursiidentifikaator
URL	*Uniform Resource Locator,* ühtne ressursilokaator
URN	*Uniform Resource Name,* ühtne ressursinimi
YAML	*Ain't Markup Language*, inimloetav andmevahetusvorming
XHTML	*Extensible HyperText Markup Language*, laiendatav hüperteksti märgistuskeel
XML	*eXtensible Markup Language*, laiendatav märgistuskeel

Sisukord

Jooniste loetelu

Tabelite loetelu

1 Sissejuhatus

Tänapäeva digiajastul on turvaohud muutunud ettevõtjate ja üksikisikute jaoks oluliseks mureks. Kuna tehnoloogia areneb ja küberohud muutuvad keerukamaks kasvab ka turvatestimise tähtsus. Turvalisuse testimine on tarkvara ja süsteemiarenduse valdkonnas ülioluline protsess. See hõlmab rakenduse, süsteemi või võrgu põhjalikku hindamist, et tuvastada haavatavused, nõrkused ja võimalikud turvaohud [1].

Töövoo automatiseerimine on lähenemisviis, mis muudab ülesannete, dokumentide ja teabe liikumise tööga seotud tegevuste vahel iseseisvaks vastavalt määratletud ärireeglitele. See on toimingute seeria, mis on vajalik kindla tegevuse täitmiseks. Töövoo automatiseerimiseks tuvastab organisatsioon kõigepealt ülesanded, millest töö koosneb ja loob reeglid ning loogika, mis reguleerivad nende ülesannete täitmist, mille pealt saab programmeerida automatiseerimise tarkvara [2].

Käesolevas lõputöös analüüsitakse turvatestimise raportite automatiseerimist ja luuakse valmislahendus, mis lihtsustab turvatestide raportite loomise töövoogu. Integreeritud lahenduse süsteem ühendab mitu toodet või teenust üheks tervikuks, mis on mõeldud mitmete probleemide lahendamiseks [3]. Loodud integratsioonilahendus peab vastama asutuse nõuetele.

Lõputöö koosneb kuuest peamisest peatükist. Probleemi analüüsi peatükis antakse põhjalik ülevaade töös käsitletava probleemi taustast, kirjeldatakse probleemi detailsemalt ning tutvustatakse kasutatavat metoodikat ja lõputöö eesmärke.

Lahenduse ülevaate peatükis antakse ülevaade kasutatavatest keskkondadest, kirjeldatakse hetkel kasutusel olevat lahendust ja loodava lahenduse eesmärki.

Lahenduse analüüsi peatükis selgitatakse, kuidas vajalikud nõuded välja selgitati ja sõnastati. Selles peatükis analüüsitakse integratsioonilahenduse nõudeid, tehnoloogiate valikut ja lähtetingimusi ning turvatestide raportite kavandamist.

Teostuse peatükis kirjeldatakse arenduse käigus tekkinud probleeme ja kitsendusi. Selles peatükis antakse ülevaade integratsioonilahenduse arendamisest, rakendamisest ja lahenduse testimisest.

Tulemused peatükis hinnatakse, kuidas saavutatud tulemused vastavad seatud eesmärkidele ja nõuetele. Lisaks tuuakse välja planeeritavad edasised tegevused lahenduse täiustamiseks.

2 Probleemi analüüs

Antud peatükis analüüsitakse lõputöö probleemi. Selles peatükis antakse ülevaade lõputöös käsitletud probleemi taustast, kirjeldatakse täpsemalt käsitletavat probleemi ja selle olulisust, tutvustatakse lõputöös kasutatavat metoodikat ning tuuakse välja lõputöö eesmärgid.

2.1 Probleemi taust

Turvaaukude tuvastamine ja turvafunktsioonide tagamine turvatestimise abil on laialdaselt rakendatav meede tarkvara turvalisuse hindamiseks ja parandamiseks. Turvatestimine kinnitab ja valideerib tarkvarasüsteemi nõudeid, mis on seotud turvaomadustega nagu konfidentsiaalsus, terviklus, käideldavus, autentimine, autoriseerimine ja ümberlükkamatus [4].

Turvatestimine koosneb järgnevatest sammudest:

1. Ettevalmistus – Koostatakse tellimus, määratakse skoop, lepitakse kokku turvatestimise periood, tutvutakse testitava rakendusega ja tellitakse vajalikud õigused ning ligipääsud.

2. Testimine – Viiakse läbi turvatestid, et tuvastada testitava rakenduse võimalikud turvanõrkused.

3. Turvaleidude kirjeldamine – Kõik tuvastatud turvaleiud kirjeldatakse Jirasse (tööriista kirjeldus on leitav peatükis 3.1) tellimuspileti alla. Kirjelduseks lisatakse turvavea detailne selgitus koos tehtud sammude ja kasutatud andmetega. Lisaks hinnatakse turvaleiu riske ja antakse soovitused turvariski maandamiseks.

4. Raporti koostamine – Koostatakse kokkuvõtlik turvatestide raport testimise tulemustest.

5. Kättesaadavaks tegemine – Valminud turvatestide raport edastatakse kõigile asjaosalistele.

Turvatestide raportite koostamine on manuaalne protsess, mis võtab märkimisväärselt aega, ehk on ajamahukas. Turvaleidude käsitsi koondamine ja raporti vormistamine suurendab vigade, dubleerimiste ning andmete kaotsimineku riski. Manuaalne protsess tekitab viivitusi otsuste ja vastumeetmete rakendamisel. Häkkekindluse testijate aeg kulub dokumenteerimisele, kuid selle asemel võiksid nad rakendusi turvatestida.

2.2 Probleemi kirjeldus

Turvatestimise käigus leitud turvaleidude põhjal tuleb koostada ülevaatlik raport, mis sisaldab kõiki turvaleidude kirjeldusi ja mille abil saavad erinevad osapooled teha edasisi otsuseid. Hetkel dokumenteeritakse turvaleiud esmalt Jirasse, kust need käsitsi kopeeritakse ja vormistatakse PDF-formaadis (*Portable Document Format*) raportiks. See manuaalne protsess on ajamahukas, ressursikulukas ja inimlikele vigadele kalduv.

Töövoogude automatiseerimisega vähendavad ettevõtted vajadust käsitsi töö ja korduvate ülesannete järele. Selle tulemusena muutuvad protsessid tõhusamaks ja töötajad kulutavad rohkem aega väärtust lisavatele tegevustele. Automatiseeritud tööprotsessid suurendavad tootlikust, tõstavad tõhustust, täpsust ja kiirust ning vähendavad kulusid [5].

Automatiseeritud töövoo kasutuselevõtuga võimaldatakse Jirasse kirjeldatud turvatestimise tellimuse pileti pealt turvatestide raporti kuvamine otse Confluence'i (tööriista kirjeldus on leitav peatükis 3.2) tööruumi (*space*). Selline automatiseerimine kiirendab kogu protsessi, vähendab inimlike vigade tekkimist ja manuaalset tööd ning tagab reaalajas uuendatud ja struktureeritud ülevaate turvatestimise tulemustest.

2.3 Lõputöö metoodika

Antud lõputöö tegemisel kasutatakse arendusuuringut. Autor valis lõputöö metoodika tuginedes Katrin Niglase koostatud koolitusmaterjalidele „Praktikale suunatud uuringudisainid“ [6]. Metoodika valik võimaldab analüüsida lõputöö probleemi, disainida, rakendada ja hinnata lahendust ning teha järeldusi koos üldistustega.

Esmalt analüüsitakse tellijaga lahendust vajavat probleemi ja sõnastatakse plaanitava lahenduse eesmärgid. Planeeritakse, milliseid tehnoloogiad kasutakse ja lepitakse kokku lähtetingimused.

Seejärel kavandatakse turvatestide raportite vaated koos testandmetega ja valideeritakse tehtud vaated üle ning selgitatakse välja vajadused ja nõuded tellijaga ning rollipõhiste isikutega.

Arendusuuringu käigus arendatakse lahendus turvatestide raportite automatiseerimiseks Jira turvatestimise tellimuspileti sisestamise pealt. Raport genereeritakse Confluence keskkonda turvatestide raportite tööruumi õige asutuse (turvatestide tellija) rakenduse lehe alla.

Valminud lahendust testitakse uute testandmetega, et kinnitada selle vastavust seatud nõuetele. Koos tellijaga hinnatakse lahenduse vastavus ootustele ning vajadusel tehakse täiendusi, et saavutada soovitud tulemus.

Saadud lahendus integreeritakse tööprotsessi ja kogutakse kasutajate tagasisidet, et teha tulevikus täiendusi ning parendusi, mis vastavad nende vajadustele.

2.4 Lõputöö eesmärgid

Lõputöö eesmärgid on käsitletavat lahendust analüüsida ja luua toimiv ning praktiline lahendus, mis automatiseerib turvatestide raportite koostamise ja haldamise protsessi.

Antud lõputöös selgitatakse välja turvatestide raportite automatiseerimise nõuded ja lähtetingimused ning analüüsitakse tehnoloogilisi võimalusi. Lõputöö raames luuakse turvatestide raportite kavandid ja tehniline lahendus, mis seob Jira ning Confluence'i süsteemid automatiseeritud raportite koostamiseks. Lõpuks hinnatakse lahenduse mõju protsessi kiirusele, täpsusele ja kasutusmugavusele.

3 Lahenduse ülevaade

Selles peatükis tehakse ülevaade lõputöö lahenduses kasutatavatest tööriistadest Jira ja Confluence, tutvustatakse hetkel kasutusel olevat lahendust ning selgitatakse lahti lahenduse eesmärk.

3.1 Jira

Jira on agiilne projektijuhtimistööriist, mida meeskonnad kasutavad tarkvara planeerimiseks, jälgimiseks, väljaandmiseks ja toetamiseks [7].

Jira on loodud keerulisteks projektideks kui ka igapäevasteks ülesanneteks, aidates planeerida ja korraldada ülesandeid. See on sobilik lühikestele projektidele või suurtele, Jira aitab jagada suuri ideid saavutatavateks sammudeks, korraldada tööd, luua verstaposte, kaardistada sõltuvusi ja palju muud. Jira seob töö ettevõtte eesmärkidega, et kõik näeksid, kuidas nende panus aitab kaasa üldistele eesmärkidele ja püsiksid kooskõlas olulisega. Tööd saab jälgida just endale sobival viisil: loendite, tahvlite, tööde ootejärjekordade ja muuga. Optimeeritud ülevaadete abil on võimalik näha projekti edenemist, mõista riske ja tuua esile reaalajas andmetest lähtuvaid teadmisi, et aidata meeskonna tulemuslikkust parandada [8].

Antud probleemi puhul on Jira kasutusel turvatestide tellimiseks ja turvaleidude kirjeldamiseks.

3.2 Confluence

Confluence on koostöövahend, mis koondab inimesed, teadmised ja ideed ühisesse tööruumi, mis võimaldab iga projekti või meeskonna jaoks eraldi ja suletud kohta, kus töötada [9].

Dünaamilised Confluence'i lehed annavad meeskonnale koha, kus luua, jäädvustada, organiseerida ja teha koostööd mis tahes projekti või idee kallal. Confluence tööruumid

aitavad meeskonnal tööd struktureerida, korraldada ja jagada, nii et igal meeskonna-liikmel on ülevaade institutsionaalsetest teadmistest ja juurdepääs teabele [10].

Confluence Storage Format on Confluence'is kasutatav XHTML-i (*Extensible HyperText Markup Language*) vorming lehtede, lehemallide, jooniste, blogipostituste ja kommentaaride sisu talletamiseks. See vorming võimaldab Confluence'is hallata makrosid, dünaamilist sius ja keerukamaid stiile [11].

XHTML on HTML-i (*Hypertext Markup Language*) ja XML-i (*eXtensible Markup Language*) kombinatsioon, mis on väga sarnane HTML-iga, kuid rangem. See on nagu veebilehtede loomise reeglistik, millest brauserid hõlpsasti aru saavad. Erinevalt HTML-ist peab olema ettevaatlik ja järgima täpselt reegleid [12].

Käesoleva lõputöö lahenduses säilitatakse Confluence'is turvatestide raporteid, et neid oluliste osapooltega mugavamalt jagada. Autor kasutas Confluence Storage Formatit probleemi lahenduses turvatestide raporti struktureerimisel.

3.3 REST API-d

Atlassian Jira ja Confluence kasutavad mõlemad REST API-sid (*Representational State Transfer Application Programming Interface*). REST API on programmeerimisliides, mis võimaldab süsteemidel omavahel suhelda, järgides REST-i arhitektuuri stiili. See on lihtne ja ühtne liides, mida kasutatakse andmete, sisu, algoritmide, meedia ja muude digitaalsete ressursside kättesaadavaks tegemiseks veebi URL-ide (*Uniform Resource Locator*) kaudu [13].

REST API abil saab luua rakendusi Jira ja Confluence jaoks, arendada integratsioone nende ja teiste rakenduste vahel või automatiseerida skriptide kaudu suhtlust nendega. Antud lahenduses on kasutatud isikupõhist pääsuluba, mis on turvaline autentimismeetod. See seob kasutajakonto pääsuloaga, pakkudes turvalist ja mugavat alternatiivi kasutajanime ning parooli kasutamisele erinevate teenustega autentimiseks. Jira ja Confluence REST API-d võimaldavad juurdepääsu ressurssidele URI-teede (*Uniform Resource Identifier*) kaudu. URI on ühtne ressursiidentifikaator, mis kõigi nime- ja aadressitüüpide üldnimetus Internetis viitavatele objektile (nt veebilehed, failid, andmeallikad) ning seda on kaks alaliiki: URL (aadressivorming interneti ressursside asukoha määramiseks ja nende juurdepääsemiseks) ja URN (*Uniform Resource Name*)

(määratleb ära ressursi unikaalse nime, kuid ei anna infot selle asukoha kohta) [14]. Päringutele vastatakse JSON-vormingus (*JavaScript Object Notation*) andmetega. Mõlemad REST API-d kasutavad oma suhtlusvorminguna standardseid HTTP-meetodeid (*Hypertext Transfer Protocol)* nagu GET (andmete pärimine), POST (uute ressursside loomine), PUT (olemasolevate ressursside muutmine) ja DELETE (ressursside kustutamine) [15], [16].

Lõputöös käsitletava probleemi lahenduses kasutatakse Jira REST API otspunkte *issue* ja *search* ning Confluence'i otspunkte *content* ja *space*. Jirast päritakse GET-päringuga *issue* otspunkti kaudu konkreetse pileti ID (*identifier*) ning *search* otspunkti kaudu leitakse JQL-i abil kindlat tüüpi piletid. Confluence'ist päritakse *content* otspunkti kaudu GET-meetodiga lehe andmed, POST-meetodiga luuakse uusi lehti ning PUT-meetodiga muudetakse olemasolevaid. Tööruumi andmeid päritakse GET-päringuga *space* otspunkti kaudu.

3.4 Kasutusel olev lahendus

Hetkel on kasutusel lahendus, kus esimese sammuna täidab turvatestimist vajava rakenduse tootejuht Jiras turvatestimise tellimuspileti, kuhu sisestatakse kõik vajalikud andmed testimise läbiviimiseks. Enne turvatestimist korraldatakse rakenduse tutvustus, et häkkekindluse testijad mõistaksid testitavat rakendust ja turvatestide ulatust. Pärast tutvustust hindavad testijad, millal nad saavad testimist teostada ning kaua see aega võtab.

Turvatestimisel tuvastatud turvaleiud dokumenteeritakse esmalt Jirasse. Seejärel kopeeritakse andmed käsitsi ja vormistatakse PDF-formaadis raportiks. Turvatestide raportite koostamisel kasutatakse täielikult manuaalset lähenemist, kus kõik leitud turvaleiud kirjeldatakse ühes tervikfailis. Raportid edastatakse krüpteeritult asjaosalistele PDF-formaadis. Selline manuaalne protsess on ajakulukas, ressursimahukas ja kaldub inimlikele vigadele, mis mõjutab oluliselt töö efektiivsust, viivitab otsustusprotsessidega ning tekitab olukorra, kus tundlikud andmed on mitmes kohas laiali.

3.5 Lahenduse eesmärk

Turvatestide raportite automatiseerimise lahenduse eesmärk on automatiseerida turvatestide tulemustest raportite loomise protsess, et vähendada manuaalset tööd,

kiirendada tulemuste kättesaadavust, minimeerida inimlike vigade tekkimise võimalust andmete koondamisel ja edastamisel ning piirata tundlike andmete levitamist kolmandate süsteemide kaudu. Lahendusega luuakse standardiseeritud ja selgelt struktureeritud rollipõhised raportid, mis on kergesti loetavad ning sisaldavad kõiki vajalikke detaile turvaleidude kohta. Jiras registreeritud turvatestide tulemused kuvatakse Confluence'is automaatselt ja ajakohastatult, mis võimaldab sidusrühmadel kiirest ligipääsu turvatestide leidudele, mis tagab kiire reageerimise ning tõhusama riskide juhtimise.

Automatiseeritud lahendus Jira ja Confluence'i vahel tagab turvatestimise tellimuspileti registreerimisel vastava Confluence'i lehe automaatse loomise ning selle reaalajas uuenemise. Confluence'i lehele kuvatud turvatestide raportid kasutavad ettemääratud malle ja sisaldavad kõiki turvatestide olulisi andmeid nagu tellimuseinfo, turvaleidude kirjeldused, leidude prioriteetsus ja staatus, soovituslikud tegevused turvavea lahendamiseks.

Turvatestide raportite automatiseeritud lahendus loob ligipääsu võimaluse värsketele turvaleidudele kõigile asjaosalistele ühest kohast. Automatiseerimine kiirendab raportite loomist, uuendamist ja loob täpsema ning ajakohase ülevaate leitud turvariskidest. Ühtne ja läbipaistev raportisüsteem vähendab kommunikatsiooniprobleeme.

Automatiseeritud lahendusega optimeeritakse organisatsiooni ressursikasutust ja suurendatakse protsesside usaldusväärsust.

4 Lahenduse analüüs

Antud peatükis käsitletakse, kuidas selgusid lahenduse nõuded, mis analüüsiti ja määratleti vastavalt tellija vajadustele, samuti selgitatakse välja asutuse poolt valitud tehnoloogiad ning lähtetingimused. Lisaks kirjeldatakse, kuidas kavandati turvatestimise raporti struktuur ja vaade, mis vastaks kõigi osapoolte nõudmistele.

4.1 Nõuete väljaselgitamine

Nõuete väljaselgitamiseks analüüsis autor koostöös tellijaga turvatestide raportite automatiseerimise vajadust, et saada selge ülevaade tellija ootustest lahendusele. Vestluse käigus keskenduti sellele, milliseid väärtusi ja eeliseid automatiseeritud lahendus peaks pakkuma, samuti milliseid konkreetseid probleeme ja kitsaskohti see peab lahendama ning määratleti peamised eesmärgid ja soovitud tulemused.

Lahendus peab minimeerima manuaalse töö mahtu ja kiirendama raportite loomist, tagades reaalajas uuendatud andmed. Turvatestide raport peab tekkima turvatestimise tellimuspileti sisestamisest Jirasse. Ootus on, et inimlike vigade tekkimise risk väheneb turvaleiudest raportite koostamisel ja raportid on ühtse struktuuri ning kujundusega, mis hõlbustaks nende lugemist ja analüüsimist. Raportid peavad olema kergesti kättesaadavad Confluence'i tööruumis erinevatele sidusrühmade (nt infoturbejuhid, tootejuhid, arendajad).

Saadud informatsiooni põhjal koostati lahenduse jaoks funktsionaalsed nõuded, mis määratlevad süsteemi vajalikud omadused ja tegevused, ning mittefunktsionaalsed nõuded, mis keskenduvad lahenduse kvaliteedile, töökindlusele ja kasutusmugavusele. Need nõuded loovad aluse lahenduse edasiseks kavandamiseks ja arendamiseks.

4.2 Lahenduse nõuded

Nõuded jagunevad üldiselt kahte tüüpi: funktsionaalsed ja mittefunktsionaalsed nõuded. Funktsionaalsed nõuded määratlevad süsteemi konkreetse käitumise või funktsioonid, mida süsteem peaks tegema, et täita kasutajate vajadusi. Nende hulka kuuluvad sellised

omadused nagu andmetöötlus, autentimine ja kasutajatevaheline suhtlus. Seevastu mittefunktsionaalsed nõuded määratlevad, kuidas süsteem oma ülesandeid täidab, keskendudes sellistele omadustele nagu jõudlus, turvalisus, skaleeritavus ja kasutatavus [17].

Selles peatükis toob autor välja loodava lahenduse funktsionaalsed ja mittefunktsionaalsed nõuded.

4.2.1 Funktsionaalsed nõuded

Loodava integratsioonilahenduse funktsionaalsed nõuded on järgmised:

- Lahendus peab automaatselt looma vastava Confluence'i lehe iga Jiras registreeritud turvatestimise tellimuspileti kohta.
- Genereeritud raportid peavad olema õige asutuse rakenduse lehe all.
- Raportid peavad sisaldama tellimuseinfot, turvaleidude kirjeldusi, leidude prioriteetsust ja staatust, samuti soovitatavaid tegevusi riskide maandamiseks.

4.2.2 Mittefunktsionaalsed nõuded

Loodava integratsioonilahenduse mittefunktsionaalsed nõuded on järgmised:

- Lahendus peab võimaldama andmete reaalajas uuendamist, et tagada alati kõige ajakohasem info.
- Lahendus on paindlik edasiarenduste suhtes.
- Turvatestimise tiimil peab olema võimalus kohandada raportite malli vastavalt vajadustele.

4.3 Lähtetingimused

Turvatestide raportite automatiseerimise lahendusele on tellija määranud lähtetingimused, millega tuleb autoril probleemi lahendusel arvestada.

Üks peamine kriteerium on, et lahendus peab olema kirjutatud Pythoni programmeerimiskeeles. Python on väga paindlik ja selle põhistruktuurid on lihtsad, selged ning hästi läbimõeldud. Sellel on lihtne süntaks, mis võimaldavad

programmeerimise põhialuste kiiret mõistmist võrreldes teiste populaarsete programmeerimiskeeltega [18]. Tellija usub, et valitud programmeerimiskeel on turvatestimise tiimile kergesti omandatav ja toetab lahenduse kiiret kasutuselevõttu.

Tellija on otsustanud, et UI (*User Interface*) vaadet lahendusele ei looda, sest see tooks kaasa suurema halduskoormuse, arendusmahu ja nõuaks täiendavaid turvameetmeid. Lahendus peab olema loodud Atlassiani toodete Jira ja Confluence abil, kus andmete liikumine toimub automaatselt Jirast Confluence'i. Selline lähenemine aitab lahendust kiiremini realiseerida.

Lahenduse teostamiseks tuleb kasutada teenusekontot, mis on loodud asutuses süsteemide vaheliseks integreerimiseks ja automatiseeritud ülesannete täitmiseks. Teenusekonto aitab tagada süsteemide vahelist sujuvat ja turvalist suhtlust, kuna see on spetsiaalselt seadistatud, et täita kindlaid ülesandeid, vältides vajadust eraldi määrata juurdepääsuõigusi igale tavakasutajale.

Lahenduse loomisel peab autor kasutama PostgreSQL-i andmebaasi, kuna see on asutusesiseselt juba kasutusel. Teise andmebaasi valimine tooks kaasa halduskoormuse suurenemise ja turvalisuse vähenemise.

Uute automatiseerimiste loomiseks on asutusel loodud standariseeritud projektimall, kus on defineeritud minimaalse projekti struktuur koos näitekonfiguratsioonifailidega, geneeriliste GitLab CI/CD (*Continuous integration and Continuous delivery*) töövoo (*pipeline*) seadistuste ja Dockeri konfiguratsioonifailidega. Malli eesmärgiks on lihtsustada ja kiirendada projektide loomist, tagades samas ühtse ja efektiivse töövoo järgimise kõigis arendustes.

4.4 Tehnoloogiliste valikute analüüs

Selles peatükis analüüsitakse tehnoloogilisi valikuid, mida lahenduse saavutamiseks kasutatakse. Autoril tuli valida Pythoni ja PostgreSQL-iga ühilduv andmebaasi teek, raporti malli failiformaat ja nende parsimiseks sobiv lahendus.

4.4.1 Andmebaasi teegi valik

Esialgses skoobis oli Jira veebihaakide (*webhook*) kasutamine, mille jaoks kasutati Flask raamistikku. Veebihaagid aitavad saata sõnumeid, hoiatusi, teateid ja reaalajas teavet

serveripoolsest rakendusest kliendipoolsesse rakendusse. Selle asemel, et andmeid korduvalt taotleda, saab vastuvõttev rakendus oodata ja saada vajalikud andmed kohe, kui need on saadaval, ilma et oleks vaja korduvaid taotlusi teisele süsteemile saata [19].

Flask osutus valituks, kuna see on lihtne ja minimalistlik Pythoni veebiraamistik ning sobis hästi Jira veebihaakide realiseerimiseks. Flaskiga saab lihtsalt määrata, millist aadressi ja milliseid päringumeetodeid rakendus kuulab. Samuti on Flaski sisseehitatud funktsionaalsus, millega saab hallata erinevaid HTTP vastusekoode. Lisaks kasutab Flaskis kavandi (*blueprint*) kontseptsiooni, millega salvestatakse toiminguid, mida täita kui see on taotluses registreeritud. Flask seostab funktsioone kavandi taotluste saatmisel ja URL-ide loomisel ühest lõpp-punktist teisele [20].

Flaski raamistikuga sobib hästi kokku SQLAlchemy andmebaasiteek, mis võimaldab kasutada ORM-i (*Object-Relational Mapping*) tehnikat. ORM-tehnoloogia eesmärk on lihtsustada andmebaasidega suhtlemist, võimaldades manipuleerida andmeid SQL-i (*Structured Query Language*) päringute asemel otse objektidega. See lähenemine vähendab oluliselt vajadust kirjutada keerukaid SQL-päringuid käsitsi, vähendades sellega süntaksivigade ja valede päringute tekkimise riski [21], [22]. Nagu eelnevalt mainitud kasutatakse asutusesiseselt PostgresSQL-i andmebaasi. SQLAlchemy pakub põhjalikku dokumentatsiooni PostgreSQL-i spetsiifiliste võimaluste kasutamiseks [23]. Antud andmebaasi teek toetab ka projekti edasiarenduse plaane ja see ei sea tulevikus piiranguid andmetega manipuleerimisele.

4.4.2 Failiformaatide valik

Turvatestide raportite konfiguratsioonifailide jaoks tuli valida formaat, mis oleks kergesti arusaadav turvatestimise meeskonna liikmetel ja võimaldaks lihtsalt luua uusi rollipõhiseid konfiguratsioonifaile vastavalt vajadusele.

Autor valis kolme kõige populaarsema andmete serialiseerimiskeele vahel: XML, JSON ja YAML (*YAML Ain't Markup Language*). XML on märgistuskeel, samas kui JSON ja YAML on andmevormingud. XML kasutab elementide määratlemiseks silte ja salvestab andmed puustruktuuri, samas kui JSON-i andmed salvestatakse võtme/väärtuse paaridega. YAML seevastu võimaldab andmeid esitada nii loendi või järjestuse vormingus kui ka võtme/väärtuse paaride kujul. XML toetab keerukaid andmetüüpe (nt diagramme, pilte ja muid mitteprimitiivseid andmetüüpe). JSON toetab ainult sõnesid,

numbreid, massiive, tõeväärtusi ja objekte. YAML seevastu toetab keerukaid andmetüüpe, nagu kuupäeva- ja ajatemplid, järjestused, pesastatud ja rekursiivsed väärtused ning primitiivsed andmetüübid. XML-is kirjutatud andmeid on raske lugeda ja tõlgendada, kuid JSON-vormingus andmeid on üsna lihtne tõlgendada ning andmeid on YAML-vormingus palju lihtsam lugeda kui JSON-vormingus. YAML sobib kõige paremini konfigureerimiseks [24], [25].

Failiformaatide võrdluse tabelis (Tabel 1) on välja toodud olulised omadused turvatestide raportite konfiguratsioonifailide koostamisest ja kasutamisest lähtudes.

Tabel 1. Failiformaatide võrdlus.

Andmevorming / **Omadus**	**XML**	**JSON**	**YAML**
Inimloetavus	Madal	Kõrge	Väga kõrge
Süntaksi keerukus (k.a. erimärgid ja sümbolid)	Väga keeruline	Keskmine	Lihtne
Andmete töötlemise kiirus	Aeglane	Kiire	Kiire
Andmete ülesehituse paindlikkus	Väga kõrge	Keskpärane	Kõrge
Kasutusmugavus	Madal, kuna keeruline lugeda ja kirjutada	Kõrge, laialdaselt kasutatav	Väga kõrge, sobib hästi konfiguratsioonideks
Vigade tuvastamine	Aeganõudev, kuid kindel	Lihtne, kuid võib olla keeruline suures andmemahus	Suhteliselt lihtne, kuid võib anda segaseid vigu

Parimaks valikuks osutus YAML, kuna raportite mallid pidid tagama inimloetavuse, lihtsa muudetavuse ja kasutusmugavuse. YAML võimaldab andmeid esitada selges ja arusaadavas struktuuris, olles samal ajal hästi loetav nii arendajatele kui ka mittekoodiga tegelevatele spetsialistidele. Lisaks on YAML hästi kohandatav ja sobib suurepäraselt konfiguratsioonifailide haldamiseks turvatestide raportite automatiseerimise lahenduse raames.

4.4.3 Failide parsimise teegi valik

Rollipõhiste turvatestide raportite uuendamisel tuli võrrelda olemasolevate ning muudetud lehtede Storage Format väärtusi. Selleks oli vaja lahendust, mis suudaks XHTML-i käsitleda nii, et atribuutide järjestus pole oluline, kuna XHTML-is pole atribuutide järjestust fikseeritud. Parimaks valikuks osutus Beautiful Soup, kuna see võimaldab kergesti võrrelda kahe lehe sisu kontrollides elementide olemasolu, atribuutide väärtusi ja hierarhiat.

Beautiful Soup toetab mitmeid Pythoni parsereid nagu *lxml*, *html.parser* ja *html5lib*. Autor valis nendest *lxml*-i, kuna see on kiireim ja sobib XHTML-i formaadi parsimiseks kõige paremini [26]. Lisaks muudavad Beautiful Soupil suur kasutajaskond ja põhjalik dokumentatsioon selle õppimise ning võimalike probleemide lahendamise lihtsaks.

Storage Formatite võrdlemiseks kasutati Beautiful Soupi sisseehitatud meetodeid *find_all* ja *get_text*. Nende abil sai XHTML-ist lihtsasti kätte olulised elemendid nagu makrod, parameetrid, nende nimed ja lehe pealkirja. See oli oluline muudatuste tuvastamiseks ja kontrollimiseks, et värskendused toimuks õigesti.

4.5 Turvatestide raportite kavandamine

Selles peatükis käsitletakse turvatestide raportite vaadete olulisi aspekte nagu Confluence'i lehepuu (*page tree*) struktuur, ligipääsude haldamist, testandmete loomist, raportite disainimist ja nende valideerimist.

4.5.1 Automatiseeritud turvatestide raportite lehepuu

Confluence's on automatiseeritud turvatestide raportite jaoks loodud struktureeritud lehepuu. Joonisel 1 on näha, et lehepuu pealeht on „Automatiseeritud turvatestide raportid". Pealehe all lisatakse kõik asutused, millele turvatestimisi teostatakse. Iga

asutuse lehe alla luuakse testitavate rakenduste lehed, mille alla omakorda lisatakse turvatestide raportite lehed. Raportid on jaotatud vastavalt rollidele.

Joonis 1. Kuvatõmmis automatiseeritud turvatestide raportite lehepuu Confluence'is.

Selline lehepuu tagab loogilise struktureeritud ülesehituse, lihtsustab navigeerimist ning võimaldab tõhusamat ligipääsuõiguste haldamist.

4.5.2 Ligipääsude määramine

Turvatestide raportite ligipääsude määramine toimub Confluence'i keskkonnas vastava tööruumi sätete kaudu. Tööruumile on loodud kasutajate gruppide, individuaalsete kasutajate ja anonüümsete kasutajate õiguste komplektid. Need komplektid panevad paika, millised tegevused ja juurdepääsuõigused on erinevatel rollidel tööruumis lubatud, tagades nii turvalisuse kui ka rollipõhise ligipääsu.

4.5.3 Testandmete loomine

Testandmete loomise vajadus oli mitmeti põhjendatud, kuna tegu on turvavaldkonnas arendatava lahendusega. Autor ei saanud kasutada klientide pärisandmeid, sest oluline on tagada nende konfidentsiaalsus ja vältida võimalikke turvariske. Testandmete loomisega hoidutakse ka pärisandmete rikkumist Jiras arenduse ajal.

Autor lõi Jirasse näidistellimuspileti turvatestide projekti. Piletil täideti kõik vajalikud turvatestimise tellimuse eelduseks olevad väljad asutuse sisemise turvatestide tellimise juhendi järgi.

Tellimuspilet sisaldab üldist kirjeldust, mille alla on rakenduse tutvustus koos oluliste linkidega (Joonis 2). Tellimuse juurde on lisatud ka testimise üksikasjad nagu mitmendat korda testimine toimub, keskkonna link, OWASP ASVS (*Open Web Application Security*

Project Application Security Verification Standard) versioon, tase, testimise skoop ja tähtajad. Lisaks sisaldab tellimuspilet infot testijate kohta, kes testivad, mis asutusest ning kes kuuluvad testimismeeskonda. Samuti on välja toodud rakenduse andmed, kes on projektijuht, kes on arendaja, mis on rakenduse nimetus ja lühend. Tellimuspileti juurde lisatakse ka rahastusega seonduv info.

Joonis 2. Kuvatõmmis tellimuspileti üldandmete ja kirjelduse näidis.

Tellimuspileti külge on lisatud link näidisturvaleidude pileti juurde, millel on täpsustatud keskkond, rakenduse hostid, ligipääsud ja õigused ning lõplik hinnang turvatestimisele. Turvaleidude pileti alla on kirjeldatud iga turvaleiu pilet eraldi, et tagada selge ülevaade leitud turvaleidudest.

Turvaleiu pileti all on kirjeldatud leiu raskusaste, klassifikaator koos selle kirjeldusega, ASVS peatüki (*chapter*) ja sektsiooni (*section*) ID-d ning nimed, turvaleiu kirjeldus, risk, soovitused ja põhiturvatestimise kuupäev.

4.5.4 Turvatestide raportite mallide disainimine

Loodud testandmete pealt disainitakse kaks turvatestide raporti malli Confluence'i. Üks raport on suunatud infoturbejuhile, kelle rolliks on juhtida ja koordineerida organisatsiooni infoturbe strateegiat ning tagada IT-süsteemide ja andmete turvalisus. Teine raport on mõeldud DevOps insenerile, kes tegeleb tarkvaraarenduse ja IT-operatsioonide protsesside väljatöötamise, haldamise ning automatiseerimisega.

Turvatestide raportite vaadete disainimiseks kasutas autor olemasolevaid kättesaadavaid raporteid ja suhtles tellijaga, infoturbejuhiga ning turvatestimise tiimiga.

Kuna osapooled soovivad erinevaid vaateid, siis tuli autoril välja selgitada, mis on kellegi jaoks oluline ning lähtudes sellest disainis autor esialgu manuaalselt lehed Confluence'i. Nende lehtede pealt lõi autor vastavad YAML-failid igale rollile, mida saab hiljem muuta. YAML-failis on kasutatud Confluence Storage Formati koodi.

Autor analüüsis kättesaadavaid turvatestide raporteid ja valis välja need osad, mis tundusid kõige olulisemad. Seejärel tegi töö autor manuaalselt Confluence keskkonnas näidisraporti kasutades Confluence makrosid nagu Jira Issue/Filter, Jira Charts, Info, Note ja Anchor.

Manuaalselt valminud turvatestimise raporti loomine oli oluline samm, et selgelt mõista kõigi osapoolte soove ja nõudeid. Autor kasutas loodud näidisraporti Confluence Storage Formati koodi, et arendada turvatestide raporti automatiseerimise lahendust.

4.5.5 Valminud turvatestide raportite mallide valideerimine

Valminud raportite esialgsed versioonid valideeriti üle tellijaga, infoturbejuhiga ja turvatestimise meeskonnaga, kes andsid omapoolsed soovitused. Peamised ettepanekud puudutasid tabelites kuvatud teabe järjestuse muutmist ning turvaleiu struktuuri täpsustusi.

5 Teostus

Teostuse peatükis antakse ülevaade turvatestide raportite automatiseerimise lahenduse arendusest. Esmalt kirjeldatakse arenduse skoobi kitsendamist ja tekkinud probleeme. Antud peatükis tuuakse välja integratsioonilahenduse arendamine koos turvatestide raportite konfiguratsioonifailide ülevaatega. Täpsemalt selgitatakse tellimuspiletite leidmist, asutuse ja rakenduse loomist ning tellimuspiletist turvatestide raporti genereerimist. Peatükis tehakse ülevaade integratsioonilahenduse rakendamisest ja testimisest.

5.1 Skoobi kitsendamine ja arenduses tekkinud probleemid

Esialgu oli arenduse skoopi planeeritud lisada kordusturvatestide tulemused turvatestimise raportitele ja kasutada Jira veebihaake andmete automaatseks edastamiseks. Arenduse käigus tekkisid mitmed viivitused ja takistused, mis olid seotud asutuse siseste ligipääsude ning õigustega.

Kordusturvatestimise tellimuste käsitlemine otsustati lahenduse skoobist välja jätta, kuna see lisas lahendusele märkimisväärset keerukust. See nõudis meeskonnaga põhjalikumaid arutelusid ja täiendavaid otsuseid, mida antud ajaraamistikus ei olnud võimalik ellu viia.

Lahenduses oli algselt planeeritud kasutada Jira veebihaake, et automatiseerida andmete edastamine. Veebihaak on sündmustepõhine side, mis saadab andmeid rakenduste vahel automaatselt HTTP kaudu [27]. Jiras loodi kuulaja (*listener*), mis konfigureeriti nii, et see käivituks iga uue turvatestimise tellimuse loomisel. Vajalikud võrguligipääsud seadistati korrektselt, kuid veebihaagid ei jõudnud sihtkohta. Täiendavalt testiti käsurealt cURL-i (*Client Uniform Resource Locator*) abil saadetud päringuid, mis jõudsid sihtkohta probleemideta, viidates sellele, et probleem ei olnud võrguühenduses, vaid tõenäoliselt Jira rakenduse konfiguratsioonis.

Autor ei saanud ka piisavalt õigusi, et näha Confluence Storage Formatit, kuna see on ligipääsetav ainult keskkonna administraatoritele ning seda funktsionaalsust ei saa eraldi lubada. See tähendas, et autor pidi Storage Formati sisu kättesaamiseks alati pöörduma

vastava rolliga kolleegi poole. Selline piirang vajab tulevikus lahendamist, sest Storage Format on aluseks uute raportite kujundamisel.

5.2 Integratsioonilahenduse arendamine

Antud peatükis antakse ülevaade turvatestide raportite automatiseerimise lahenduse arendusest Jira ja Confluence vahel. Esmalt selgitakse, millised on turvatestide raportite konfiguratsioonifailid. Seejärel kirjeldatakse, kuidas leitakse tellimuspiletid Jirast ja mis nendest saab ning kuidas luuakse raportid Confluence'i.

5.2.1 Turvatestide raportite konfiguratsioonifailid

Turvatestide raportite genereerimiseks loodud rollipõhised konfiguratsioonifailid asuvad projekti *pageconfigs* kaustas ja on mõeldud lehtede vaadete disainimiseks Confluence'is. Antud konfiguratsioonifailid on YAML-formaadis (näidis leitav Lisa 2 alt).

Kõikide turvatestide raportite YAML failid sisaldavad vastava lehe pealkirja ja legendi, kuidas raportit lugeda, ning turvaleidude kirjeldusi. Pealkiri on alati esimene väärtus failis. Legend on paigutatud alati lehe lõppu. Confluence'is on kasutatud Note makrot, et legendile lehe ülemises osas viidata. Legendile endale on lisatud Anchor makro, millega saab lehe sees viitamisi teha täpselt kohale, kus on Anchor lisatud. Turvaleiu kirjeldused on alati kindla struktuuriga ja paigutatakse õige ASVS peatüki alla järjestatult klassifikaatori numbri alusel. Turvaleiul kuvatakse Jira pileti nimi, millele saab klikkides liikuda pileti enda juurde. Lisaks on leitavad leiu klassifikaatori number koos kirjeldusega, põhiturvatestimise kuupäev, leiu raskusaste, kirjeldus, riskid ja soovitused. ASVS peatükkides, mille all leide ei ole, kuvatakse tekst: „Leiud puuduvad, pole skoobis või pole testitud.".

Erinevatel rollidel on spetsiifilisi osasid, mida raportis kuvatakse, näiteks infoturbejuhile mõeldud turvatestide raport on üks informatiivsemaid, kuna sisaldab testimise infot, ajaperioodi, testijate meeskonda, põhiturvatestimise tulemuse koondtabelit, hinnangut, turvaleidude koondtabelit ja leide eraldi. Osade rollide raportites on erilisi lähenemisi nagu turvatestide raport DevOpsile sisaldab infot ainult turvaleidude tabelist, mis on lahendamata ja nende staatustega leidude kirjeldusi. Parandatud leiud kaovad nii ülevaatlikust tabelist kui ka leidude loetelust.

5.2.2 Tellimuspiletite leidmine

Uue tellimuspileti puhul kasutatakse programmis *queue_new_test_issuses*, mis pannakse käima kõige kõrgema taseme meetodis *run*. Meetod leiab uued piletid JQL (*Jira Query Language*) abil (Joonis 3).

```
security_test_issues = self.jira.jql(f"project = PROJEKTINIMI and issuetype =
\"Security test\" and created > \"{scan_info['createdAt']}\" ")
```

Joonis 3. Uute tellimuspiletite muutuja ja kasutatav JQL.

Leitud piletid käiakse läbi ning lisatakse andmebaasi SecurityTest olemina.

5.2.3 Asutuse ja rakenduse loomine

Asutuste jaoks on projektis *config* kaustas loodud fail, kus on asutuste lühendid vastavusse viidud nende täispikkade nimedega. Confluence lehepuus ei kuvata lühendeid asutuste tasandil.

Asutuse info saadakse tellimuspiletist *handle_organization_page* meetodiga, kus loetakse sisse asutuste konfiguratsioonifaili, kus on kirjeldatud lühendid ja nendele vastav täispikk nimi. Meetodis vastandatakse sõnastiku võtit tellimuspiletist saadud asutuse lühendiga ning lisatakse Confluence'i vastava asutuse leht *handle_security_test_page* meetodis kui see pole juba eelnevalt lisatud.

Rakenduse info võetakse tellimuspiletist *handle_application_page* meetodiga, kus leitakse kindel väli ja tagastatakse see. Rakenduse leht luuakse samuti *handle_security_test_page* meetodiga.

5.2.4 Tellimuspiletist turvatestide raporti genereerimine

Arenduses on kasutatud regulaaravaldist (Joonis 4), et asendada YAML-failis konkreetsed väärtused (*placeholder*). Joonis 4 näites on YAML-failis asendatud vastava sõnastiku *template* väärtuses *security_bug.key* element.

```
result = re.sub(r'\{security_bug.key}', security_bug_issue['key'],
section['template'])
```

Joonis 4. Regulaaravaldisega YAML-failis väärtuse asendamise näide.

Meetod *handle_security_test_page* teeb turvatestide raporti vaate Confluence'i. See meetod lisab uue raporti lehe kui ka uuendab olemasolevat. *handle_security_test_page* meetod kasutab mitmeid teisi meetodeid nagu *get_page_title*, *handle_organization_page*,

handle_application_page, *render_section_single*, *render_section_multiple*, *build_page_data* ja *handle_existing_page*.

Meetodis *get_page_title* otsitakse lehe konfiguratsioonifailist pealkirja väärtus ja lisatakse juurde rakenduse nime lühend ning kuupäev. See lahendus on selleks, et saaks sama lehe konfiguratsioonifaili kasutada kõigi rakenduste jaoks ja et tekiks unikaalne lehe pealkiri, sest Confluence ei lase samas tööruumis lehti ühtemoodi nimetada.

Meetodite *handle_organization_page* ja *handle_application_page* tööpõhimõtted on kirjeldatud peatükis 5.2.3.

Meetodid *render_section_single* ja *render_section_multiple* tegelevad YAML-failis kindlate osadega. *render_section_single* meetod asendab YAML-failis piletite võtmete ajutised elemendid. *render_section_multiple* meetod tegeleb leidude kirjeldamise Storage Formatiga, mida lisatakse raportisse korduvalt. Selles meetodis lisatakse ASVS peatükid ning paigutatakse Jiras kirjeldatud leiud järjestatult õigete peatükkide all. Lisaks lisatakse leidudeta peatükkide alla informatiivne tekst, mis viitab leidude puudumisele.

Meetodis *build_page_data* määratakse lehe andmed nagu lehe tüüp, tööruum, pealkiri, eellehed, sisu ja versioon.

Olemasoleva turvatestide raporti lehe uuendamiseks kasutatakse meetodit *handle_existing_page*, mis võrdleb olemasoleva lehe ja loodava lehe sõnastikuks parsitud XHTML-e. Sõnastikeks parsimine tagab täpse võrdluse, kuna XHTML-is võib atribuutide järjestus muutuda.

Lisa 3 on leitavad loodud turvatestide raporti osade näidised, mis visualiseerivad hästi, milline näeb välja valmislahendus.

5.3 Integratsioonilahenduse rakendamine

Valminud lahendus integreeriti tööprotsessi kasutades Gitti, projektimallis defineeritud CI/CD töövoogu, Dockerfile'i, docker-compose faile ja asjastatud tööde (*cron job*) plaani töövoo ajakavas (*schedule*).

Autor arendas lahendust Gitis loodud projekti eraldi harus (*branch*), mis tagab arendustöö organiseerituse ning hoiab peaharu puhtana. Arendusharu (*dev branch*) on selleks, et

tsentraliseerida arenduses olulised tegevused nagu uued funktsioonid, parendused, veaparandused ja testimine. See haru tagab arenduse stabiilsuse, kuna vähendab vigade sattumist tootesse. Kui lahendus on arendusharus testitud ning kontrollitud, siis liidetakse (*merge*) arendusharu peaharusse (*main*).

Projektimallis on defineeritud geneerilised CI/CD töövoog, Dockerfile, docker-compose failid. Töövoos luuakse projektis Dockeri tõmmis, mis lisatakse konteinerite registrisse. Seejärel käivitatakse töövoos *deploy*-samm, mille käigus tõmmatakse konteinerite registrist lisatud tõmmis Docker hostile ja luuakse konteiner vastavalt docker-compose failis defineeritud seadistustest. Töövoo ajakava kaudu aktiveeritakse loodud konteiner vastavalt ajastatud tööde plaanile.

Cron-süntaksit kasutatakse tööde ajastamiseks, et tagada regulaarne töötlus ja ajakohased toimingud, nagu näiteks teenuse või konteineri käivitamine automaatselt määratud ajahetkedel. See aitab hoida töövoogu sujuvana ja automatiseerida arenduse, testimise ja juurutamise etappe.

5.4 Lahenduse testimine

Peale lahenduse integreerimist tööprotsessi testiti loodud lahendust testandmetega. Selleks loodi uued testandmetega tellimuspiletid Jirasse, et kontrollida, kas programm tuvastab uued tellimused. Automatiseerimine oli ajastatud kindlaks kellaajaks, kuid selleks, et lahendust kohe jooksutada ning tagasisidet saada, käivitati lahendus manuaalselt. Peale käivitamist saab tulemusi kontrollida Confluence'i tööruumist, kas tellimusse märgitud asutuse alla loodi rakendus ning turvatestimise raport. Peale igat käimist edastab lahendus Microsoft Teamsi kanalisse teate, kas protsess jooksis edukalt või tõi see kaasa vea.

Selleks, et kontrollida turvaleidude leidmist tuli luua testandmetega turvaleidude pilet ja see linkida tellimuspileti külge ning uuesti programmi jooksutada. Peale edukat käivitamist sai kontrollida, kas turvaleidude piletist kuvatakse info raportis. Lahendus kontrollib ka lingitud pileti tüüpi, mille kontrolliks linkis autor turvatestide tellimuse piletile teist tüüpi pileti ning kontrollis ka ilmneb veateade. Selline lahendus on selleks, et tagada vaid vajalike andmete kasutamist ning vältida vigade tekkimist.

Järgmise sammuna kirjeldati testandmetega turvaleiud turvaleidude pileti alla ja käivitati taas programm, et kontrollida turvaleidude kuvamist raportis. Kui turvaleidusid pole eelnevalt Jirasse kirjeldatud kuvatakse turvatestide raportis ASVS peatükke kirjega, mis viitab, et testi tulemusi veel pole.

Autor lisas ka turvatestide raportite jaoks uusi konfiguratsioonifaile, et kontrollida lahenduse võimet neid õigesti lugeda ja töödelda.

Peale neid samme edastas autor lahenduse tellijale ja turvatestimise meeskonnale, kes vaatasid lahenduse üle ning andsid endapoolse hinnangu. Tellija ja turvatestimise meeskonna liikmete arvates vastas lahendus ootustele ning täitis vajalikud eeldused, et turvatestide raporteid automaatselt genereerida.

6 Tulemused

Tulemused peatükis tehakse ülevaade lõputöö tulemustest lähtudes lahenduse eesmärgist automatiseerida turvatestide raportite genereerimine ja seatud nõuete täitmisest. Lisaks tuuakse välja planeeritavad edasiarendused.

6.1 Lahenduse eesmärkide täitmine

Tulemusi testides selgus, et autor täitis peatükis 3.5 on kirjeldatud lahenduse eesmärgid. Turvatestide raportite automatiseerimise lahendus vähendab manuaalset tööd ja inimlike vigade tekkimise võimalust andmete koondamisel ning edastamisel. Automatiseerimine kiirendab tulemuste kättesaadavust kõigile asjaosalistele ühest kohast. Lahendusega loodi standardiseeritud ja selgelt struktureeritud rollipõhised raportid, mis on kergesti loetavad ning sisaldavad kõiki vajalikke detaile turvatestide kohta nagu tellimuseinfo, turvaleidude kirjeldused, leidude prioriteetsus ja staatus, soovituslikud tegevused turvavea lahendamiseks.

6.1.1 Turvatestide raporti koostamise meetodite kokkuvõte

Automatiseeritud turvatestide raportite genereerimise lahendusega optimeeriti organisatsiooni ressursikasutust. Ühe turvaleiu kirjeldamine Jirasse võtab keskmiselt aega 30 minutit. Tabelis 2 on toodud võrdlus kolme erineva meetodi vahel, mida hinnati järgmiste kriteeriumite alusel: raportite koostamise aeg (10 turvaleiuga raporti puhul), andmete täpsus, sihtrühmade kasutusmugavus ja ühilduvus olemasolevate süsteemidega.

Esiteks saab Jirasse kirjeldatud turvaleide kopeerida käsitsi Microsoft Wordi dokumenti ja selle salvestamine PDF-failina. See on aeganõudev, kuna Jira piletides on andmed erinevate väljade peal ja nende kopeerimine on tülikas. Kümne leiuga turvatestide raporti koostamine võtab aega ligikaudu 3 tundi. Selline manuaalne lähenemise tõttu on andmete täpsus madal, kuna Jiras võib aja jooksul toimuda turvaleiu info uuendamist ja need täpsustused tuleks manuaalselt üle kontrollida koostatud raportis. Turvatestide raportite kasutusmugav sihtrühmadele on keskmine või madal, kuna raportid edastatakse krüpteeritult PDF-formaadis ja nende uuesti leidmise mugavuse eest vastutab osapool ise.

Käsitsi turvatestide raportite koostamise ühilduvus olemasolevate süsteemidega on madal, kuna need on iseseisvad ning pole ühildatud ühegi olemasoleva süsteemiga.

Teiseks saab koostada manuaalselt Confluence tööruumi õige asutuse ning rakenduse alla lehe kasutades makrosid. Confluence makrod aitavad Jirasse kirjeldatud piletite pealt lihtsasti infot kuvada ning ei ole vaja käsitsi kopeerida. See on poolautomaatne lahendus, kus info uueneb, kuid lehe struktuuri tuleb siiski manuaalselt uuendada. Kümne leiuga turvatestide raporti koostamine selliselt võtab aega ligikaudu 1,5 tundi. Selle meetodi puhul on andmete täpsus kõrge, kuna Jiras tehtud muudatused kuvatakse automaatselt Confluence lehel. Turvatestide raportite kasutusmugav sihtrühmadele on kõrge, kuna kõik turvatestide raportid on leitavad ühest kohast ja osapooled ei pea dekrüpteerimisega tegelema või nende hoiustamisega. Selline turvatestide raportite koostamise meetodi ühilduvus olemasolevate süsteemidega on keskmine, kuna need küll impordivad infot makrode abil Jirast, kuid lehtede kokkupanemine toimub siiski manuaalselt.

Kolmandaks on automaatikaga turvatestide raporti genereerimine. See lahendus on kõige kiirem, võtab aega kümne leiuga turvatestide raporti genereerimisel ligikaudu 2 sekundit, kuna automaatikalahendus teeb kõik vajalikud tegevused ise ära. Automaatselt turvatestide raportite genereerimisel on andmete täpsus kõrge, kuna integratsioonilahendus kontrollib tehtud uuendusi ja uuendab raporteid vastavalt ning tagab andmete värskuse. Sihtrühmade jaoks on kasutusmugavus kõrge, sest kõik turvatestide raportid on leitavad ühest kohast. Automaatne lahenduse ühilduvus olemasolevate süsteemidega on kõrge, kuna see toetab juba kasutusel olevaid töövooge ja sobib hästi edasiarendusteks.

Tabel 2. Turvatestide raporti koostamise meetodite ülevaade.

Raporti koostamise meetod	**Raporti koostamise aeg**	**Andmete täpsus**	**Sihtrühmade kasutus-mugavus**	**Ühilduvus olemasolevate süsteemidega**
Käsitsi Microsoft Wordi dokumendina, mis salvestatakse PDF-ina	~ 3 tundi	Madal	Keskmine või madal	Madal

Manuaalselt Confluence lehe loomine	~ 1,5 tundi	Keskmine	Kõrge	Keskmine
Automaatikaga genereerimine	~ 2,11 sekundit	Kõrge	Kõrge	Kõrge

6.2 Nõuete täitmine

Loodava lahenduse funktsionaalsed ja mittefunktsionaalsed nõuded on kirjeldatud peatükis 4.2. Selles peatükis hinnatakse, kas seatud nõuded said lahenduse teostuse lõpuks täidetud või mitte.

Kirjeldatud funktsionaalsetest nõuetest täidab lahendus automaatsete Confluence lehtede loomist iga Jiras registreeritud turvatestimise tellimuspileti pealt ja genereerib turvatestide raporti õige asutuse rakenduse alla ning raportid sisaldavad tellimuseinfot, turvaleidude kirjeldusi, leidude prioriteetsust ja staatust, soovitatavaid tegevusi riskide maandamiseks.

Turvatestide raportite automatiseerimise lahendus täidab mittefunktsionaalsete nõuetest andmete reaalajas uuendamist, paindlikust edasiarenduste osas ja turvatestimise tiimil on võimalus kohandada raportite malli vastavalt vajadustele.

6.3 Planeeritud edasiarendused

Lõputöö raames tehtud lahendusele on planeeritud ka edasiarendused. Turvatestide raportite poolelt on olulisem edasiarendus seotud kordusturvatestimise tulemuste lisamisega ja üldises vaates on plaan täita ära ka Confluence'i lehepuus olevate lehtede vaated. Nende vaadete täitmine on tähtis just suurema ülevaate saamiseks ning mõistmaks, millised on tüüpilised turvavead, mida arenduses tehakse jms.

Tulevikus on planeeritud asutuse lehtedel kuvada asutusega seonduvat olulist infot ning rakenduste statistikat. Asutuse ülevaate leht peab sisaldama palju turvatestide leide on olnud (arv iga prioriteedi kohta), palju on hetkel avatud/lahendamata leide (arv iga prioriteedi kohta), iga rakenduse kohta – millal oli viimane testimine, palju leide on

vaatamise hetkel veel aktuaalsed (arv iga prioriteedi kohta), iga äriprotsessi kohta – palju on ärirakendust toetavatel teenustel kokku lahendamata leide, iga kliendi kohta eraldi vaade, mis sisaldab palju on hetkel avatud/lahendamata leide (arv iga prioriteedi kohta), iga rakenduse kohta – millal oli viimane testimine, palju leide on vaatamise hetkel veel aktuaalsed (arv iga prioriteedi kohta), iga äriprotsessi kohta- palju on ärirakendusest toetavatel teenustel kokku lahendamata leide.

Rakenduse lehele hakatakse kuvama rakenduse olulist infot ning statistikat turvatestide kohta. Leht peab kuvama ülevaateid palju turvatestide leide on olnud (arv iga prioriteedi kohta), palju on hetkel avatud/lahendamata leide (arv iga prioriteedi kohta). Iga testimise korra kohta on info millal algas, lõppes, kes testis, palju oli leide (arv prioriteedi kohta), palju on leide vaatamise hetkel veel aktuaalsed (arv iga prioriteedi kohta).

Iga asutuse vaate all on eraldi äriprotsessi kohta oma leht. Äriprotsessi leht on oma struktuurilt sama, mis asutuse vaade, aga info on piiratud ühe äriprotsessi rakendustele.

7 Kokkuvõte

Käesoleva lõputöö eesmärk oli luua turvatestide raportite automatiseerimise lahendus. Turvatestide raportid on olulised, kuna need annavad ülevaate rakenduse turvalisusest ja aitavad hinnata süsteemi vastupidavust välistele ohtudele. Nendest saadud info põhjal saavad erinevad osapooled teha otsuseid parandusmeetmete valiku osas ja hinnata rakenduse valmidust tootmiskeskkonda paigaldamiseks. Turvatestide raportid annavad ülevaate võimalikest riskidest, mis võivad mõjutada rakenduse usaldusväärsust ja kasutajate turvalisust.

Lahenduse tulemused ja saadud tagasiside kinnitavad, et lõputöö raames püstitatud eesmärgid ja nõuded said täidetud. Protsessi efektiivsuse tõstmiseks automatiseeriti turvatestide raportite loomine. Turvatesti tellimuspileti pealt kuvatakse Confluence'i turvatestide raportit õige asutuse ning rakenduse alla.

Turvatestide raportite automatiseerimise lahendus on aluseks paljudele planeeritud edasiarendustele, mis täiustavad antud lahendust ja tagavad ressursside kokkuhoidu ka tulevikus.

Kasutatud kirjandus

[1] H. P. Mythili Raju, „What Is Security Testing: With Examples And Best Practices,“ 25. september 2023. [Võrgumaterjal]. Available: https://www.lambdatest.com/learning-hub/security-testing. [Kasutatud 2. detsember 2024].

[2] A. S. G. Mary K. Pratt, „Workflow automation,“ 27. september 2023. [Võrgumaterjal]. Available: https://www.techtarget.com/searchcontentmanagement/definition/workflow-automation. [Kasutatud 2. detsember 2024].

[3] N. Lamb, „What Are Integrated Solutions?,“ 21. august 2024. [Võrgumaterjal]. Available: https://rcntechnologies.com/what-are-integrated-solutions/#:~:text=At%20the%20core%20of%20its,streamline%20processes%2C%20and%20improve%20efficiency. [Kasutatud 2. detsember 2024].

[4] M. B. J. D. B. B. P. Michael Felderer, Advances in Computers, Elsevier, 2016.

[5] G. Leme, „What Is Workflow Automation? 2024 Guide,“ 14. august 2024. [Võrgumaterjal]. Available: https://www.pipefy.com/blog/what-is-workflow-automation/#:~:text=Workflow%20automation%20refers%20to%20the,manual%20work%20and%20repetitive%20tasks.. [Kasutatud 2. detsember 2024].

[6] K. Niglas, „Praktikale suunatud uuringudisainid,“ 30 august 2012. [Võrgumaterjal]. Available: https://www.tlu.ee/~katrin/seminar/Alternatiivsed_disainid.pdf. [Kasutatud 2. detsember 2024].

[7] „Welcome to Jira,“ Atlassian, [Võrgumaterjal]. Available: https://www.atlassian.com/software/jira/guides/getting-started/introduction#what-is-jira-software . [Kasutatud 2. detsember 2024].

[8] „Great outcomes start with Jira,“ Atlassian, [Võrgumaterjal]. Available: https://www.atlassian.com/software/jira . [Kasutatud 2. detsember 2024].

[9] „What is Confluence Cloud?,“ Atlassian, 9. september 2024. [Võrgumaterjal]. Available: https://support.atlassian.com/confluence-cloud/docs/what-is-confluence-cloud/. [Kasutatud 2. detsember 2024].

[10] „Confluence basics,“ Atlassian, [Võrgumaterjal]. Available: https://www.atlassian.com/software/confluence/resources/guides/get-started/overview#about-confluence. [Kasutatud 2. detsember 2024].

[11] „Confluence Storage Format,“ Atlassian, 14. veebruar 2024. [Võrgumaterjal]. Available: https://confluence.atlassian.com/doc/confluence-storage-format-790796544.html . [Kasutatud 2. detsember 2024].

[12] „XHTML Introduction,“ GeeksforGeeks, 10. jaanuar 2024. [Võrgumaterjal]. Available: https://www.geeksforgeeks.org/xhtml-introduction/. [Kasutatud 2. detsember 2024].

[13] T. P. Team, „What Is a REST API? Examples, Uses, and Challenges,“ 9. juuli 2020. [Võrgumaterjal]. Available: https://blog.postman.com/rest-api-examples/ . [Kasutatud 7. detsember 2024].

[14] „URI,“ AKIT, [Võrgumaterjal]. Available: https://akit.cyber.ee/term/1454-uri. [Kasutatud 7. detsember 2024].

[15] „Jira Data Center platform REST API reference,“ Atlassian, [Võrgumaterjal]. Available: https://docs.atlassian.com/software/jira/docs/api/REST/9.17.0/ . [Kasutatud 7. detsember 2024].

[16] „Confluence REST API Documentation,“ Atlassian, [Võrgumaterjal]. Available: https://docs.atlassian.com/atlassian-confluence/REST/6.6.0/ . [Kasutatud 7. detsember 2024].

[17] „Functional vs. Non Functional Requirements,“ GeeksforGeeks, 22. oktoober 2024. [Võrgumaterjal]. Available: https://www.geeksforgeeks.org/functional-vs-non-functional-requirements/. [Kasutatud 2. detsember 2024].

[18] J. Zelle, Python Programming: An Introduction to Computer Science (Second Edition), Franklin, Beedle & Associates Incorporated, 2010.

[19] B. Marji, „Webhooks in Python with Flask - The Python Code,“ 20. juuni 2021. [Võrgumaterjal]. Available: https://thepythoncode.com/article/webhooks-in-python-with-flask. [Kasutatud 31. detsember 2024].

[20] „Welcome to Flask,“ [Võrgumaterjal]. Available: https://flask.palletsprojects.com/en/stable/. [Kasutatud 31. detsember 2024].

[21] M. Makai, „Object-relational Mappers (ORMs) - Full Stack Python,“ [Võrgumaterjal]. Available: https://www.fullstackpython.com/object-relational-mappers-orms.html. [Kasutatud 31. detsember 2024].

[22] S. Yegulalp, „The best ORMs for database-powered Python apps,“ 15. november 2023. [Võrgumaterjal]. Available: https://www.infoworld.com/article/2335270/6-orms-for-every-database-powered-python-app.html. [Kasutatud 31. detsember 2024].

[23] „PostgreSQL, SQLAlchemy 2.0 Documentation,“ 27. detsember 2024. [Võrgumaterjal]. Available: https://docs.sqlalchemy.org/en/20/dialects/postgresql.html. [Kasutatud 31. detsember 2024].

[24] A. Stevenson, „XML vs. JSON vs. YAML,“ 11. november 2022. [Võrgumaterjal]. Available: https://community.cisco.com/t5/devnet-general-knowledge-base/xml-vs-json-vs-yaml/ta-p/4729758. [Kasutatud 01. jaanuar 2025].

[25] A. S. Ayub, „XML, YAML, JSON; Differences and Similarities,“ 14 mai 2024. [Võrgumaterjal]. Available: https://dev.to/anwar_sadat/ml-yaml-json-differences-and-similarities-3n17. [Kasutatud 1. jaanuar 2025].

[26] L. Richardson, „Beautiful Soup Documentation,“ [Võrgumaterjal]. Available: https://www.crummy.com/software/BeautifulSoup/bs4/doc/. [Kasutatud 27. detsember 2024].

[27] „What is a webhook?,“ Red Hat, 1. veebruar 2024. [Võrgumaterjal]. Available: https://www.redhat.com/en/topics/automation/what-is-a-webhook. [Kasutatud 2. detsember 2024].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks[1]

Mina, Kerli Saarniit

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Turvatestide raportite automatiseerimine“, mille juhendaja on Einar Kivisalu
   1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
   1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

06.01.2025

1 Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingulise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 – Turvatestide raporti infoturbejuhi YAML-faili näidis

```
title: "{security_test.customfield_10313} Turvatestimise raport
infoturbejuhile ({security_test.duedate})"
sections:
  - section_type: single
    template: >-
      <ac:layout>
      <ac:layout-section ac:type="single"><ac:layout-cell>
      <h2><strong>Testimise info</strong></h2>
      <p><ac:structured-macro ac:name="jira" ac:schema-version="1" ac:macro-
id="bd590092-88dd-4943-ad5c-626f2c852310"><ac:parameter
ac:name="server">Internal Jira</ac:parameter><ac:parameter
ac:name="columnIds">customfield_10362,customfield_10312</ac:parameter><ac:par
ameter ac:name="columns">Organization,Nimetus</ac:parameter><ac:parameter
ac:name="maximumIssues">20</ac:parameter><ac:parameter ac:name="jqlQuery">key
= {security_test.key}   </ac:parameter><ac:parameter
ac:name="serverId">xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx</ac:parameter></ac:structured-macro></p>
      <p><strong><ac:structured-macro ac:name="jira" ac:schema-version="1"
ac:macro-id="0f26598d-e6bf-48c1-ac89-96286fe06269"><ac:parameter
ac:name="server">Internal Jira</ac:parameter><ac:parameter
ac:name="columnIds">customfield_10200,customfield_10304</ac:parameter><ac:par
ameter ac:name="columns">Keskkond,Lingid
keskkondadele</ac:parameter><ac:parameter
ac:name="maximumIssues">20</ac:parameter><ac:parameter ac:name="jqlQuery">key
= {security_test.key}    </ac:parameter><ac:parameter
ac:name="serverId">xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx</ac:parameter></ac:structured-macro></strong></p>
      <p><strong><ac:structured-macro ac:name="jira" ac:schema-version="1"
ac:macro-id="dc80b5db-4fe6-47f5-bcea-3b80490f6e59"><ac:parameter
ac:name="server">Internal Jira</ac:parameter><ac:parameter
ac:name="columnIds">customfield_11201,customfield_11202</ac:parameter><ac:par
ameter ac:name="columns">OWASP ASVS versioon,OWASP ASVS
tase</ac:parameter><ac:parameter
ac:name="maximumIssues">20</ac:parameter><ac:parameter
ac:name="jqlQuery">key={security_test.key} </ac:parameter><ac:parameter
ac:name="serverId">xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx</ac:parameter></ac:structured-macro></strong></p>
      <h3><strong>Testimise ajaperiood</strong></h3>
      <p><strong><ac:structured-macro ac:name="jira" ac:schema-version="1"
ac:macro-id="92eda0bc-2dba-41a3-9a33-9f9f6aa2575b"><ac:parameter
ac:name="server">Internal Jira</ac:parameter><ac:parameter
ac:name="columnIds">customfield_10301,duedate</ac:parameter><ac:parameter
ac:name="columns">Valmiduse kuupäev,due</ac:parameter><ac:parameter
ac:name="maximumIssues">20</ac:parameter><ac:parameter ac:name="jqlQuery">key
= {security_test.key}  </ac:parameter><ac:parameter
ac:name="serverId">xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx</ac:parameter></ac:structured-macro></strong></p>
      <h3><strong>Testijad</strong></h3>
```

<ac:structured-macro ac:name="jira" ac:schema-version="1" ac:macro-id="f4f35ebd-6f7f-4481-8110-df5e5235caad"><ac:parameter ac:name="server">Internal Jira</ac:parameter><ac:parameter ac:name="columnIds">customfield_10309,customfield_10311</ac:parameter><ac:parameter ac:name="columns">Firma,Meeskond</ac:parameter><ac:parameter ac:name="maximumIssues">20</ac:parameter><ac:parameter ac:name="jqlQuery">key = {security_test.key} </ac:parameter><ac:parameter ac:name="serverId">xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx</ac:parameter></ac:structured-macro>

 </ac:layout-cell></ac:layout-section><ac:layout-section ac:type="single"><ac:layout-cell>
<h2>Põhiturvatestimise tulemused</h2>
<ac:structured-macro ac:name="jirachart" ac:schema-version="1" ac:macro-id="f43c776a-a1d0-42a9-94ba-5a66a532cc98"><ac:parameter ac:name="server">Internal Jira</ac:parameter><ac:parameter ac:name="sortDirection" /><ac:parameter ac:name="jql">parent%20%3D%20{security_bug.key}%20and%20type%20%3D%20'Bug%20sub-task'%20%20AND%20resolution%20%3D%20Unresolved%20and%20status%20!%3D%20Canceled</ac:parameter><ac:parameter ac:name="ystattype">customfield_11002</ac:parameter><ac:parameter ac:name="chartType">twodimensional</ac:parameter><ac:parameter ac:name="width" /><ac:parameter ac:name="sortBy" /><ac:parameter ac:name="isAuthenticated">true</ac:parameter><ac:parameter ac:name="numberToShow">5</ac:parameter><ac:parameter ac:name="xstattype">customfield_10587</ac:parameter><ac:parameter ac:name="serverId">xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx</ac:parameter></ac:structured-macro> </ac:layout-cell></ac:layout-section><ac:layout-section ac:type="single"><ac:layout-cell>
<h2>Hinnang</h2>
<ac:structured-macro ac:name="jira" ac:schema-version="1" ac:macro-id="69bd43c7-6f8d-4ffd-961f-53af4cd058ba"><ac:parameter ac:name="server">Internal Jira</ac:parameter><ac:parameter ac:name="columnIds">customfield_11217</ac:parameter><ac:parameter ac:name="columns">Hinnang</ac:parameter><ac:parameter ac:name="maximumIssues">20</ac:parameter><ac:parameter ac:name="jqlQuery">key = {security_bug.key}   </ac:parameter><ac:parameter ac:name="serverId">xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx</ac:parameter></ac:structured-macro> </ac:layout-cell></ac:layout-section><ac:layout-section ac:type="single"><ac:layout-cell><ac:structured-macro ac:name="note" ac:schema-version="1" ac:macro-id="18a6ea30-8849-4d16-97d3-c0f195812fca"><ac:parameter ac:name="title">Leiud, riskihinnangud, metoodika ja leiu raskusastmete legend</ac:parameter><ac:rich-text-body>
Dokumendis kasutatud leidude elementide, riskihinnangu, metoodika ja leiu raskusatmete legendiga saab tutvuda dokumendi lõpus peatükis: <ac:link ac:anchor="Leiud, riskihinnangud, metoodika ja leiu raskusastmete legend" />. </ac:rich-text-body></ac:structured-macro></ac:layout-cell></ac:layout-section><ac:layout-section ac:type="single"><ac:layout-cell>
<h2> Turvaleiud</h2>

```
      <p><ac:structured-macro ac:name="jira" ac:schema-version="1" ac:macro-
id="baa94063-90b0-4094-bb9e-ea75c1d46948"><ac:parameter
ac:name="server">Internal Jira</ac:parameter><ac:parameter
ac:name="columnIds">issuekey,description,customfield_11002,customfield_11207,
customfield_10587</ac:parameter><ac:parameter
ac:name="columns">key,description,Chapter Name,Põhiturvatestimise
kuupäev,Leiu raskusaste</ac:parameter><ac:parameter
ac:name="maximumIssues">20</ac:parameter><ac:parameter
ac:name="jqlQuery">project = Turvatestid AND parent = {security_bug.key}
</ac:parameter><ac:parameter ac:name="serverId">xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx</ac:parameter></ac:structured-macro></p></ac:layout-
cell></ac:layout-section>
  - section_type: multiple
    foreach_in_jql: project = PROJEKTINIMI AND issue in
subtaskIssuesFromQuery("issue in
linkedIssues('{security_test.jira_issue_id}') and issuetype = 'Security
Bug'")
    group_by_custom: chapters
    template: >-
      <ac:structured-macro ac:name="info" ac:schema-version="1" ac:macro-
id="38e9de9e-ddd4-4770-a337-74950dc6d493"><ac:parameter
ac:name="icon">false</ac:parameter><ac:rich-text-body>
      <h1><ac:structured-macro ac:name="jira" ac:schema-version="1" ac:macro-
id="dab34e36-7ab5-4d69-ab39-c713684bf83e"><ac:parameter
ac:name="server">Internal Jira</ac:parameter><ac:parameter
ac:name="columnIds">summary</ac:parameter><ac:parameter
ac:name="columns">summary</ac:parameter><ac:parameter
ac:name="maximumIssues">20</ac:parameter><ac:parameter ac:name="jqlQuery">key
= {} </ac:parameter><ac:parameter ac:name="serverId">xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx</ac:parameter></ac:structured-macro></h1>
      <p><em><ac:structured-macro ac:name="jira" ac:schema-version="1"
ac:macro-id="c9b9d827-6c43-495d-93a1-68355ce3e321"><ac:parameter
ac:name="server">Internal Jira</ac:parameter><ac:parameter
ac:name="columnIds">customfield_10302,customfield_11005</ac:parameter><ac:par
ameter ac:name="columns">Leiu klassifikaator,Leiu klassifikaatori
kirjeldus</ac:parameter><ac:parameter
ac:name="maximumIssues">20</ac:parameter><ac:parameter ac:name="jqlQuery">key
= {}   </ac:parameter><ac:parameter ac:name="serverId">xxxxxxxx-xxxx-xxxx-
xxxx-xxxxxxxxxxxx</ac:parameter></ac:structured-macro></em></p>
      <p><ac:structured-macro ac:name="jira" ac:schema-version="1" ac:macro-
id="e3ba9f5e-dae3-4b3b-bdcd-40c99336fc91"><ac:parameter
ac:name="server">Internal Jira</ac:parameter><ac:parameter
ac:name="columnIds">customfield_11207,customfield_10587</ac:parameter><ac:par
ameter ac:name="columns">Põhiturvatestimise kuupäev,Leiu
raskusaste</ac:parameter><ac:parameter
ac:name="maximumIssues">20</ac:parameter><ac:parameter ac:name="jqlQuery">key
= {}   </ac:parameter><ac:parameter ac:name="serverId">xxxxxxxx-xxxx-xxxx-
xxxx-xxxxxxxxxxxx</ac:parameter></ac:structured-macro></p>
      <p><ac:structured-macro ac:name="jira" ac:schema-version="1" ac:macro-
id="15f17da8-a792-48da-82e5-6f471ecf908d"><ac:parameter
ac:name="server">Internal Jira</ac:parameter><ac:parameter
ac:name="columnIds">customfield_11208</ac:parameter><ac:parameter
ac:name="columns">Leiu kirjeldus</ac:parameter><ac:parameter
ac:name="maximumIssues">20</ac:parameter><ac:parameter ac:name="jqlQuery">key
= {} </ac:parameter><ac:parameter ac:name="serverId">xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx</ac:parameter></ac:structured-macro></p>
```

```
      <p><ac:structured-macro ac:name="jira" ac:schema-version="1" ac:macro-
id="83e66ac6-ca5b-4993-96d4-dc06d85e21d6"><ac:parameter
ac:name="server">Internal Jira</ac:parameter><ac:parameter
ac:name="columnIds">customfield_10408</ac:parameter><ac:parameter
ac:name="columns">Risk</ac:parameter><ac:parameter
ac:name="maximumIssues">20</ac:parameter><ac:parameter ac:name="jqlQuery">key
= {} </ac:parameter><ac:parameter ac:name="serverId">xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx</ac:parameter></ac:structured-macro></p>
      <p><ac:structured-macro ac:name="jira" ac:schema-version="1" ac:macro-
id="43adf103-5a0d-4fa1-a01f-7fd00f684f82"><ac:parameter
ac:name="server">Internal Jira</ac:parameter><ac:parameter
ac:name="columnIds">customfield_10381</ac:parameter><ac:parameter
ac:name="columns">Soovitused</ac:parameter><ac:parameter
ac:name="maximumIssues">20</ac:parameter><ac:parameter ac:name="jqlQuery">key
= {} </ac:parameter><ac:parameter ac:name="serverId">xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx</ac:parameter></ac:structured-macro></p></ac:rich-text-
body></ac:structured-macro>
  - section_type: single
    template: >-
      <ac:layout-section ac:type="single"><ac:layout-cell>
      <h2><strong><ac:structured-macro ac:name="anchor" ac:schema-version="1"
ac:macro-id="c68d2e14-7ba8-4138-be66-214d08029d4b"><ac:parameter
ac:name="">Leiud, riskihinnangud, metoodika ja leiu raskusastmete
legend</ac:parameter></ac:structured-macro>Leiud, riskihinnangud, metoodika
ja leiu raskusastmete legend</strong></h2><ac:structured-macro ac:name="info"
ac:schema-version="1" ac:macro-id="aea77ec3-a52a-44ad-a98e-
55a0cfc3c69e"><ac:parameter ac:name="icon">false</ac:parameter><ac:parameter
ac:name="title">Leiu kirjeldus</ac:parameter><ac:rich-text-body>
      <p style="text-align: left;">Leidudena on kirjas testimise aluseks
oleva standardi verifikatsiooninõuete kohta tuvastatud mittevastavused,
korrektsed teostused või muud esile toomist väärivad tähelepanekud.<br />Leiu
kirjeldus sisaldab järgmiseid elemente:</p>
      <ul>
      <li>tiitel</li>
      <li>leiu raskusaste</li>
      <li>turvaleiu kirjeldus</li>
      <li>risk</li>
      <li>soovitused parandamiseks</li></ul></ac:rich-text-
body></ac:structured-macro><ac:structured-macro ac:name="info" ac:schema-
version="1" ac:macro-id="a4c60720-52d8-4626-b465-dc5ca4cd4bf5"><ac:parameter
ac:name="icon">false</ac:parameter><ac:parameter
ac:name="title">Riskihinnangud</ac:parameter><ac:rich-text-body>
      <p style="text-align: left;">Raportis on leidude kriitilisusele antud
hinnang lähtudes peamiselt tehnilisest vaatevinklist ja testijatele teada
olnud asjaoludest.</p>
      <p style="text-align: left;">Raportis antud hinnangud on indikatsioonid
tellijale enda hinnangu loomiseks täpsemat informatsiooni arvestades -
näiteks on turvatestijale mõneti keeruline hinnata ärilise mõju (<em>business
impact</em>) komponente nagu finantsiline kaotus, reputatsiooni kahjud või
juriidikasse kuuluvaid privaatsuspoliitikaga vastuolusid ja nendega seotud
ohte.</p>
```

Testija ei kuulu tarkvara arendusprotsessi, ei tee otsuseid parandusi vajavate või mittevajavate probleemide osas, ei kirjuta ette rakenduse uuenduspoliitikat. Testija edastab vaid hinnangu vastavalt valitud standardi nõuetele. </ac:rich-text-body></ac:structured-macro><ac:structured-macro ac:name="info" ac:schema-version="1" ac:macro-id="b7bde950-da0b-4c6c-9850-5cfc8a87b7ad"><ac:parameter ac:name="icon">false</ac:parameter><ac:parameter ac:name="title">Metoodika</ac:parameter><ac:rich-text-body>
Riskihinnanguid määrates on kasutatud OWASP riski hindamise metoodikat.
https://owasp.org/www-community/OWASP_Risk_Rating_Methodology
OWASP riski hindamise metoodika järgi kujuneb risk kahest peamisest komponendist:

tõenäosus (likelihood) - parameetrid Threat Agent Factors, Vulnerability Factors
mõju (impact) - parameetrid Technical Impact Factors, Business Impact Factors
Mõlemat komponenti hinnatakse selle vektorite ja parameetrite järgi eraldiseisvalt, mõlemad saavad skoori järgi tulemuseks MADAL, KESKMINE või KÕRGE ning selle põhjal kujuneb leiu riskihinnang alloleva tabeli põhjal:
<table class="wrapped relative-table" style="width: 42.059%;"><colgroup><col style="width: 23.8464%;" /><col style="width: 18.7102%;" /><col style="width: 19.3217%;" /><col style="width: 19.0771%;" /><col style="width: 19.0771%;" /></colgroup>
<tbody>
<tr>
<td style="text-align: center;" colspan="2" rowspan="2">
</td>
<td class="highlight-#ffebe6" style="text-align: center;" title="Background colour : Light red 35%" colspan="3" data-highlight-colour="#ffebe6">Tõenäosus (likelihood)</td></tr>
<tr>
<td style="text-align: center;">KÕRGE</td>
<td style="text-align: center;">KESKMINE</td>
<td style="text-align: center;">MADAL</td></tr>
<tr>
<td class="highlight-#ffebe6" style="text-align: center;" title="Background colour : Light red 35%" rowspan="3" data-highlight-colour="#ffebe6">
Mõju
(impact) </td>
<td style="text-align: center;">KÕRGE</td>
<td class="highlight-#6554c0" style="text-align: center;" title="Background colour : Medium purple 100%" data-highlight-colour="#6554c0">KRIITILINE</td>
<td class="highlight-#de350b" style="text-align: center;" title="Background colour : Dark red 85%" data-highlight-colour="#de350b">KÕRGE</td>
<td class="highlight-#ff8b00" style="text-align: center;" title="Background colour : Dark orange 100%" data-highlight-colour="#ff8b00">KESKMINE</td></tr>

```
      <tr>
      <td style="text-align: center;">KESKMINE</td>
      <td class="highlight-#de350b" style="text-align: center;"
title="Background colour : Dark red 85%" data-highlight-
colour="#de350b"><span style="color: rgb(255,255,255);"
title="">KÕRGE</span></td>
      <td class="highlight-#ff8b00" style="text-align: center;"
title="Background colour : Dark orange 100%" data-highlight-
colour="#ff8b00"><span style="color: rgb(0,0,0);"
title="">KESKMINE</span></td>
      <td class="highlight-#ffe380" style="text-align: center;"
title="Background colour : Medium yellow 45%" data-highlight-
colour="#ffe380"><span style="color: rgb(0,0,0);">MADAL</span></td></tr>
      <tr>
      <td style="text-align: center;">MADAL</td>
      <td class="highlight-#ff8b00" style="text-align: center;"
title="Background colour : Dark orange 100%" data-highlight-
colour="#ff8b00"><span style="color: rgb(0,0,0);"
title="">KESKMINE</span></td>
      <td class="highlight-#ffe380" style="text-align: center;"
title="Background colour : Medium yellow 45%" data-highlight-
colour="#ffe380"><span style="color: rgb(0,0,0);" title="">MADAL</span></td>
      <td class="highlight-#0052cc" style="text-align: center;"
title="Background colour : Dark blue 85%" data-highlight-
colour="#0052cc"><span style="color: rgb(255,255,255);"
title="">INFO</span></td></tr></tbody></table>
      <p><br /></p></ac:rich-text-body></ac:structured-macro><ac:structured-
macro ac:name="info" ac:schema-version="1" ac:macro-id="9fe3d5ed-1f49-4551-
bbd6-588f7cbe3a49"><ac:parameter
ac:name="icon">false</ac:parameter><ac:parameter ac:name="title">Leiu
raskusastmete legend</ac:parameter><ac:rich-text-body>
      <p style="text-align: left;">Leidude raskusastmete paremaks visuaalseks
jälgimiseks ning eristamiseks on igale tasemele omistatud värvikood. Sama
värvikood on kasutusel ka kategooriate ja raporti versiooni kokkuvõtte
juures, kus värv määratakse alajaotuses kõrgeima riskiga leiu järgi.</p>
      <p>🟣 Kriitiline - Kriitilise riskiga leid (vastavalt
riskihindamismaatriksile)</p>
      <p>🔴 Kõrge - Kõrge riskiga leid (vastavalt
riskihindamismaatriksile)</p>
      <p>🟠 Keskmine - Keskmise riskiga leid (vastavalt
riskihindamismaatriksile)</p>
      <p>🟡 Madal - Madala riskiga leid (vastavalt
riskihindamismaatriksile)</p>
      <p>🔵 Info - Info riskiga leid (vastavalt riskihindamismaatriksile).
Täiendava kaitsemeetme võimalus, kuid selle puudumine ei ole ainsa ja
eraldiseisva leiuna haavatavus. Teoreetiline (ilma kontseptsioonitõestuseta)
nõrkus.</p>
      <p>🟢 OK - Varasemalt raporteeritud probleem on antud raporti
versiooniks parandatud</p>
      <p>⚪ Teadmata - Kategooria staatusena - testimise jooksul antud
kategooria nõuete hulgast vigu ei leitud või ei ole testitud või ei kuulunud
testimise skoopi. Leiu staatusena - varasemalt raporteeritud probleemi ei
olnud võimalik üle testida, kuna funktsionaalsus ei olnud toimiv või
kättesaadav.</p></ac:rich-text-body></ac:structured-macro>
      <p><br /></p></ac:layout-cell></ac:layout-section></ac:layout>
```

Lisa 3 – Turvatestide raporti näidise osad

RAK Turvatestimise tervikraport infoturbejuhile (08.11.2024)

Created by [redacted], last modified a minute ago

Testimise info

Organization	Nimetus
STAT	Uus rakenduse

Keskkond	Lingid keskkondadele
TEST keskkond	https://test_rakendus.ee

OWASP ASVS versioon	OWASP ASVS tase
4.0.3	2

Testimise ajaperiood

Valmiduse kuupäev	Due
Nov 01, 2024	Nov 08, 2024

Testijad

Firma	Meeskond
RMIT	Eesnimi Perekonnanimi Eesnimi Perekonnanimi Eesnimi Perekonnanimi Eesnimi Perekonnanimi

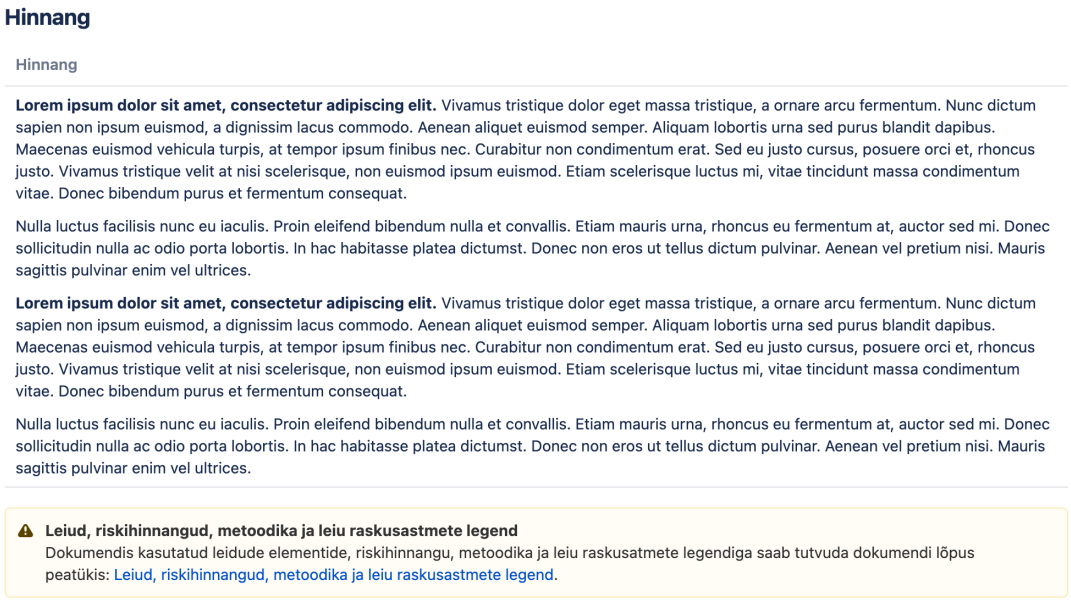
Joonis 5. Kuvatõmmis turvatestide raporti testimise info näidis.

Põhiturvatestimise tulemused

Chapter Name	Leiu raskusaste				
	🔴 Kõrge	🔵 Info	🟠 Keskmine	🟡 Madal	T:
Access Control	1	0	0	1	**2**
Architecture, Design and Threat Modeling	0	0	1	0	**1**
Error Handling and Logging	0	1	0	0	**1**
Validation, Sanitization and Encoding	0	1	0	0	**1**
Total Unique Issues:	**1**	**2**	**1**	**1**	**5**

Showing **4** of **4** statistics.
View in Jira

Joonis 6. Kuvatõmmis põhiturvatestimise tulemuste ülevaatlik tabel.



Hinnang

Hinnang

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tristique dolor eget massa tristique, a ornare arcu fermentum. Nunc dictum sapien non ipsum euismod, a dignissim lacus commodo. Aenean aliquet euismod semper. Aliquam lobortis urna sed purus blandit dapibus. Maecenas euismod vehicula turpis, at tempor ipsum finibus nec. Curabitur non condimentum erat. Sed eu justo cursus, posuere orci et, rhoncus justo. Vivamus tristique velit at nisi scelerisque, non euismod ipsum euismod. Etiam scelerisque luctus mi, vitae tincidunt massa condimentum vitae. Donec bibendum purus et fermentum consequat.

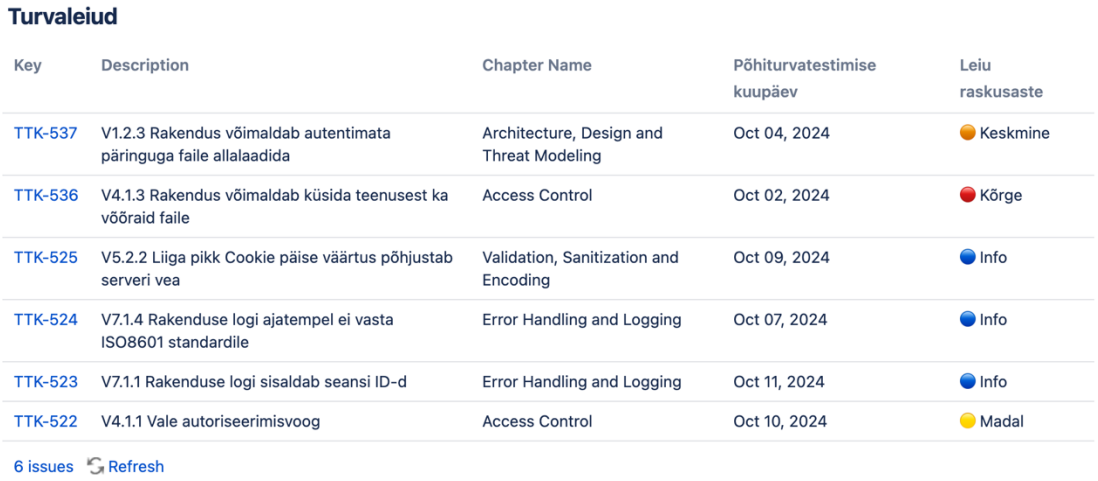
Nulla luctus facilisis nunc eu iaculis. Proin eleifend bibendum nulla et convallis. Etiam mauris urna, rhoncus eu fermentum at, auctor sed mi. Donec sollicitudin nulla ac odio porta lobortis. In hac habitasse platea dictumst. Donec non eros ut tellus dictum pulvinar. Aenean vel pretium nisi. Mauris sagittis pulvinar enim vel ultrices.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tristique dolor eget massa tristique, a ornare arcu fermentum. Nunc dictum sapien non ipsum euismod, a dignissim lacus commodo. Aenean aliquet euismod semper. Aliquam lobortis urna sed purus blandit dapibus. Maecenas euismod vehicula turpis, at tempor ipsum finibus nec. Curabitur non condimentum erat. Sed eu justo cursus, posuere orci et, rhoncus justo. Vivamus tristique velit at nisi scelerisque, non euismod ipsum euismod. Etiam scelerisque luctus mi, vitae tincidunt massa condimentum vitae. Donec bibendum purus et fermentum consequat.

Nulla luctus facilisis nunc eu iaculis. Proin eleifend bibendum nulla et convallis. Etiam mauris urna, rhoncus eu fermentum at, auctor sed mi. Donec sollicitudin nulla ac odio porta lobortis. In hac habitasse platea dictumst. Donec non eros ut tellus dictum pulvinar. Aenean vel pretium nisi. Mauris sagittis pulvinar enim vel ultrices.

⚠ **Leiud, riskihinnangud, metoodika ja leiu raskusastmete legend**
Dokumendis kasutatud leidude elementide, riskihinnangu, metoodika ja leiu raskusatmete legendiga saab tutvuda dokumendi lõpus peatükis: Leiud, riskihinnangud, metoodika ja leiu raskusastmete legend.

Joonis 7. Kuvatõmmis turvatestide hinnang ja legendile viide.



Turvaleiud

Key	Description	Chapter Name	Põhiturvatestimise kuupäev	Leiu raskusaste
TTK-537	V1.2.3 Rakendus võimaldab autentimata päringuga faile allalaadida	Architecture, Design and Threat Modeling	Oct 04, 2024	Keskmine
TTK-536	V4.1.3 Rakendus võimaldab küsida teenusest ka võõraid faile	Access Control	Oct 02, 2024	Kõrge
TTK-525	V5.2.2 Liiga pikk Cookie päise väärtus põhjustab serveri vea	Validation, Sanitization and Encoding	Oct 09, 2024	Info
TTK-524	V7.1.4 Rakenduse logi ajatempel ei vasta ISO8601 standardile	Error Handling and Logging	Oct 07, 2024	Info
TTK-523	V7.1.1 Rakenduse logi sisaldab seansi ID-d	Error Handling and Logging	Oct 11, 2024	Info
TTK-522	V4.1.1 Vale autoriseerimisvoog	Access Control	Oct 10, 2024	Madal

6 issues Refresh

Joonis 8. Kuvatõmmis turvaleidude ülevaatlik tabel.

V4 Access Control

Summary

(L) - V4.1.1 Vale autoriseerimisvoog

Leiu klassifikaator	Leiu klassifikaatori kirjeldus
V4.1.1	Verify that the application enforces access control rules on a trusted service layer, especially if client-side access control is present and could be bypassed.

Põhiturvatestimise kuupäev	Leiu raskusaste
Oct 10, 2024	🟡 Madal

Leiu kirjeldus

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tristique dolor eget massa tristique, a ornare arcu fermentum. Nunc dictum sapien non ipsum euismod, a dignissim lacus commodo. Aenean aliquet euismod semper. Aliquam lobortis urna sed purus blandit dapibus. Maecenas euismod vehicula turpis, at tempor ipsum finibus nec. Curabitur non condimentum erat. Sed eu justo cursus, posuere orci et, rhoncus justo. Vivamus tristique velit at nisi scelerisque, non euismod ipsum euismod. Etiam scelerisque luctus mi, vitae tincidunt massa condimentum vitae. Donec bibendum purus et fermentum consequat.

Risk

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tristique dolor eget massa tristique, a ornare arcu fermentum. Nunc dictum sapien non ipsum euismod, a dignissim lacus commodo. Aenean aliquet euismod semper. Aliquam lobortis urna sed purus blandit dapibus. **Maecenas euismod vehicula turpis**, at tempor ipsum finibus nec. Curabitur non condimentum erat. Sed eu justo cursus, posuere orci et, rhoncus justo. Vivamus tristique velit at nisi scelerisque, non euismod ipsum euismod. Etiam scelerisque luctus mi, vitae tincidunt massa condimentum vitae. Donec bibendum purus et fermentum consequat.

Soovitused

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tristique dolor eget massa tristique, a ornare arcu fermentum. Nunc dictum sapien non ipsum euismod, a dignissim lacus commodo. Aenean aliquet euismod semper. Aliquam lobortis urna sed purus blandit dapibus. Maecenas euismod vehicula turpis, at tempor ipsum finibus nec. Curabitur non condimentum erat. Sed eu justo cursus, posuere orci et, rhoncus justo. Vivamus tristique velit at nisi scelerisque, non euismod ipsum euismod. Etiam scelerisque luctus mi, vitae tincidunt massa condimentum vitae. Donec bibendum purus et fermentum consequat.

Joonis 9. Kuvatõmmis turvaleiuga ASVS peatükist.