**Handleiding**

Project SecMon

Opdrachtgever: Arnim Eijkhoudt, Hogeschool van Amsterdam

Project: Project SecMon

Originele Auteurs: Dave van Ast, Stanley Numan, Jasper Groot

Verbeterd door: Christiaan Druif

Datum: 30 juni 2014

Versie: 0.3

Status: Final

# Versiebeheer

|  |  |  |  |
| --- | --- | --- | --- |
| Versie | Datum | Omschrijving | Auteur |
| 0.1 | 28-06-2013 | Final versie | Dave van Ast, Stanley Numan en Jasper Groot |
| 0.2 | 27-06-2014 | Concept van vernieuwde versie | Christiaan Druif |
| 0.3 | 30-06-2014 | Final versie | Christiaan Druif |
| 0.4 | 07-03-2015 | Webpanel updates, Final versie | Niels Hilgesom |

Table of Contents

[Versiebeheer 2](#_Toc413774343)

[1. Inleiding 4](#_Toc413774344)

[2. Overzicht applicatie 5](#_Toc413774345)

[3. Beschrijving modules 6](#_Toc413774346)

[4. Configuratie van de applicatie 7](#_Toc413774347)

[4.1. Mappenstructuur van de applicatie 7](#_Toc413774348)

[4.2. Algemene applicatieconfiguratie 7](#_Toc413774349)

[4.3. Configuratie definities 8](#_Toc413774350)

[4.3.1. Aanmaken nieuwe definitie 8](#_Toc413774351)

[4.4. Configuratie rule bestanden 9](#_Toc413774352)

[4.4.1. Aanmaken van een rule 10](#_Toc413774353)

[4.5. Configuratie rule bestanden 11](#_Toc413774354)

[4.5.1. Aanmaken actiescripts 11](#_Toc413774355)

[5. Webpagina 12](#_Toc413774356)

[5.1. Mappenstructuur van het webpanel 12](#_Toc413774357)

[5.2. Starten en stoppen van het webpanel 12](#_Toc413774358)

[5.3 Webpanel gebruiken 13](#_Toc413774359)

[5.4 Beschrijving modules 19](#_Toc413774360)

[6. Applicatie starten, stoppen, herstarten en status opvragen. 20](#_Toc413774361)

# Inleiding

Dit document dient als handleiding voor de applicatie Security Monitor (SecMon). In dit document wordt de opbouw van de applicatie beschreven. Daarnaast worden de verschillende modules van de applicatie beschreven en wordt beschreven hoe de applicatie geconfigureerd en beheerd kan worden.

De applicatie SecMon is een product wat is voortgekomen uit het project Security Monitor, onderdeel van de minor Forensic Intelligence and Security aan de Hogeschool van Amsterdam.

Het project is voor het eerst in 2013 gemaakt door Dave van Ast, Stanley Numan, Jasper Groot, Sujen Kandasamy en Vinesh Koenjbiharie. In 2014 is het project door Christiaan Druif aangepast en verbeterd.

Het doel van het project was het opleveren van een applicatie die de VMWare omgeving van de Hogeschool van Amsterdam kan beschermen tegen cyber-aanvallen van zowel binnen uit als buitenaf. De applicatie moet in staat zijn verschillende aanvallen te herkennen en daarop volgend een actie uit te voeren. Dit moet gerealiseerd worden door middel van het parsen van de firewall log(s). Daarna is de applicatie uitgebreid waardoor het nu ook mogelijk is om andere logfiles, zoals Apache logfiles te parsen.

# Overzicht applicatie

Onderstaand een overzicht van alle onderdelen van de applicatie.

* **secmon** *(root directory)*
  + **Core** *(core componenten directory)*
    - Config.txt
    - Configuration.py
    - Daemon.py
    - FileManager.py
    - Main.py
    - Monitor.py
    - securitymonitor.py
    - Trigger.py
    - Rules.py
    - MatchManager.py
    - SearchManager.py
  + **Custom** *(gebruikers componenten directory)*
    - **Actions**
      * Email.py
    - **Rules**
      * RuleDefinitionTable.txt
      * <naam rule>.txt

De structuur van de applicatie is als volgt:

/secmon/core

/secmon/custom

/secmon/custom/rule

/secmon/custom/actions

# Beschrijving modules

In de core componenten directory ‘secmon/core/’ bevinden de volgende modules:

* Configuration.py
  + Deze module leest Config.txt, waar de configuratie in staat. Andere bestanden maken hier gebruik van. Bijvoorbeeld om de mail configuratie in te stellen.
* Daemon.py
  + Deze module zorgt ervoor dat de applicatie daadwerkelijk gestart, gestopt en herstart kan worden als een Disk And Execution MONitor. Deze module wordt door securitymonitor.py aangesproken door wijze van argumenten.
* FileManager.py
  + Deze module zorgt ervoor dat de rules worden uitgelezen door Rules.py aan te spreken en leest tevens de logfiles uit.
* Main.py
  + Deze module zorgt ervoor dat de monitor wordt aangeroepen en start daarmee in principe het monitoren van de applicatie.
* Monitor.py
  + Deze module is de kern van de applicatie en is het combinatiestuk met alle andere modules.
* Rules.py
  + Deze module zorgt ervoor dat de verschillende bestanden in de rules directory worden uitgelezen en zet deze in een dictionary. Tevens wordt de RuleDefinitionTable.txt uitgelezen en in een dictionary gezet.
* Securitymonitor.py
  + Deze module zorgt ervoor dat de applicatie als Daemon kan worden gestart door middel van argumenten. De volgende vier mogelijkheden kunnen worden gebruikt: start, stop, status of restart.
* MatchManager.py
  + Deze module zorgt ervoor dat de matches die in de rule zijn opgegeven in een aparte lijst worden gezet. De matches zijn de sleutelwoorden waarop er in de log gezocht gaat worden. Deze matchlist wordt in deze module opgebouwd en gecombineerd met de dictionary van de RuleDefinitionTable.txt
* SearchManager.py
  + Deze module ontvangt de matchlist van de MatchManager.py. Aan de hand van deze matchlist wordt er een regex opgebouwd waarmee de opgegeven logfile wordt geparsed op zoek naar matches. Tevens worden hier de counts bijgehouden en indien nodig wordt er aangegeven of er een actie uitgevoerd moet worden. Dit wordt doorgegeven aan de Trigger.py
* Trigger.py
  + De module zorgt ervoor dat er een actie wordt uitgevoerd. De actie die wordt uitgevoerd wordt uit de rule dictionary gehaald.

# Configuratie van de applicatie

## Mappenstructuur van de applicatie

Onderstaand de onderdelen van de applicatie benoemd met een korte uitleg.

**/secmon/**

Dit is de hoofdmap van de applicatie, waarin de submappen /core en /custom staan.

**/secmon/core**

Dit is de map waar alle kritieke onderdelen van de applicatie staan.

**/secmon/custom**

Dit is de map waar de actions en rules staan.

**/secmon/custom/rules**

Dit is de map waar de rules staan. Elk bestand in deze map wordt uitgelezen, niet elk bestand kan gebruikt worden. Het bestand moet wel plain-tekst zijn.

**/secmon/custom/actions**

In deze map staan de actiebestanden, standaard staat alleen de Email.py actie in deze map.

## Algemene applicatieconfiguratie

Nadat de applicatie op de server is geplaatst zijn er een aantal punten die aangepast dienen te worden aan de specifieke locatie van applicatie en de modules. De bestanden die aangepast moeten worden zijn:

* Config.txt

In de Config.txt dienen de juiste waardes en paden opgegeven te worden achter het ‘=’ teken. De standaard opmaak ziet er als volgt uit.

// directories of the Files

logDir = [pad naar de firewall log]

ruleDir = [pad naar de rulebestanden]

actionDir = [pad naar actiebestanden]

interval = 30

fromaddr = sendfrom@example.com

toaddr = sendto@example.com

username = [smtpserver username]

password = [password]

server = smtp.server.com:[poort]

sleeptimer = 0.01

Deze waardes kunnen aangepast worden naar eigen configuratie en credentials. Om deze file aan te passen is er een optionele mogelijkheid om gebruik te maken van een webGUI. zie hoofdstuk Webpagina.

## Configuratie definities

RuleDefinitionTable.txt is een configuratiebestand die optioneel is om te wijzigen. Dit bestand zal alleen worden gewijzigd in een situatie waar een nieuwe definitie aangemaakt moet worden. Dit bestand wordt gebruikt als een mapping van sleutelwoorden die te vinden zijn in logs en die gebruikt kunnen worden in de rule bestanden.

De standaard inhoud van het bestand is als volgt:

PROTOCOL = 'PROTO='

SOURCEIP = 'SRC='

SOURCEPT = 'SPT='

TARGETIP = 'DST='

TARGETPT = 'DPT='

Er kunnen op elk moment definities toegevoegd worden.

Het woord *voor* de ‘=’ teken is een definitie die gebruikt kan worden in de rule bestanden voor het definiëren van rules. Het woord/waarde wat *ná* de ‘=’ komt is een sleutelwoord die te vinden is in de firewalllog en deze moet overeenkomen met wat er daadwerkelijk in een firewalllog voorkomt. Anders werkt de mapping niet.

### Aanmaken nieuwe definitie

Dit bestand is dus alleen maar nodig als er extra informatie nodig is voor een match. Bijvoorbeeld: In een logfile van iptables willen we als er boven de 400 UDP pakketten van een bepaald IP-adres afkomstig zijn, een actie ondernemen. In de logfile van iptables wordt het protocol als PROTO=UDP aangegeven. In de rule file is PROTOCOL = UDP opgegeven bij protocol. In de RuleDefinitionsTable.txt staat PROTOCOL = ‘PROTO=’. Na combinatie wordt er gezocht op PROTO=UDP, exact zoals het in de logfile staat.

Een nieuwe definitie wordt automatisch uitgelezen en er wordt automatisch een match gevonden bij de rule. Wel moet dezelfde keyword gebruikt worden. Dus PROTOCOL in de rule, moet ook PROTOCOL in de RuleDefinitionsTable.txt zijn.

## Configuratie rule bestanden

In de directory /var/secmon/custom/rule/ kunnen rulebestanden worden aangemaakt en gewijzigd. In een rulebestand worden sleutelwoorden met waarden aangemaakt. De meeste sleutelwoorden zijn woorden die ‘gemapped’ zijn aan sleutelwoorden die logs teruggevonden kunnen worden, zoals het sleutelwoord: TARGETIP. Deze is gemapped aan het sleutelwoord ‘DST=’ binnen de firewalllog. Sleutelwoorden zoals: NAME, DESCRIPTION, COUNT en ACTION zijn geen mappings die gekoppeld zijn aan de firewalllog, maar zijn wel essentieel voor de applicatie.

Hieronder een aantal voorbeeld sleutelwoorden. De sleutelwoorden met een \* zijn verplicht.

NAME = ‘’

\*DESCRIPTION = ‘’

SOURCEIP =

SOURCEPT =

TARGETIP =

TARGETPT =

PROTOCOL =

MESSAGE =

\*COUNT [<,<=,>,>=,=]

INTERVAL = hh:mm:ss

\*ACTION = ‘’

\*MATCH =

\*LOG =

|  |  |  |
| --- | --- | --- |
| Sleutelwoord | Operator | Uitleg |
| Name | = | Hier kan binnen de ‘’ een naam opgegeven worden van de rule. |
| Description | = | Hier kan binnen de ‘’ een beschrijving opgegeven worden van de rule. |
| SOURCEIP | = | Hier kan een source IP adres opgegeven. |
| SOURCEPT | = | Hier kan een source poort opgegeven. |
| TARGETIP | = | Hier kan een target IP adres opgegeven. |
| TARGETPT | = | Hier kan een target poort opgegeven. |
| PROTOCOL | = | Hier kan een protocol, zoals: UDP of TCP opgegeven worden. |
| COUNT | [<,<=,>,>=,=] | Hier kan een packet teller opgegeven worden. |
| INTERVAL | hh:mm:ss | Hier kan een tijdbereik opgegeven worden. |
| ACTION | = | Hier kan één of meerdere action script opgegeven worden |
| LOG | = | De logfile waar de rule op doelt |
| MESSAGE | = | Een zin waar de logfile naar kan kijken, zoals een bepaalde error melding in een Apache error log |
| MATCH | = | Een lijst van sleutelwoorden, waar in de rules op gezocht wordt. Dus bijvoorbeeld SOURCEIP, TARGETIP, PROTOCOL |

### Aanmaken van een rule

Rules kunnen worden aangemaakt in plain-text formaat onder de locatie /var/secmon/custom/rules/.

Een nieuwe rule hoeft niet aan speciale eisen te voldoen, alleen dat het een plain-tekst bestand is. Zoals bovenstaand aangegeven zijn de volgende sleutelwoorden verplicht:

**DESCRIPTION**

Verplicht omdat deze in Email.py wordt gebruikt om aan te geven welke rule het betreft wat getriggerd is.

**COUNT**

Verplicht omdat de applicatie moet weten bij welke hoeveelheid matches er getriggerd moet worden

**ACTION**

Verplicht omdat de applicatie moet weten welke actie(s) ondernomen moeten worden bij een trigger.

**MATCH**

Wordt gebruikt voor de matchlist. Hierin staan vermeld op welke sleutelwoorden er in een logbestand gezocht moeten worden. Zonder deze sleutelwoord wordt er niets gematcht.

**LOG**

Geeft aan welke log gebruikt moet worden.

Onderstaand een voorbeeld van hoe een rule eruit kan zien:

NAME = 'DoS TCP flood attack'

LOG = log.txt

DESCRIPTION = 'Rule check for: (TCP flood)-TCP packets with a threshold of 400 or more that are targeted at a specific target IP.'

SOURCEIP = 145.92.6.10

TARGETIP = 192.168.2.128

PROTOCOL = TCP

COUNT > 20

ACTION = 'Email.py'

MATCH = SOURCEIP, PROTOCOL, TARGETIP

## Configuratie rule bestanden

Onderstaand een beschrijving van hoe een actie script aangemaakt kan worden.

### Aanmaken actiescripts

Actiescripts kunnen naar eigensmaak worden gemaakt bijvoorbeeld in Python. In Trigger.py worden argumenten meegegeven aan actiescripts die staan opgegeven in een een rulebestand. In de standaardlevering van de applicatie is een email notificatie script meegeleverd. Deze is te vinden onder /var/secmon/custom/actions. Binnen deze directory kunnen meerdere actiescripts worden aangemaakt, vergeet niet om deze op te geven in een rulebestand anders werkt de actiescript niet. Om de actiescript op te geven in een rulebestand moet er achter het sleutelwoord ACTION = en binnen de single quotes de naam + extensie van de actiescript worden opgegeven. Bijvoorbeeld: ACTION = ‘Email.py’ of ACTION = ‘Email.py, test.test’. Meerdere opgegeven actiescripts moeten worden gescheiden met een komma.

# Webpagina

## Mappenstructuur van het webpanel

Onderstaand de onderdelen van de applicatie benoemd met een korte uitleg.

**/assets/**

Deze map bevat client-side statische code die clients nodig hebben voor het opbouwen van het webpanel, denk hierbij aan javascript en CSS bestanden. Deze folder is publiekelijk toegankelijk bij het starten van het webpanel.

**/lib/**

Lib staat voor library. In deze folder staan verschillende serverside python scripts die de applicatie gebruikt.

**/log/**

Deze folder bevat één logbestand van het webpanel met informatie over welk IP-adres, welke pagina op welk tijdstip heeft geopend. Deze folder zal automatisch worden aangemaakt zodra het webpanel voor de eerste keer zal worden gestart.

**/pid/**

In deze folder worden een aantal bestanden geplaatst zodra het webpanel zal worden gestart. Deze bestanden worden automatisch verwijderd zodra de webserver weer zal worden stopt. Deze folder zal automatisch worden aangemaakt bij het starten van het webpanel.

**/view/**

Deze folder bevat statische HTML code gemixt met dynamische Python code. In deze directory zijn alle pagina’s van het webpanel terug te vinden. Deze folder is niet publiekelijk toegankelijk, de webserver bouwt a.d.h.v. deze bestanden volledig statische HTML pagina’s op die, als de gebruiker de juiste rechten heeft, op aanvraag worden geserveerd aan de clients.

## Starten en stoppen van het webpanel

De webserver zal zichzelf automatisch configureren bij elke start van het webpanel. Starten van het webpanel gaat als volgt:

* Cd /secmon/web/
* Python2.7 webpanel.py start

Stoppen van het webpanel gaat als volgt:

* Cd /secmon/web/
* Python2.7 webpanel.py stop

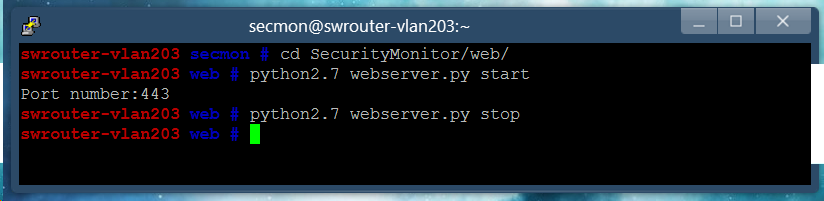
Let op: het starten en stoppen van het webpanel heeft geen effect op de security monitor daemon. Het webpanel is een zelfstandige daemon die eventueel kan worden gebruikt het beheer van rule bestanden te vereenvoudigen. De security monitor kan echter ook enkel via de CLI worden gebruikt.

Let op: Het webpanel gaat er standaard vanuit dat de security monitor applicatie zich één niveau lager bevind in de directory structuur.

Let op: niet alle python library scripts zijn geschikt voor python 3. Starten met het command “python webpanel.py start” kan resulteren in foutmeldingen of een ‘hangende’ socket. Mocht dit toch gebeuren dan zal het PID nummer van het python proces moeten worden gezocht (b.v. met het netstat commando) en worden gestopt. Vervolgens is het weer mogelijk om het webpanel te starten.

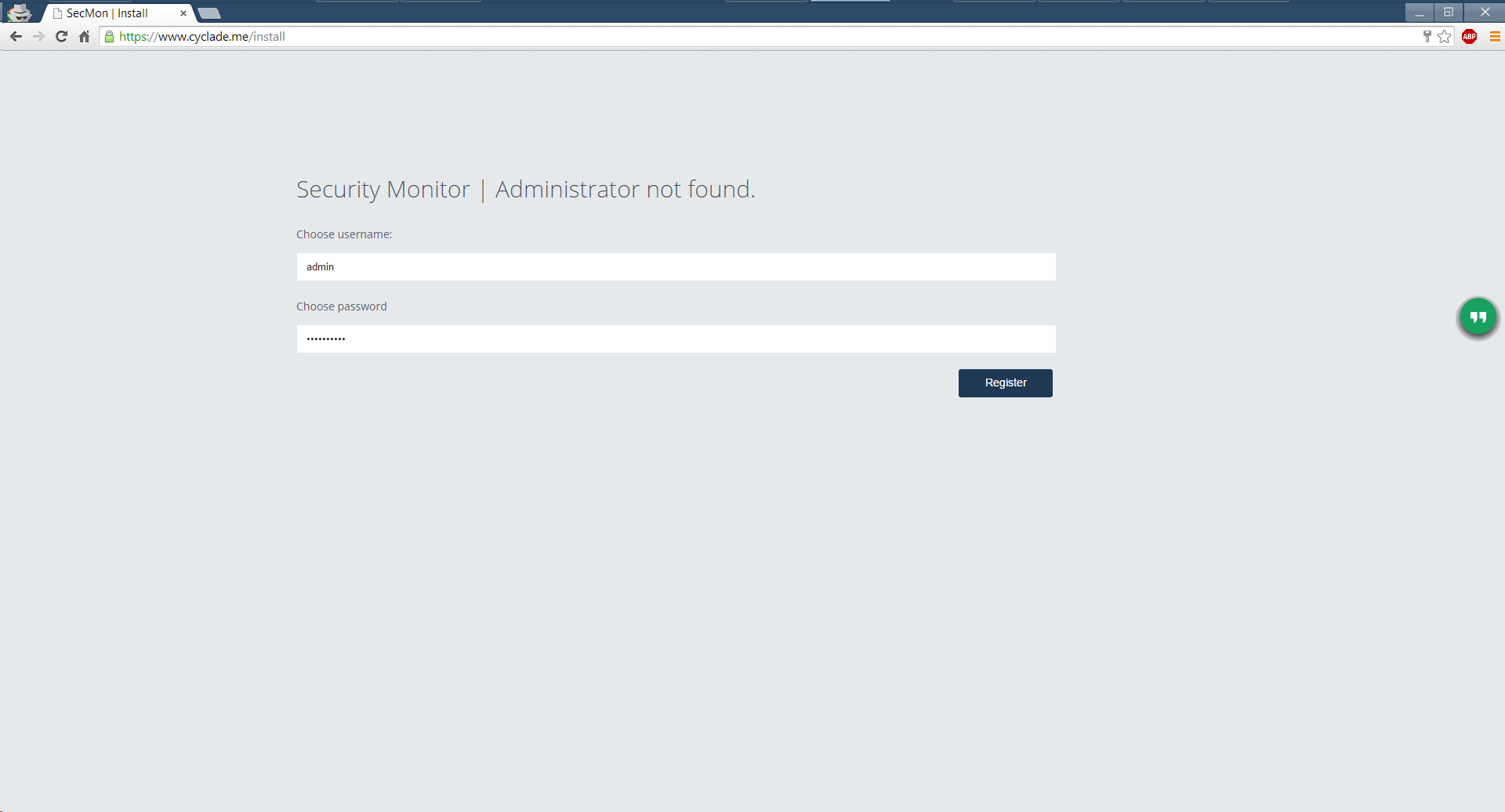
## 5.3 Webpanel gebruiken

Bij de eerste start van het webpanel in de CLI zal de gebruiker een paar vragen krijgen. Zo zal er worden gevraagd om een poort nummer en mogelijk een aantal vragen voor het genereren van een SSL certificaat. Deze gegevens zijn nodig om het webpanel correct te laten functioneren. Als er al een SSL certificaat aanwezig en/of als er al een configuratie bestand met een poortnummer is, dan zullen deze vragen niet worden gesteld.



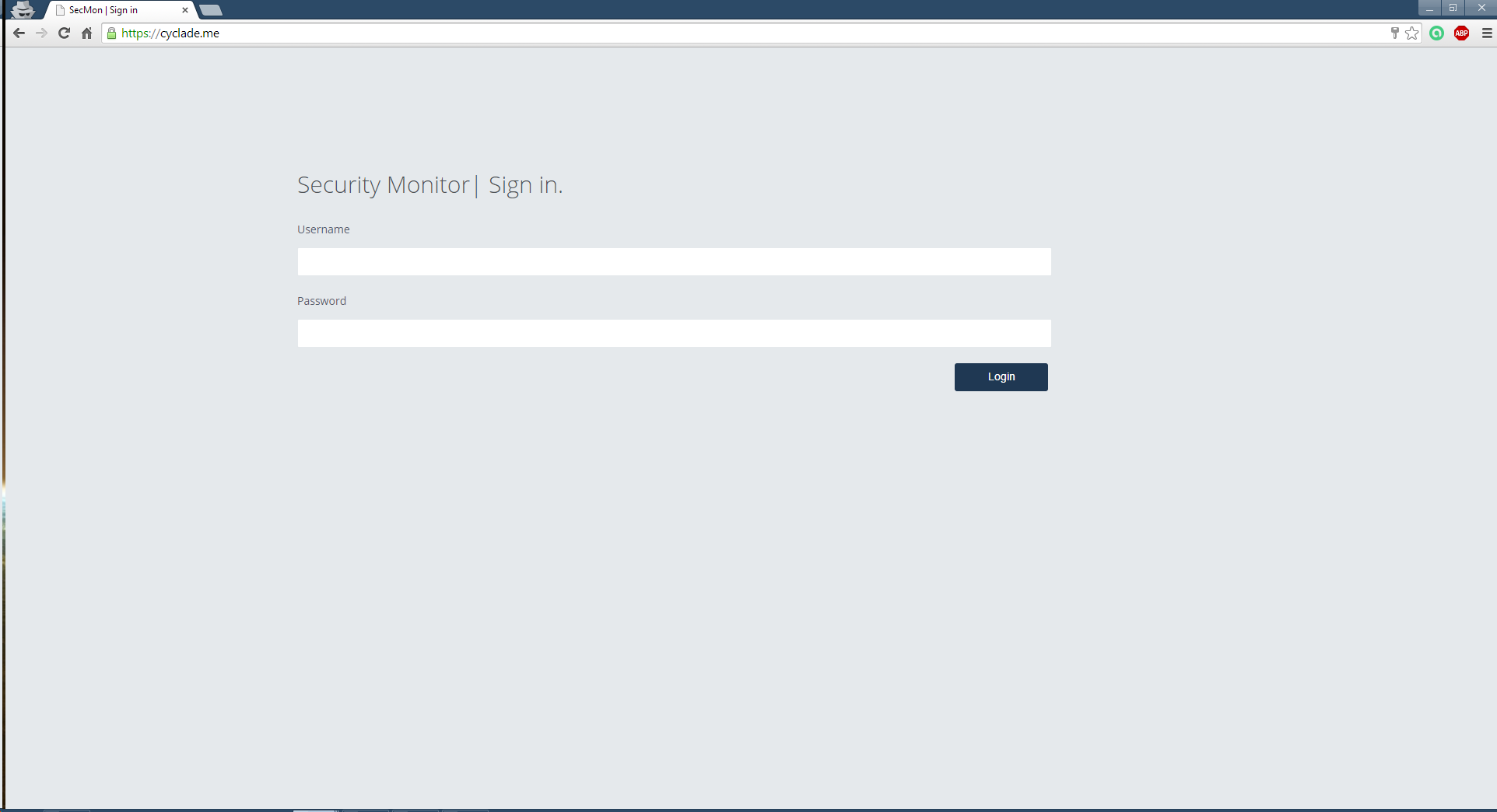
*Afbeelding 5.3.1: het starten en stoppen van het webpanel.*

Na het accepteren van het SSL certificaat zal er voor de eerste keer een administrator account moeten worden aangemaakt in de browser. Het is nu mogelijk om het webpanel te benaderen in de browser.



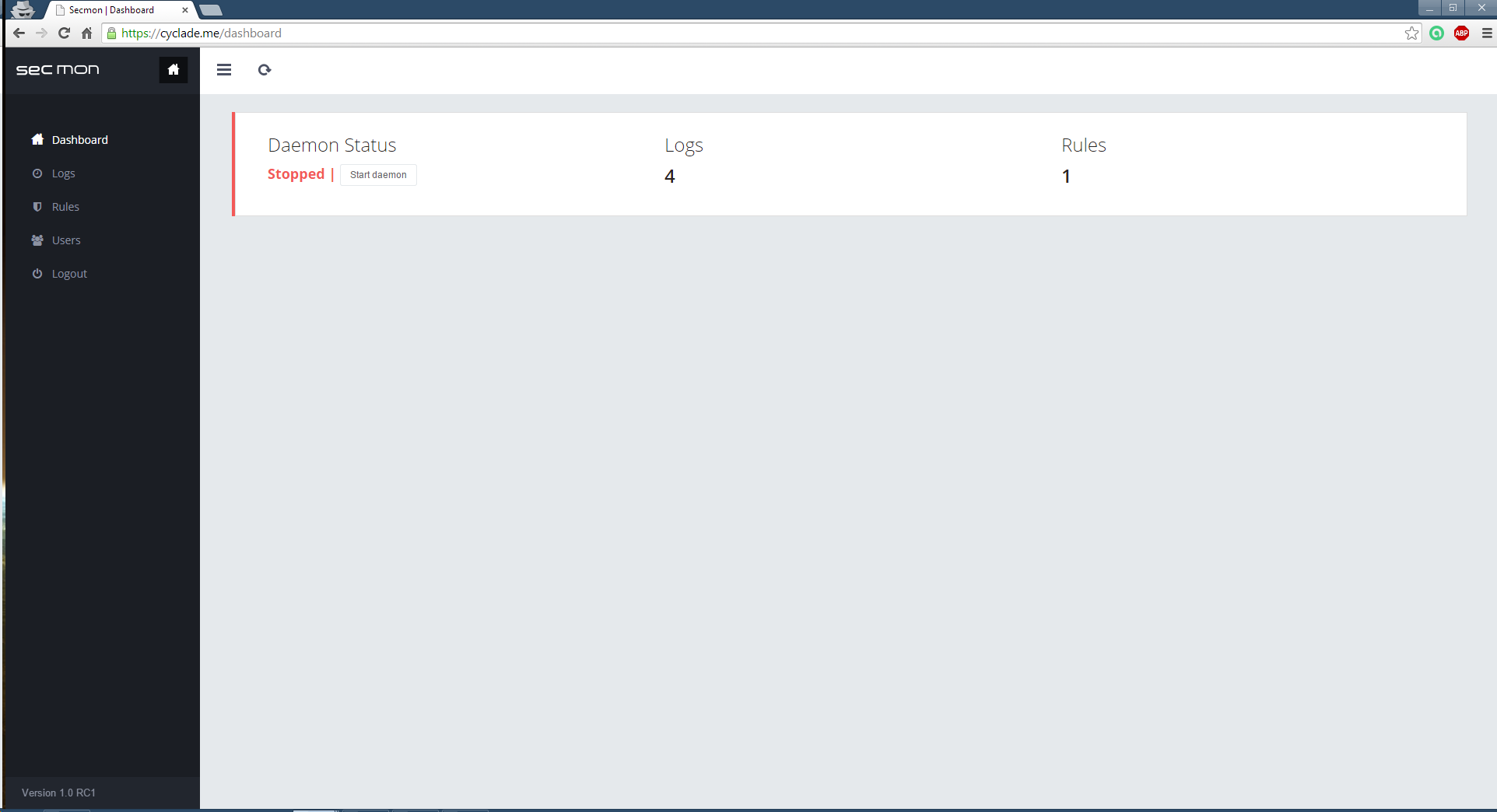
*Afbeelding 5.3.2: er zal een administrator account moeten worden aangemaakt.*

Na het aanmaken van de eerste administrator account bestaat de mogelijkheid om in te loggen. Het webpanel zal in de /lib/ folder een auth.db bestand aanmaken met usernames en gehashte wachtwoorden.



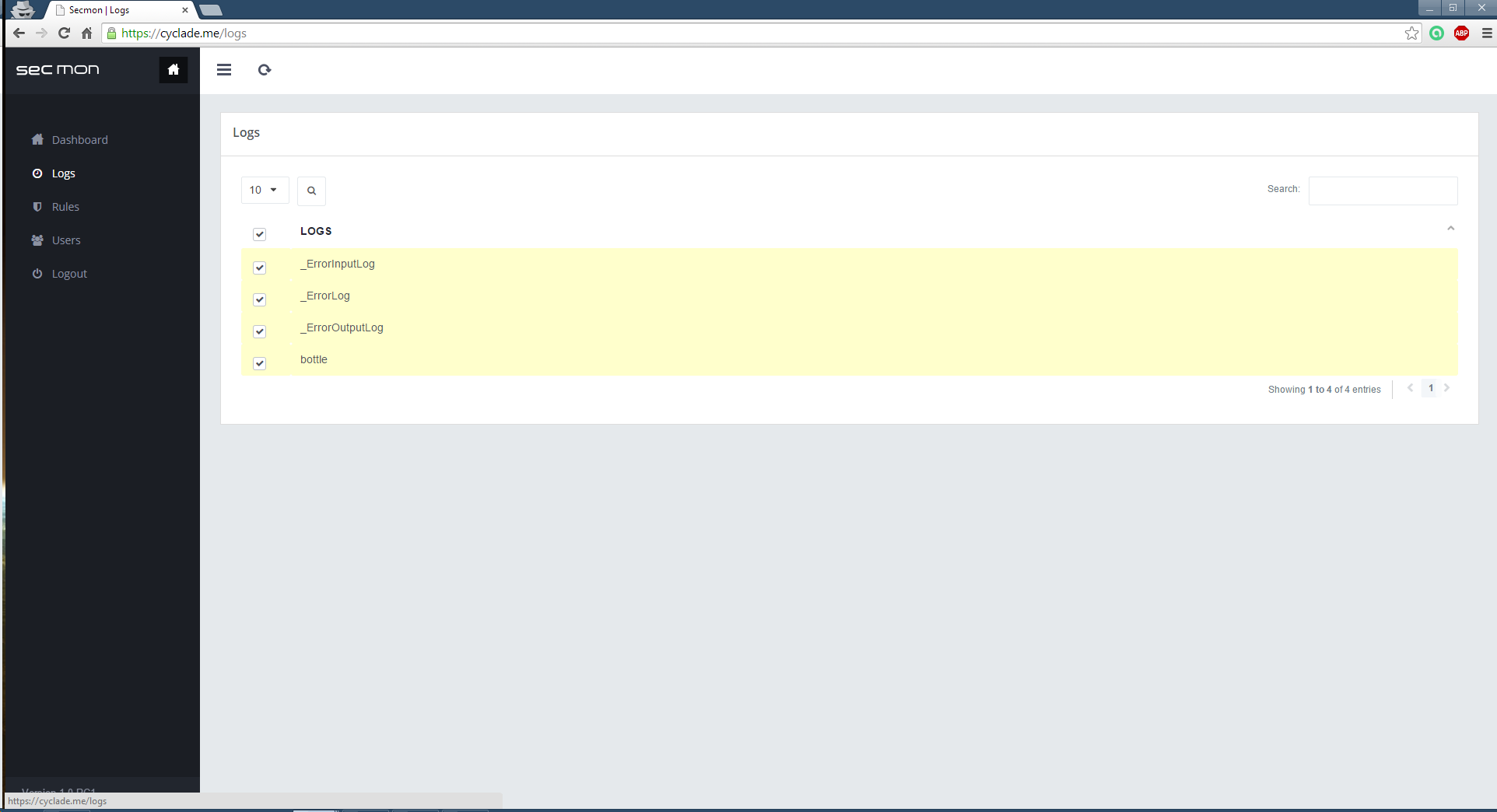
*Afbeelding 5.3.3: na het aanmaken van de administrator account kan er worden ingelogd.*

Na een correcte log-in zal het dashboard worden weergegeven.



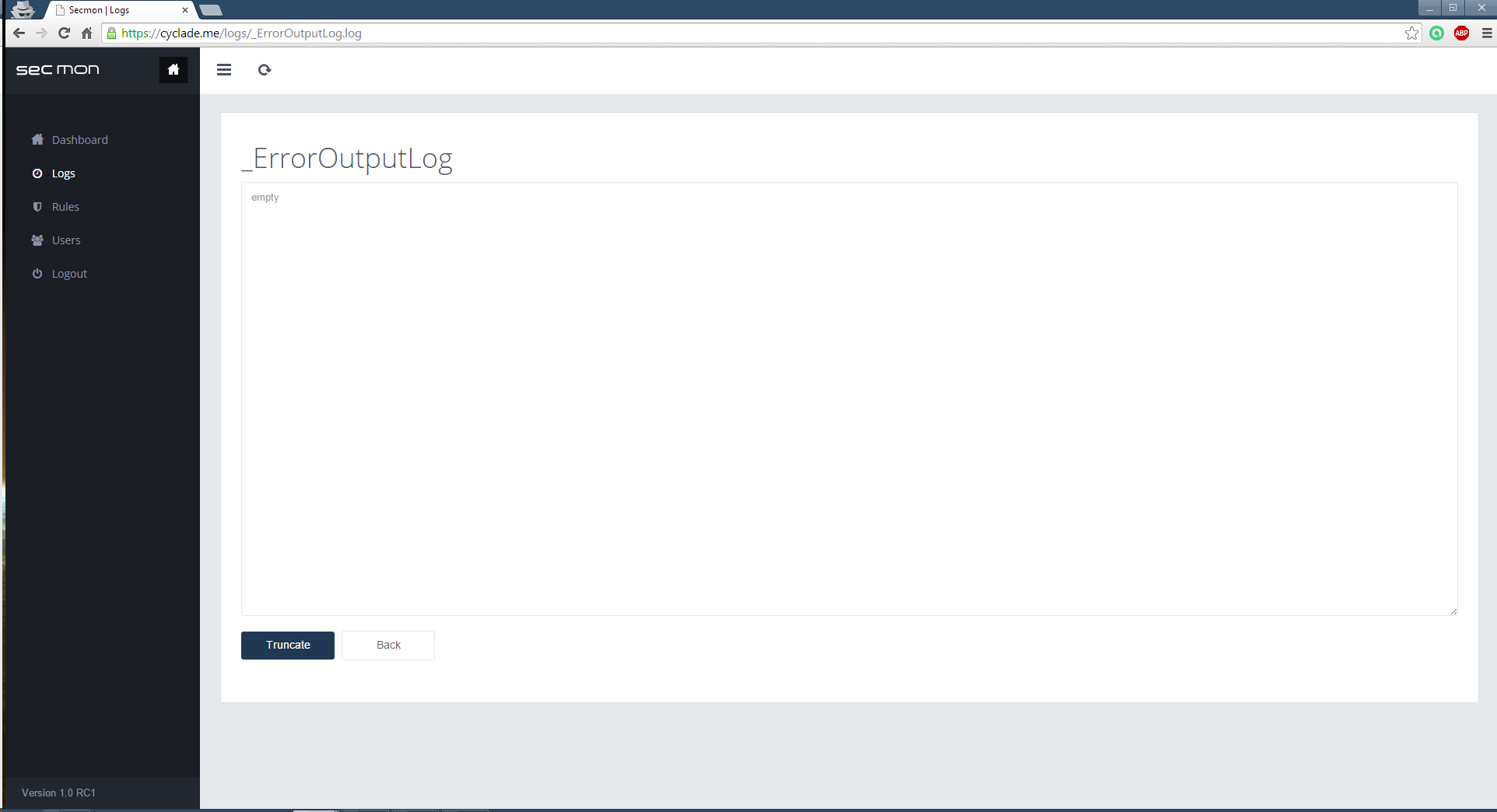
*Afbeelding 5.3.4: het dashboard geeft een overzicht over het aantal logs, rules en bied de mogelijkheid om de daemon te starten of te stoppen.*

Aan de linkerkant worden verschillende menu opties weergegeven.



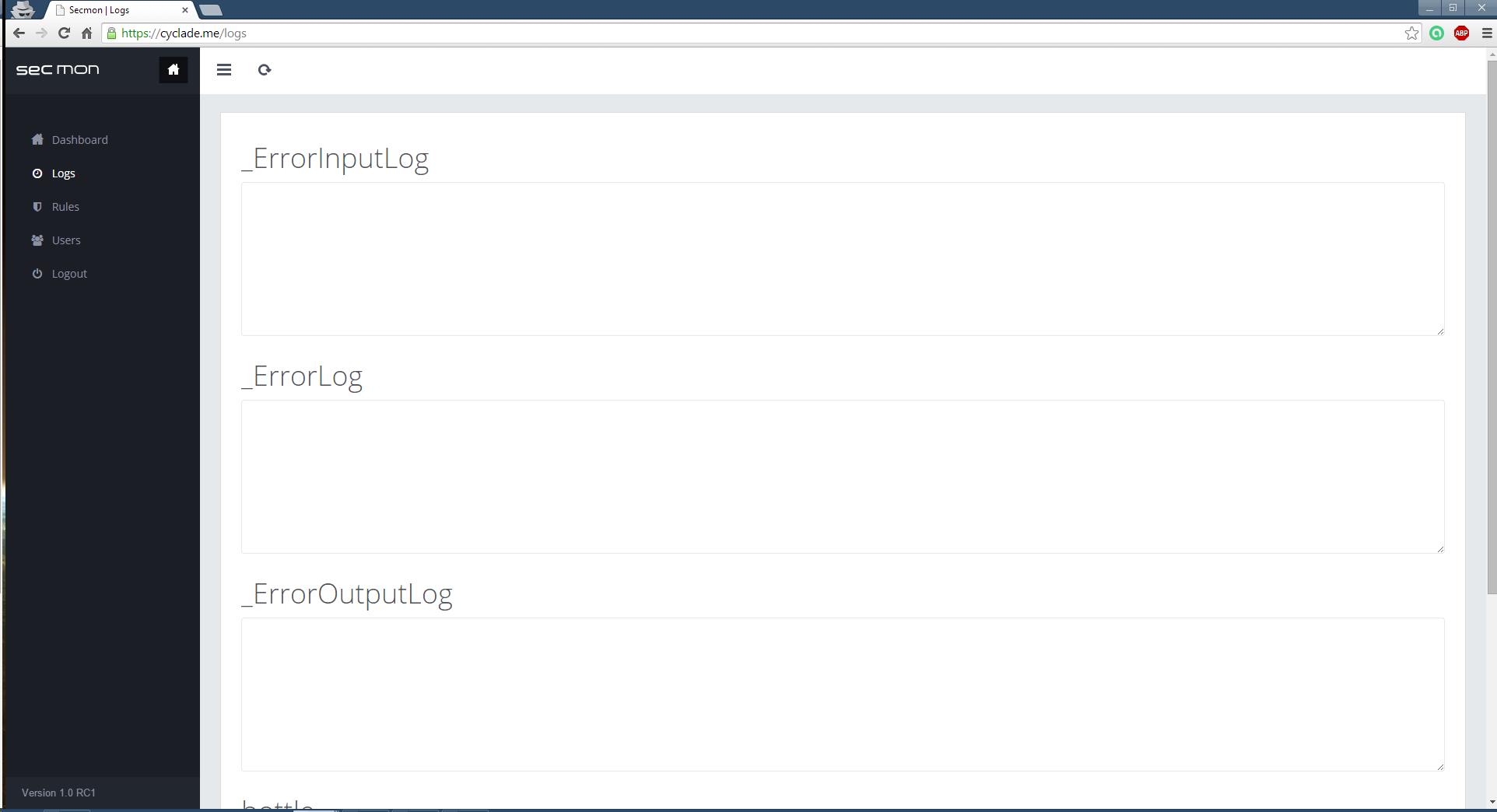
*Afbeelding 5.3.5: logs pagina.*

Op de logs pagina worden een aantal log bestanden van de security monitor weergegeven. De ingelogde gebruiker kan hier op een logbestand in de lijst klikken om de loginhoud te zien. Ook kunnen er meerdere logbestanden worden geselecteerd en worden bekeken.

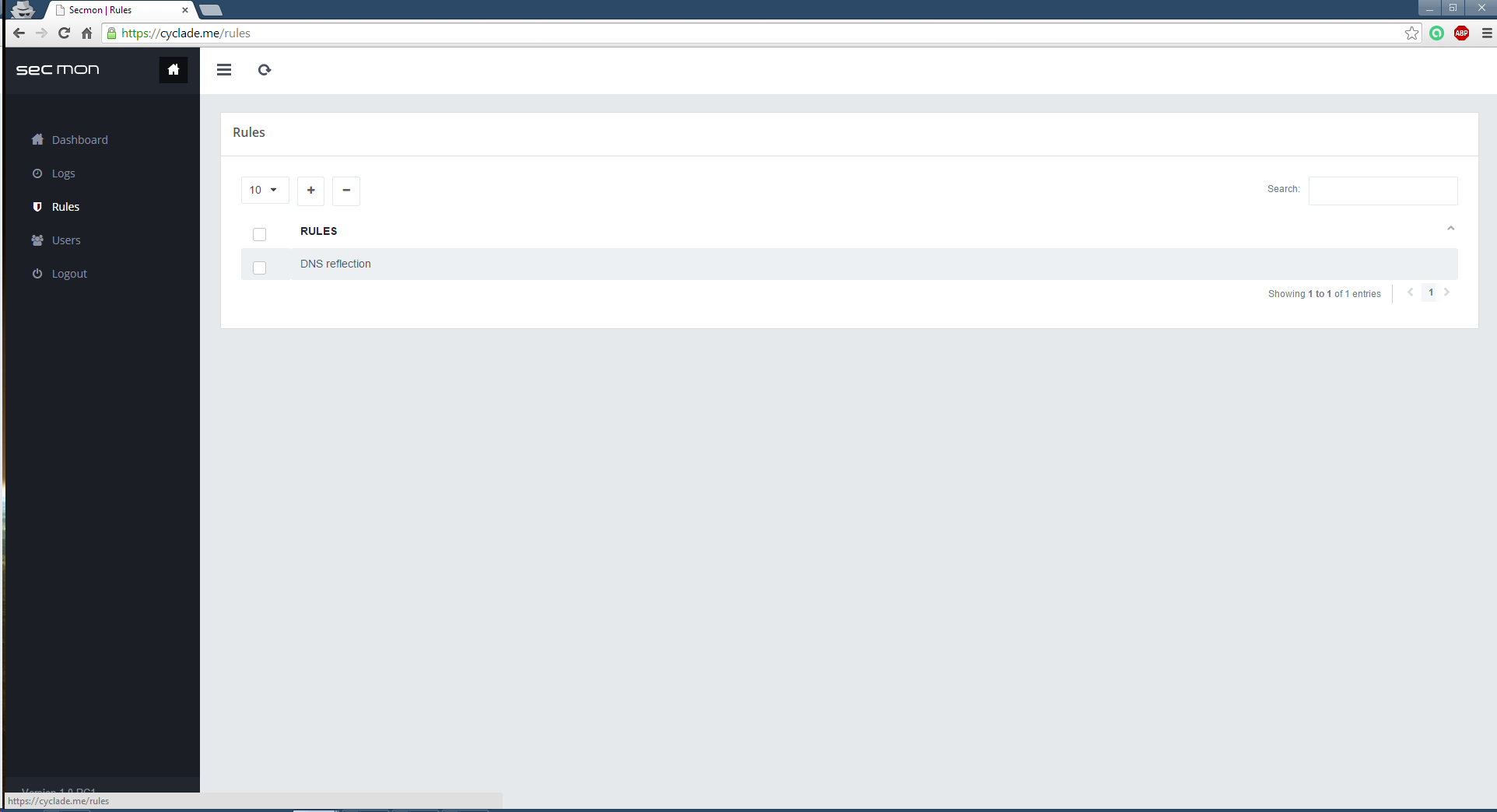


*Afbeelding 5.3.6: weergave van één enkel logbestand.*

Bij het bekijken van één enkel logbestand zal ook een “truncate” knop zichtbaar zijn waarmee het logbestand kan worden gewist.



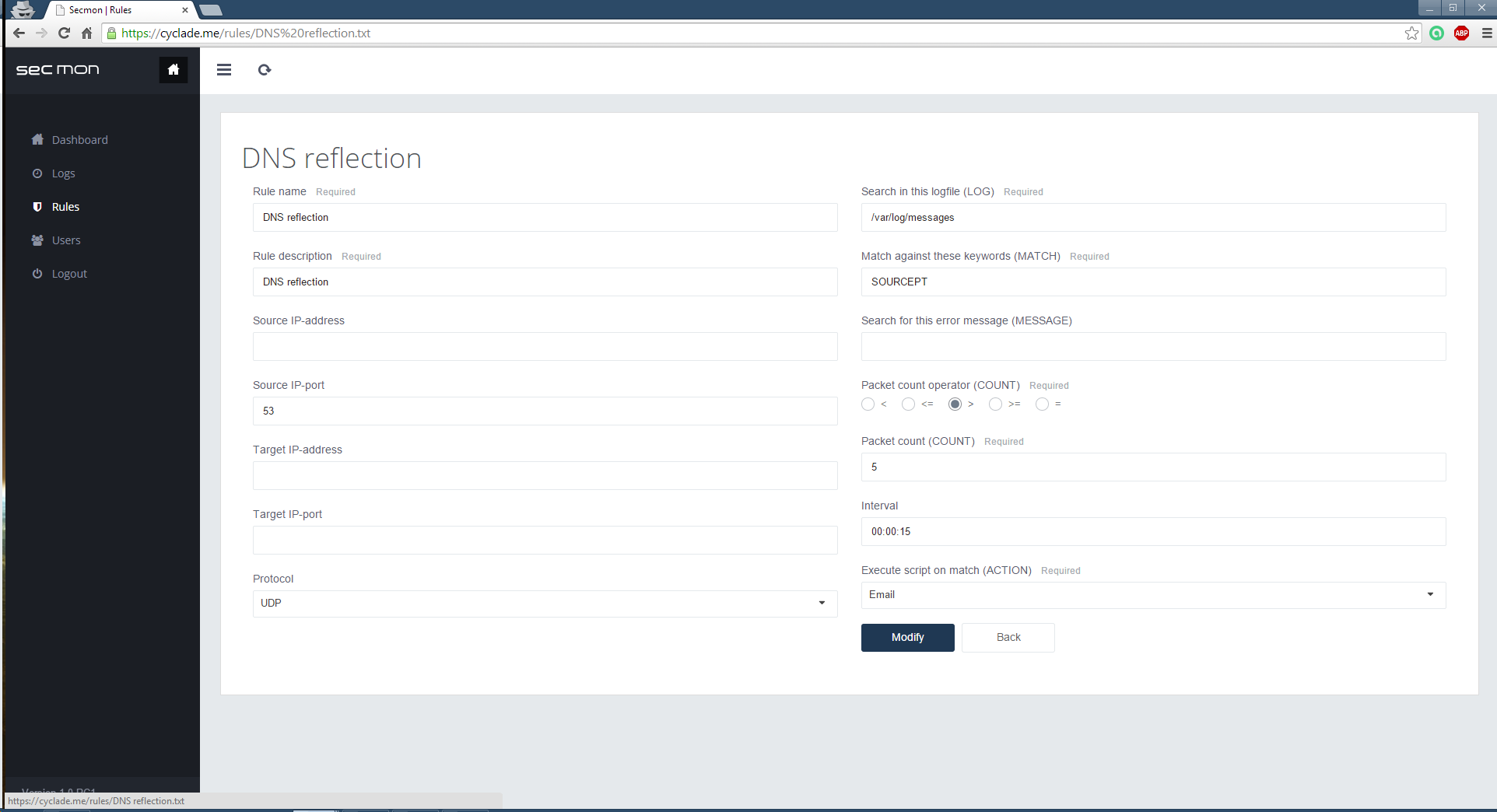
*Afbeelding 5.3.7: meerdere logbestanden kunnen in één overzicht worden weergeven.*



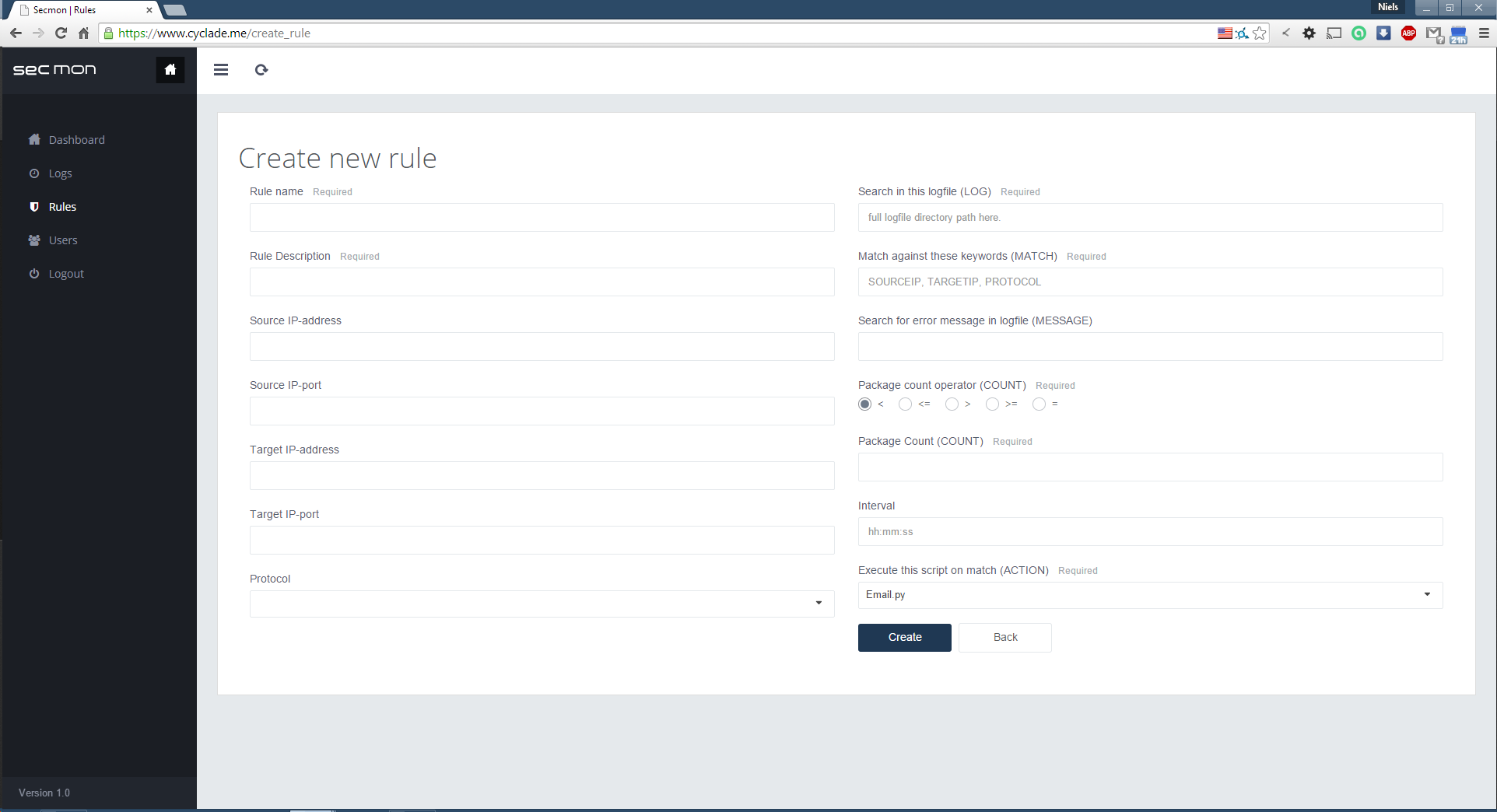
*Afbeelding 5.3.8: rules pagina.*

In het rules scherm worden alle ingestelde rules weergegeven. Er kan d.m.v. een zoekfunctie worden gezocht naar verschillende rules en bij een groot aantal rules worden er pagina’s aangemaakt en kan worden gekozen hoeveel rules er per pagina worden weergegeven. Een bestaande rule (definitie) aanpassen kan door op de betreffende rule te klikken in de lijst.

Uiteraard kunnen er in het overzicht ook nieuwe rules worden aangemaakt met de “+” knop of worden verwijderd door een rule te selecteren en te klikken op de “-” knop.

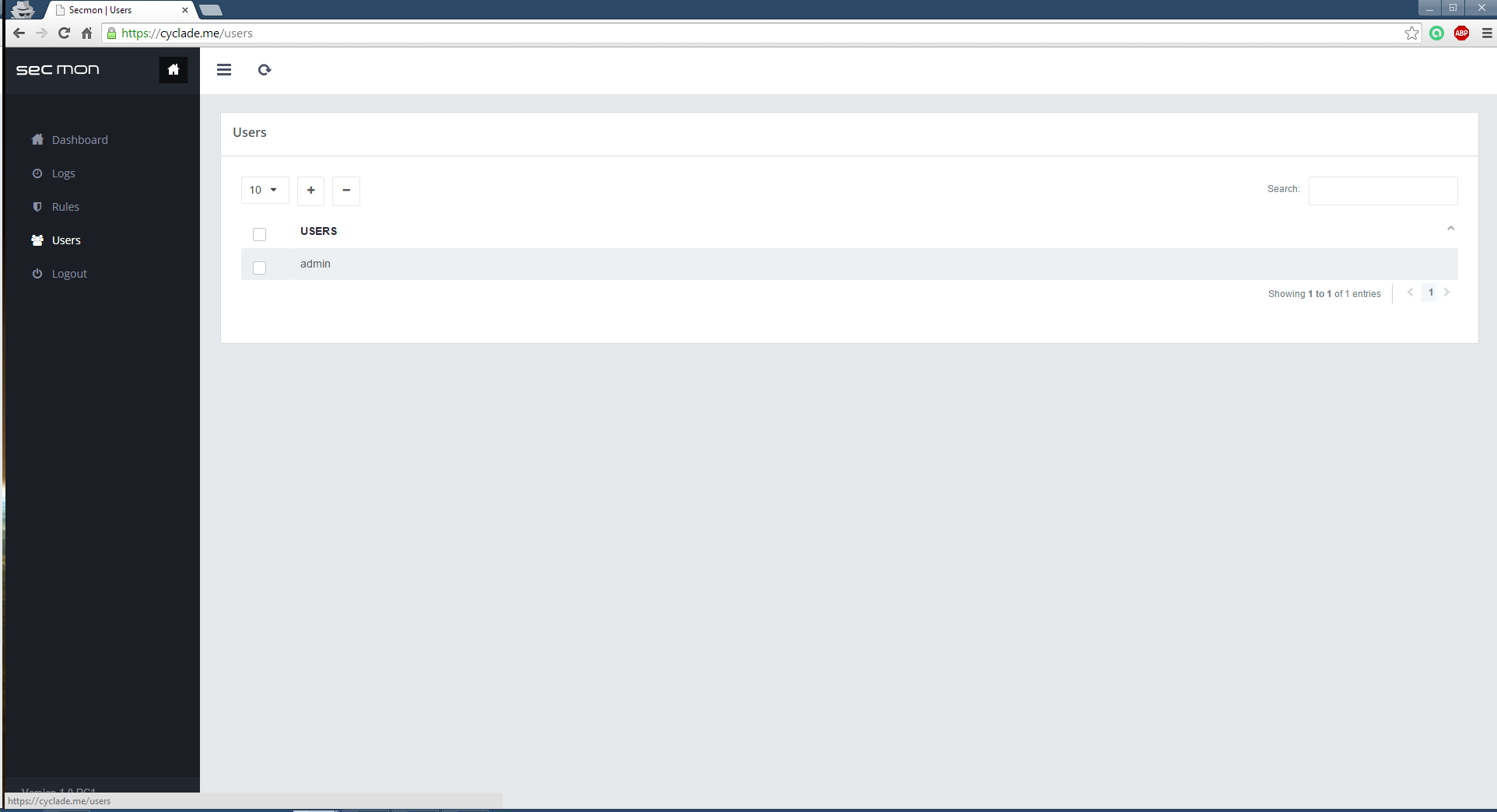


*Afbeelding 5.3.9: een bestaande rule aanpassen.*

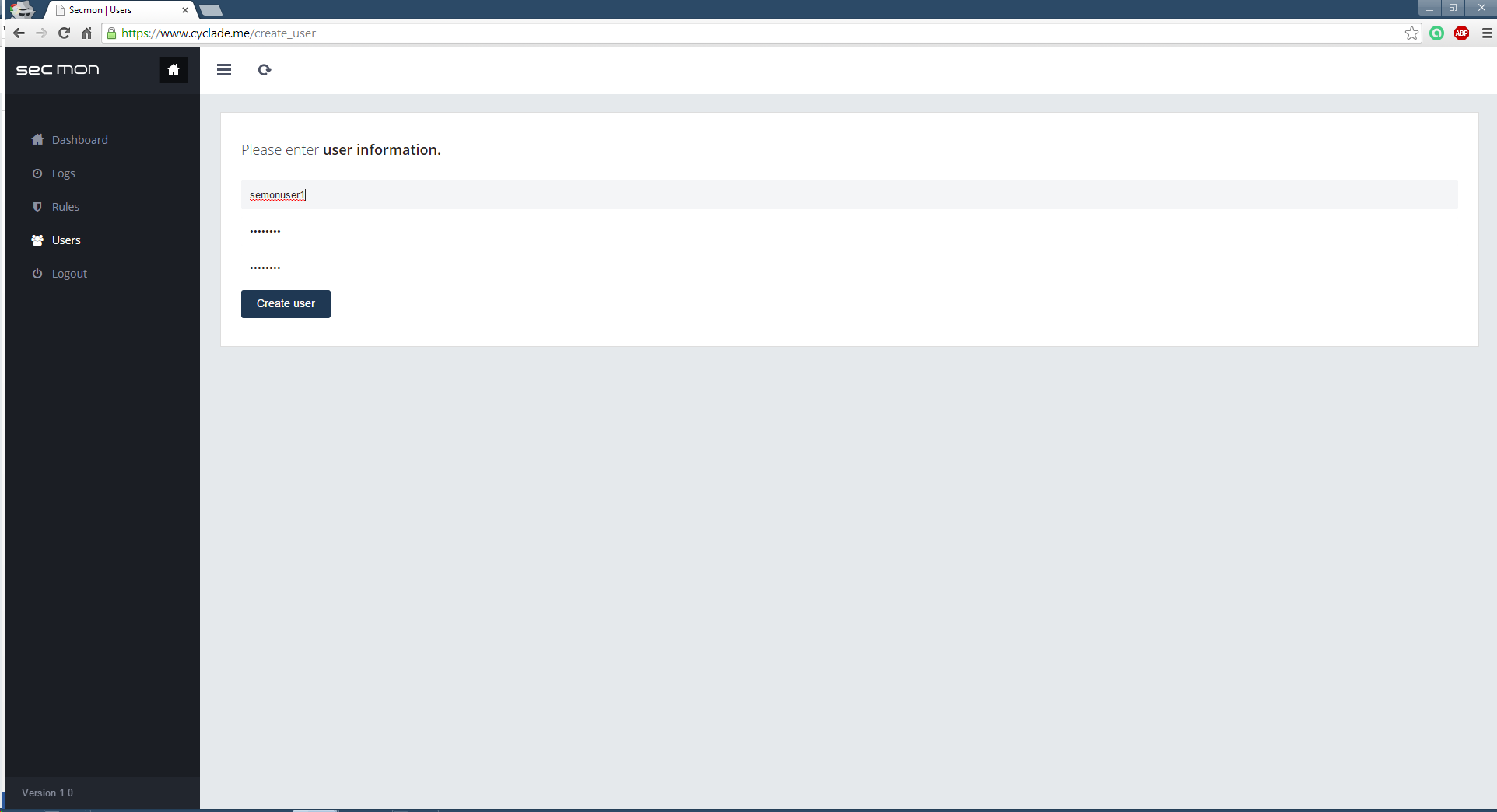


*Afbeelding 5.3.10: een nieuwe rule aanmaken.*

Tot slot bestaat er ook nog een mogelijkheid om meerdere gebruikers te registreren en gebruikers te verwijderen.



*Afbeelding 5.3.11: alle gebruikers weergeven.*



*Afbeelding 5.3.12: een nieuwe gebruiker aanmaken.*

Het is raadzaam om na elke sessie te klikken op de “logout” tekst omdat deze de huidige sessie wist zowel in de database als de cookies in de browser.

## 5.4 Beschrijving modules

Het webpanel leunt sterk op twee frameworks. Voor de serverside code is gebruik gemaakt van het [python bottle framework](http://bottlepy.org/docs/0.12/) versie [0.12.8](https://pypi.python.org/pypi/bottle). Dit framework zorgt er voor dat de server kan reageren op client requests en door gebruik te maken van de documentatie kunnen er vrij snel web applicaties worden ontwikkeld.

Het uiterlijk van het webpanel is gebasseerd op het [webarch framework](http://themeforest.net/item/webarch-responsive-admin-dashboard-template/6157416) versie 2.7.1. Na het kopen van een licentie kan de nieuwste versie worden gedownload. Dit framework bestaat enkel uit clientside code zoals javascript en css bestanden.

Tijdens de ontwikkeling van het security monitor webpanel zijn beide frameworks aan elkaar gekoppeld echter blijven beide frameworks nog heel simpel te updaten. Door een nieuwe versie van het “bottle.py” bestand te overschrijven in de /lib/ folder kan het python framework worden geupdate. Alle webarch code is terug te vinden in de /assets/ folder en ook deze kan worden overschreven om de client-side code van een update te voorzien.

* webserver.py
  + Dit is de kern van het security monitor webpanel. In dit bestand staat omschreven welke pagina’s er zijn, welke pagina publiek toegankelijk is en welke afgesloten, welke “view” bij welke pagina moet worden geladen en hoe de server moet reageren op verschillende post requests uit HTML forms in de applicatie. Het overgrote deel van het gedrag van de applicatie is terug te vinden in dit script en ontwikkeld volgens de bottle documentatie standaard.
* lib/bottledaemon.py
  + Dit script zorgt er voor dat de bottle server als een daemon achtergrond proces op de server kan functioneren. Daarnaast zorgt dit script voor een “python SSL socket” en zorgt het er voor dat het webserver.py bestand argumenten kan worden meegegeven zoals start en stop.
* lib/lockfile
  + Dit is een standaard library file die nodig is voor de bottledaemon.py script.
* lib/daemon
  + Dit is een standaard library file die nodig is voor de bottledaemon.py script.
* lib/\_\_init\_\_.py
  + Dit is een standaard bestand dat er voor zorgt dat python bestanden in deze directory kan lezen.
* lib/bottle.py
  + Dit is een standaard library file die nodig is voor de bottledaemon.py en het webserver.py script.

# 6. Applicatie starten, stoppen, herstarten en status opvragen.

Het uitvoeren van het onderstaande vereist minimaal zonder enige toepassingen/wijzigingen dat de gebruiker in de directory zit waar het bestand ‘securitymonitor.py’ zich ook bevindt. Dit wordt gerealiseerd door de cd command, die hieronder staat opgegeven.

Het starten van de applicatie kan door middel van het onderstaande worden uitgevoerd:

* + - cd /var/secmon/core
    - securitymonitor.py start

Het stoppen van de applicatie wordt via de volgende manier uitgevoerd:

* + - cd /var/secmon/core
    - securitymonitor.py stop

Het herstarten van de applicatie wordt via de volgende manier uitgevoerd:

* + - cd /var/secmon/core
    - securitymonitor.py restart

De status opvragen van de applicatie wordt op de volgende manier uitgevoerd:

* + - cd /var/secmon/core
    - securitymonitor.py status

Het starten van het webpanel wordt via de volgende manier uitgevoerd:

* + - cd /var/secmon/web
    - python2.7 webserver.py start

Het stoppen van het webpanel wordt via de volgende manier uitgevoerd:

* + - cd /var/secmon/web
    - python2.7 securitymonitor.py stop