

---

# **Galactic Inspector Documentation**

***Release 2***

**Wolnosciewicz Team**

**Jun 08, 2019**



---

## Contents:

---

<b>1</b>	<b>Quick start</b>	<b>3</b>
1.1	Installing with PIP and running natively . . . . .	3
1.2	Running with docker . . . . .	3
1.3	Running with docker-compose . . . . .	4
<b>2</b>	<b>Configuration reference</b>	<b>5</b>
<b>3</b>	<b>Proxy / Hidden monitoring service</b>	<b>7</b>
3.1	Conception . . . . .	7
<b>4</b>	<b>From authors</b>	<b>9</b>



Tiny application for centralized monitoring of remote servers. In comparison to traditional health checks, the *galactic-inspector* is executing commands using **SSH**.

**Functionality:**

- SOCKS proxy support: Possibility to hide service in the internet using TOR
- Health checks: Execute remote command, check exit code. Execute other command on failure to repair simple things
- Authenticity check: Check if remote filesystem is untouched by third-party (eg. by hosting provider, by other hosting users, by the government)
- Detailed Slack/Mattermost notifications (also can be configured to work with SOCKS proxy) on each event

**Main conception**

The main conception is to monitor files integrity like some of IDS systems are doing. What is different from other IDS systems is a native slack/mattermost support, focus on hiding the monitoring service behind a proxy, and the size - it's tiny.

**High anonymity**

To protect the infrastructure against eg. government censorship in politically active projects, the IPS can be hidden behind a SOCKS proxy eg. TOR network



# CHAPTER 1

---

## Quick start

---

Application is written in Python 3, the required Python version is at least 3.6. If you would decide to use a docker image, then you do not need to worry about the technologies and dependencies.

### Most important notes:

- “/root/.ssh” and “/root/.ssh-server-audit/expectations” needs to be a volume eg. named volume, as the expectations are generated once, also the SSH keys need to be persisted
- You need to pass your configuration file to the container, and place it under “/usr/local/etc/ssh-server-audit”

## 1.1 Installing with PIP and running natively

```
pip install galactic-inspector
```

## 1.2 Running with docker

```
docker run \
  -p 80:80 \
  -v "./containers/ssh-server-audit:/usr/local/etc/ssh-server-audit" \
  -v "expectations:/root/.ssh-server-audit/expectations" \
  -v "openssh:/root/.ssh" \
  --entrypoint="ssh-server-audit --port=80 --sleep-time=500 --expectations-directory=/
  ↪root/.ssh-server-audit/expectations"
  wolnosciewicz/ssh-server-audit
```

## 1.3 Running with docker-compose

```
version: '2'
volumes:
  expectations:
  openssh:
services:
  app_auditor:
    image: wolnosciowiec/ssh-server-audit:latest
    volumes:
      # here you attach your configuration file as a volume
      - "./containers/ssh-server-audit:/usr/local/etc/ssh-server-audit"
      - "expectations:/root/.ssh-server-audit/expectations"
      - "openssh:/root/.ssh"
    expose:
      - 80
    ports:
      - "80:80"
    environment:
      # gateway configuration (see RiotKit's Harbor, nginx-letsencrypt-
      ↪ companion, nginx proxy-gen)
      - VIRTUAL_HOST=audit.localhost
      - VIRTUAL_PORT=80
      #depends_on:
      #  - tor
    entrypoint: "ssh-server-audit --port=80 --sleep-time=500 --expectations-
    ↪ directory=/root/.ssh-server-audit/expectations"
```



## CHAPTER 2

---

### Configuration reference

---

```
# create as much services as you need and name them as you want
test_vagrant_volume:

    # optional SOCKS proxy
    socks_host: ""
    socks_port: 9150

    # SSH host, port, user
    host: "localhost"
    port: 2422
    user: root
    password: "root"
    auth_method: password

    # SSH options
    ssh_tcp_timeout: 300
    ssh_banner_timeout: 120
    ssh_auth_timeout: 300
    verify_ssh_fingerprint: true # use "false" ONLY for testing, never use "false" on ↵
    ↵production

    # public key, and your passphrase to the public key
    public_key: ""
    passphrase: ""

    # mattermost/slack
    notifications:
        type: "none" # mattermost, slack, none
        url: "http://some-url-here" # (slack/mattermost only)
        resend_after: 300 # (slack/mattermost only)
        connection_timeout: 300 # (slack/mattermost only)
        #proxy: "socks://some-socks-server:9050" # (slack/mattermost only)
        proxy_retry_num: 3 # (slack/mattermost only)
        # when prixy will fail all retries, then should we skip using proxy to send a ↵
    ↵notification? (slack/mattermost only)
```

(continues on next page)

(continued from previous page)

```

    proxy_fallback_on_failure: false

    # host files integrity checking mechanism
    checksum_method: "sha256sum"

    # a command to execute when checksum failed, can be empty, or can be a some_
    ↪ resume command
    on_security_violation: "echo 'Checksum failed... unmounting secure data disk,
    ↪ waiting for administrator intervention...'"

    # files to keep eye on integrity
    checksum_files:
        sh: '/bin/sh'
        bash: '/bin/bash'
        losetup: '$(whereis losetup|awk "{print \$2}")'

    # checks to perform on the host to validate additionally if everything is ok
    healthchecks:
        - command: "ps aux |grep SOOOOMETHING"
          on_failure: "echo 'Something on failure'"

        # Set to false if you do not want to execute commands from "on_failure"
    ↪ when checksum security violation was detected
        on_failure_even_if_security_violation: false

        - command: "ps aux |grep ON_VIOLATION_WILL_EXECUTE"
          on_failure: "echo 'Something on failure - on_failure_even_if_security_
    ↪ violation: true'"
          on_failure_even_if_security_violation: true

        - command: "ps aux |grep bash"
          on_failure: "echo 'This should not show'"
          on_failure_even_if_security_violation: false

```

---

### Proxy / Hidden monitoring service

---

Hiding the service in the internet is a one of the coolest features that *Galactic Inspector* offers in comparison to other monitoring software. The feature was implemented with considering the anarchist organizations needs - a hidden, secure monitoring service that no any government can censor.

#### 3.1 Conception

*Galactic Inspector* can be installed anywhere, on some VPS or dedicated server, hidden somewhere on the NAT network on a Raspberry Pi, or in other place. The SOCKS proxy can be used to make connections from *Galactic Inspector* to the internet, so all SSH connections, all Slack/Mattermost notifications can be sent via eg. TOR network. Nobody can know where this came from!



## CHAPTER 4

---

From authors

---

Project was started as a part of RiotKit initiative, for the needs of grassroot organizations such as:

- Fighting for better working conditions syndicalist (International Workers Association for example)
- Tenants rights organizations
- Various grassroot organizations that are helping people to organize themselves without authority

*RiotKit Collective*