

Artificial Intelligence Honeypot User Manual



SPHINX

A Universal Cyber Security Toolkit for
Health-Care Industry



Table of contents

1	Introduction.....	3
2	Installation/Deployment	3
2.1	Deployment using Docker	3
2.2	Deployment using Kubernetes	3
3	Explanation of the Honeypot Dashboard	4
4	Basic Case Example 1	7
4.1	Actor	7
4.2	Instructions.....	7
4.3	Expected Outcome	7
5	Case Example 2	9
5.1	Actor	9
5.2	Instructions.....	9
5.3	Expected Outcome	9

Table of figures

Figure 3.1	Honeypot login screen.....	4
Figure 3.2	HP Dashboard User Interface/Home	5
Figure 3.3	Create new HP – pop-up Window	5
Figure 3.4	Retrieve information about an existing HP	6
Figure 3.5	Retrieve HP attack information and MLID data	6
Figure 3.6	Activate/De-activate/Remove HPs	7
Figure 4.2	Expected result	8
Figure 4.1	Values that should be inputed	8
Figure 4.3	The connection on the SSH Service.....	9
Figure 5.1	Values that should be inputed	10
Figure 5.2	The expected Result	11
Figure 5.3	Blank page from HTTP Honeypot.....	11
Figure 5.4	Logs from HTTP Service	12





1 Introduction

The SPHINX AI Honeypot components are not a solution for ensuring security, it is a good tool that supplements other security technologies in order to form an alternative active defence system. More specifically, SPHINX Honeypots aim to lure the attackers in using their provided services and learn from their attacks, in order to afterwards modify and deploy the necessary security controls that will address the detected attacks. To achieve this, SPHINX HPs emulate commonly used services/protocols to serve an attractive for exploitation system to the cyber attackers. Currently, a SPHINX HP consists of six components namely, the HP Core, the HP Message Queue, the HP Data Consumer, the HP Storage DB, the HP Data Processor and the HP API, and is able to support up to four emulated services (i.e. SSH, FTP, HTTP, SMTP). Apart from emulating common services to gather attack information from the intruders, SPHINX HPs also perform sophisticated algorithms in order to properly process the attack information and generate data in a format that AI algorithms can understand, manage and use to detect attacks. Additionally, SPHINX HPs support HTTP (synchronous) communication with other SPHINX components that can access the HP data via the REST API named HP API. Finally, the HP Dashboard was built to improve the usability and facilitate the deployment, maintenance and management of SPHINX HPs.

2 Installation/Deployment

The installation of the SPHINX AI Honeypot components is based on docker images that can be used to deploy the AI Honeypot in any system that include the following prerequisites:

- Linux
- Git
- Docker and Docker-Compose
- Root Access
- Access to the Internet
- Access to Intracom's GitLab Server

2.1 Deployment using Docker

First of all, you should clone the repository of the SPHINX AI Honeypot located in Intracom's GitLab server. You can do this by using this command

```
git clone https://sphinx-repo.intracom-telecom.com/sphinx-project/artificial-intelligence-honeypot/hp
```

After that open a terminal window and go inside the newly created folder (by using the cd command). When inside the directory start the deployment script by typing the command

```
sudo bash deploy-all.sh
```

Now the SPHINX AI Honeypot's docker containers should be up and running. To verify this just open a internet browser window and go to <http://localhost:8084>. You should now see the SPHINX AI Honeypot's Dashboard.

2.2 Deployment using Kubernetes



First of all, you should clone the repository of the SPHINX Knowledge Base Repository located in Intracom's GitLab server. You can do this by using this command

```
git clone https://sphinx-repo.intracom-telecom.com/sphinx-project/artificial-intelligence-honeypot/hp
```

You should also go into the folder `scripts/deployment/KUBERNETES` of the newly created folder (by using the `cd` command). When inside the folder start the Kubernetes deployment using the `honeypot-kubernetes.yaml` file. This will deploy the necessary services, secrets and deployments for the SPHINX AI Honeypot.

3 Explanation of the Honeypot Dashboard

In order to improve the usability and facilitate the deployment, maintenance and management of SPHINX HPs, a dedicated front-end UI, called HP Dashboard was built. The HP Dashboard makes it really easy for SPHINX admins to deploy multiple HP instances on a device (physical or virtual), interact with them and manage them as well.

In order to start interacting with the Dashboard the user has to log into the system using unique credentials (username/password).

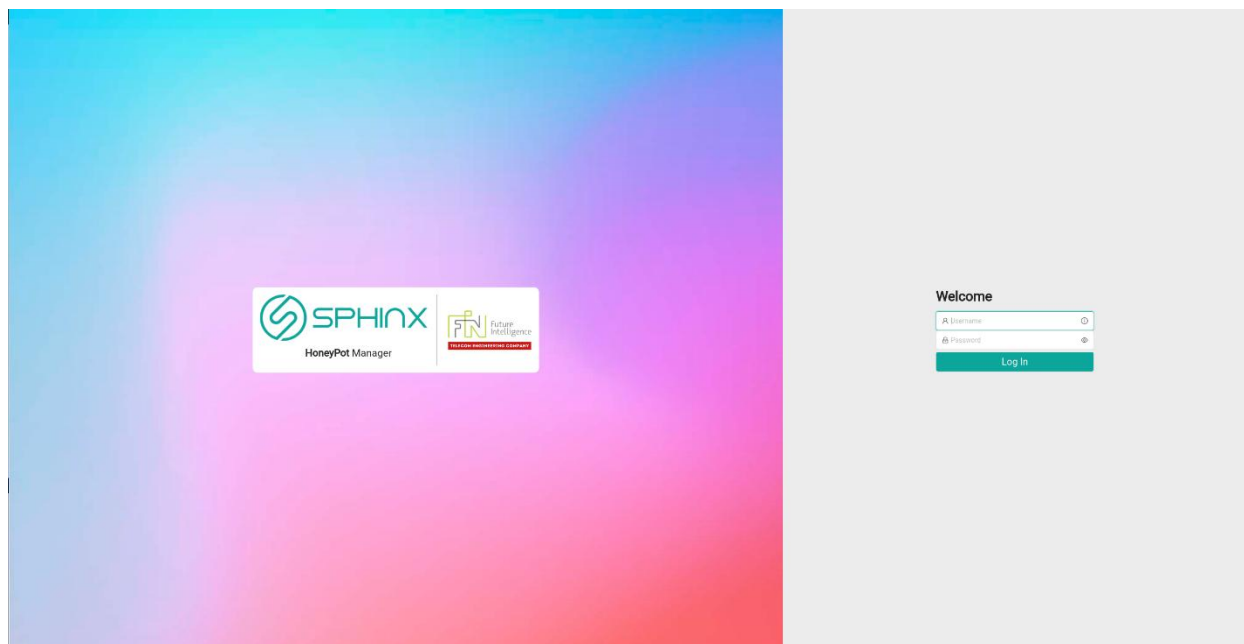


Figure 3.1 Honeypot login screen

In the home page (**Figure 3.2**), users can easily see the number of HPs deployed on a particular system, as well as the number of times that each of the four services supported by the SPHINX HPs at present, exists in the system.

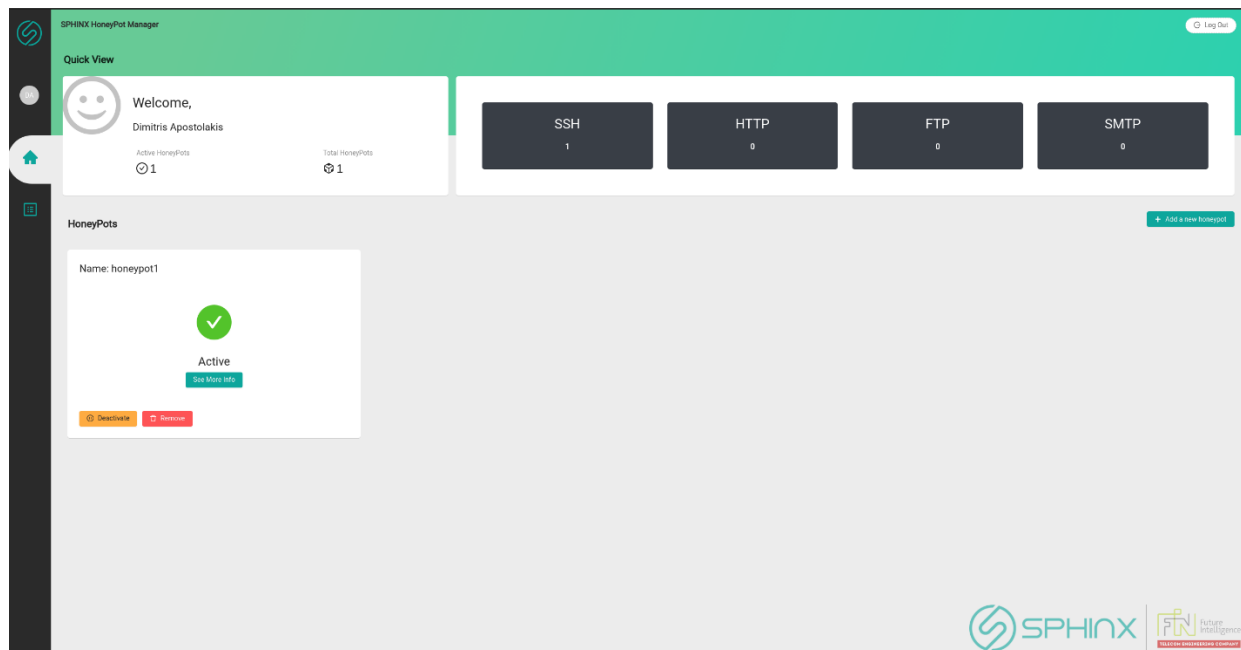


Figure 3.2 HP Dashboard User Interface/Home

By selecting “Add a new honeypot” users are able to create HPs really fast; in the pop-up window that appears on the screen (**Figure 3.3**) the user must simply provide a name for the new HP instance, and also define the emulated services within this instance and the corresponding ports to be used.

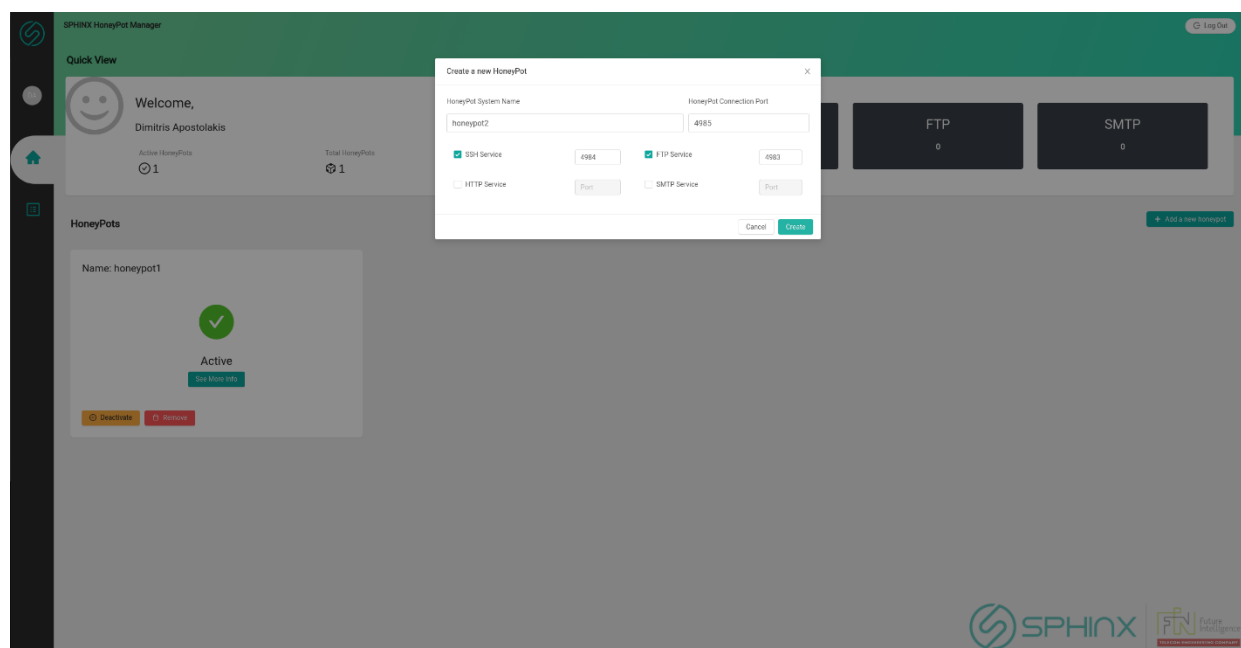


Figure 3.3 Create new HP – pop-up Window

Users can always retrieve detailed information about a particular HP instance created, by just pressing the “See More Info” button of the respective instance-pane displayed on the home page (**Figure 3.4**).

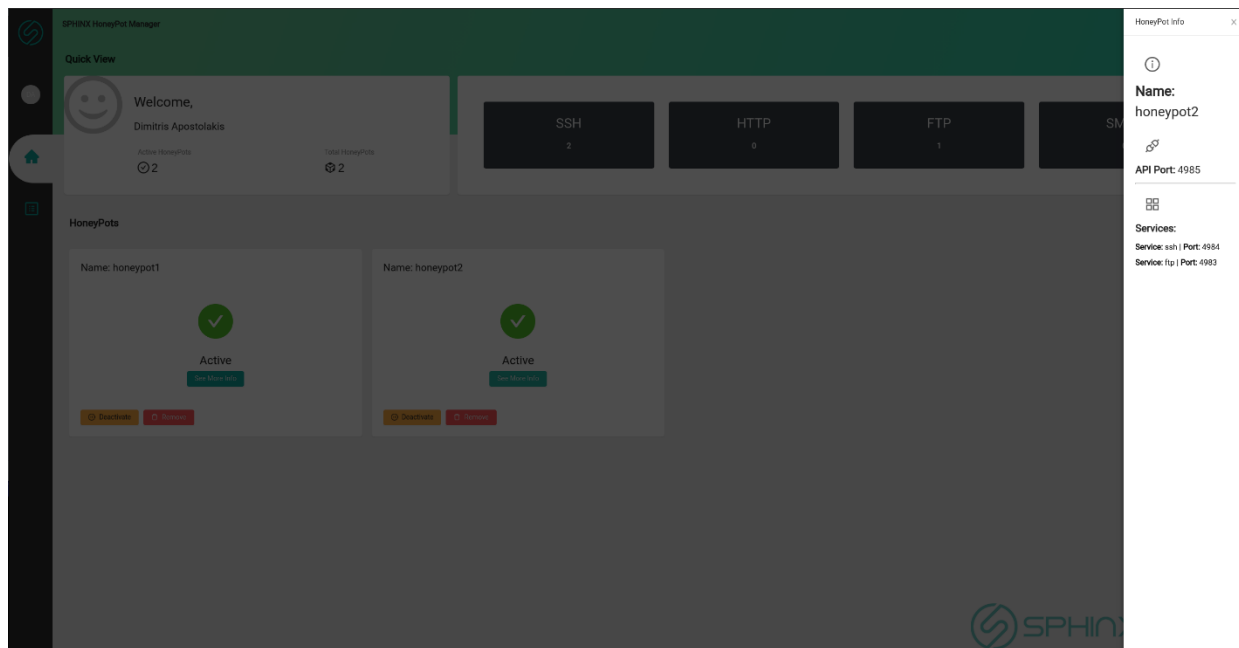


Figure 3.4 Retrieve information about an existing HP

Furthermore, the HP Dashboard provides direct access to the attack information gathered by a deployed HP and the related MLID data generated (**Figure 3.5**).

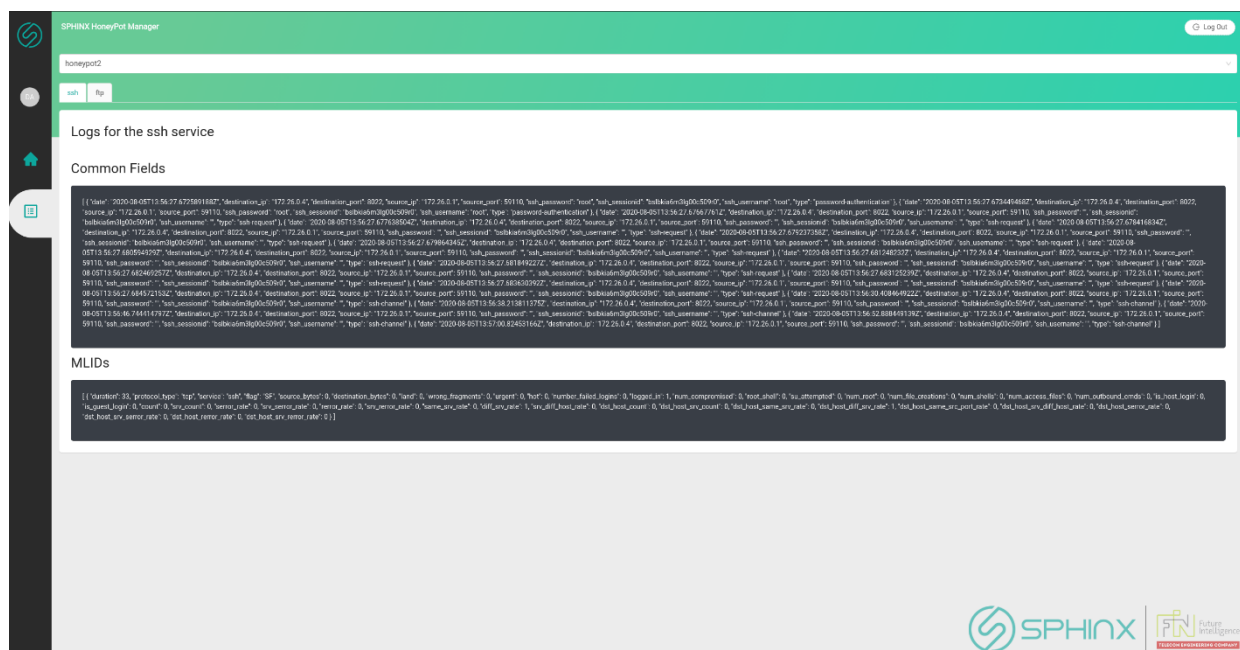


Figure 3.5 Retrieve HP attack information and MLID data

Finally, admins can easily activate/deactivate or remove deployed HPs through the HP Dashboard, at any time, by just pressing the corresponding buttons in the home page (**Figure 3.6**)

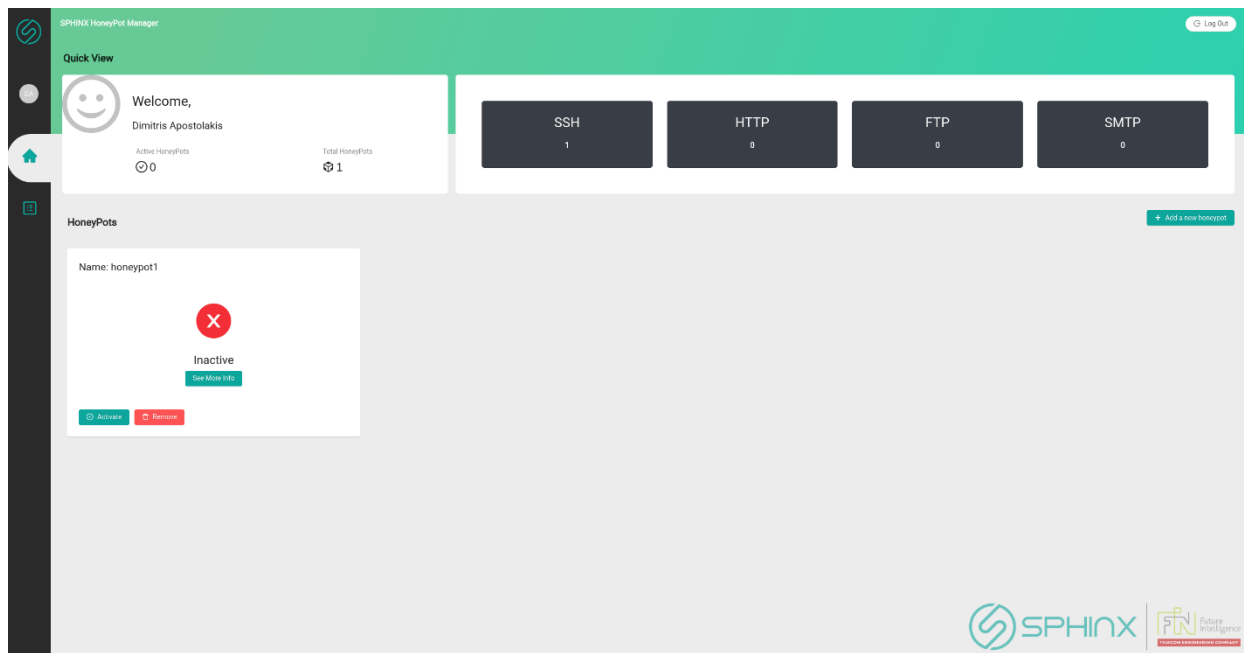


Figure 3.6 Activate/De-activate/Remove HPs

4 Basic Case Example 1

For this tutorial we have two case examples for the SPHINX AI Honeypot usage. These case examples should help familiarize the reader with the SPHINX AI Honeypot's usage and interface.

4.1 Actor

The actor for this procedure will be an Advanced IT Advisor / Personnel of a Hospital. The actor should have access to the SPHINX AI Honeypot's Dashboard and the permission to create new Honeyspots.

4.2 Instructions

For this case example you should enter the SPHINX AI Honeypot's Dashboard and deploy a new Honeypot with SSH and HTTP Services in ports 8022 and 8081. The Honeyspots API Port should be 8060. You should then open a terminal window and connect via SSH to the honeypot you created with the command `ssh root@localhost -p 8022`. When it asks for ssh password, input `root`.

4.3 Expected Outcome

If everything has been followed correctly, the values in the New Honeypot modal should match the values in the following image.

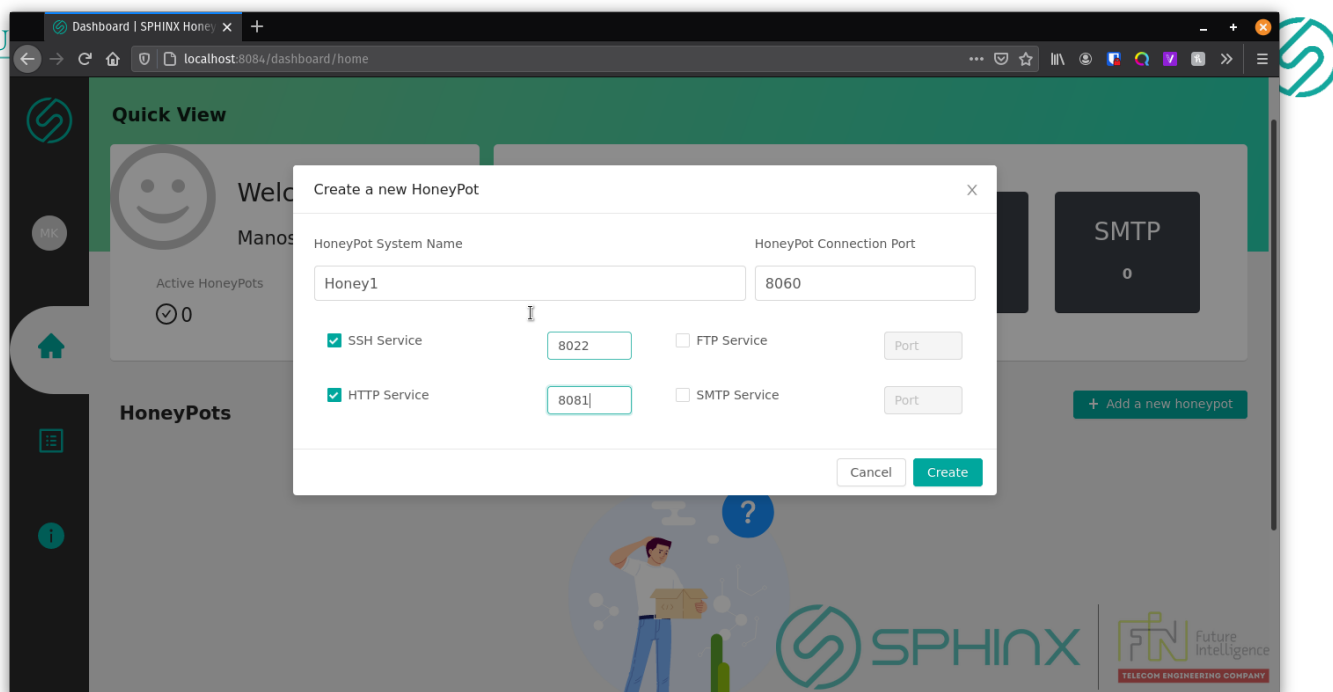


Figure 4.1 Values that should be inputed

If the docker containers have been deployed successfully, the dashboard should now present your new honeypot in the Home Page. By pressing the See More Info button you should see the exact values as the following image.

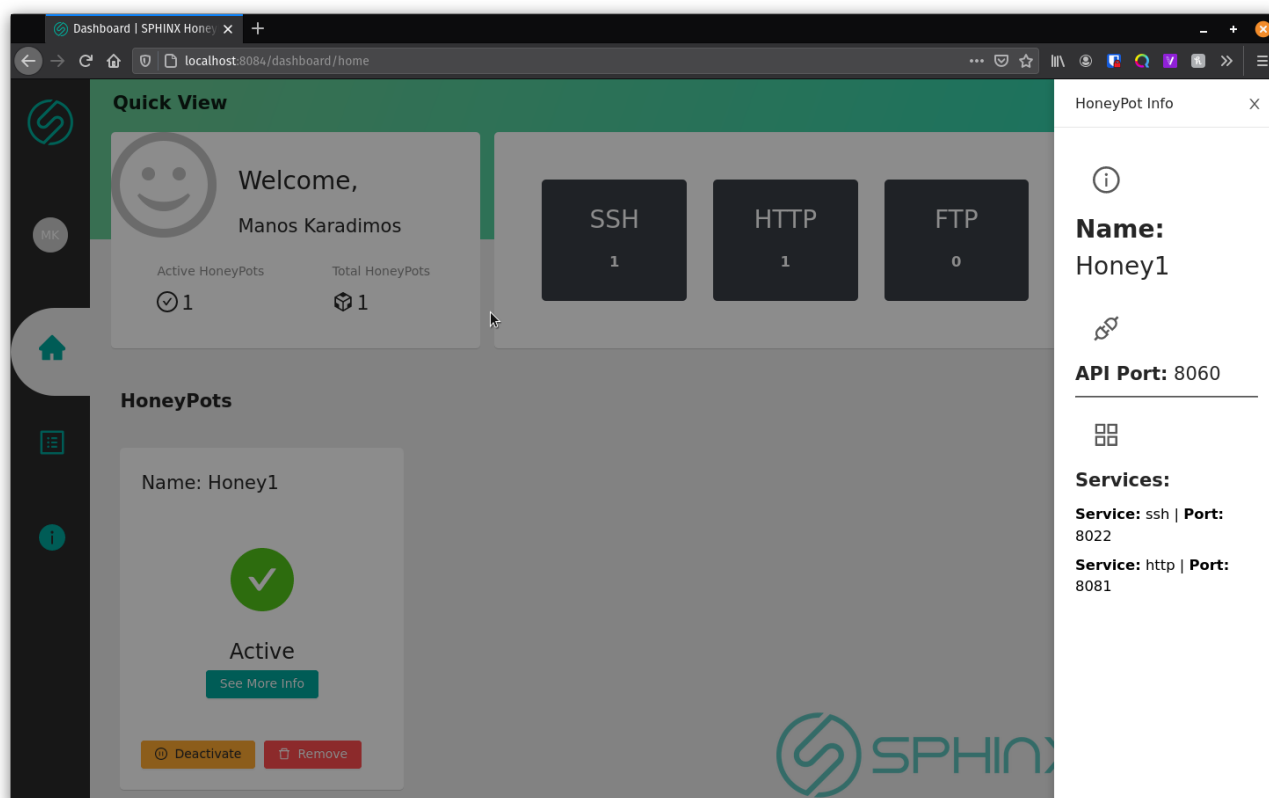


Figure 4.2 Expected result

If you now open the terminal window and connect to the SSH Service you should see this outcome.



```
techgeekster@tw3-eisei: ~  
RSA key fingerprint is SHA256:yqKCZTc16I2xi+KmTVrJrRSHTst6H/FmmskRIImoyIlk.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '[localhost]:8022' (RSA) to the list of known hosts.  
root:root@localhost's password:  
Permission denied, please try again.  
root:root@localhost's password:  
  
techgeekster@tw3-eisei:~$ ssh root@localhost -p 8022  
root@localhost's password:  
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-31-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
524 packages can be updated.  
270 updates are security updates.  
  
-----  
Ubuntu 16.04.1 LTS                                built 2016-12-10  
-----  
last login: Sun Nov 19 19:40:44 2017 from 172.16.84.1  
root@host:~$
```

Figure 4.3 The connection on the SSH Service.

5 Case Example 2

5.1 Actor

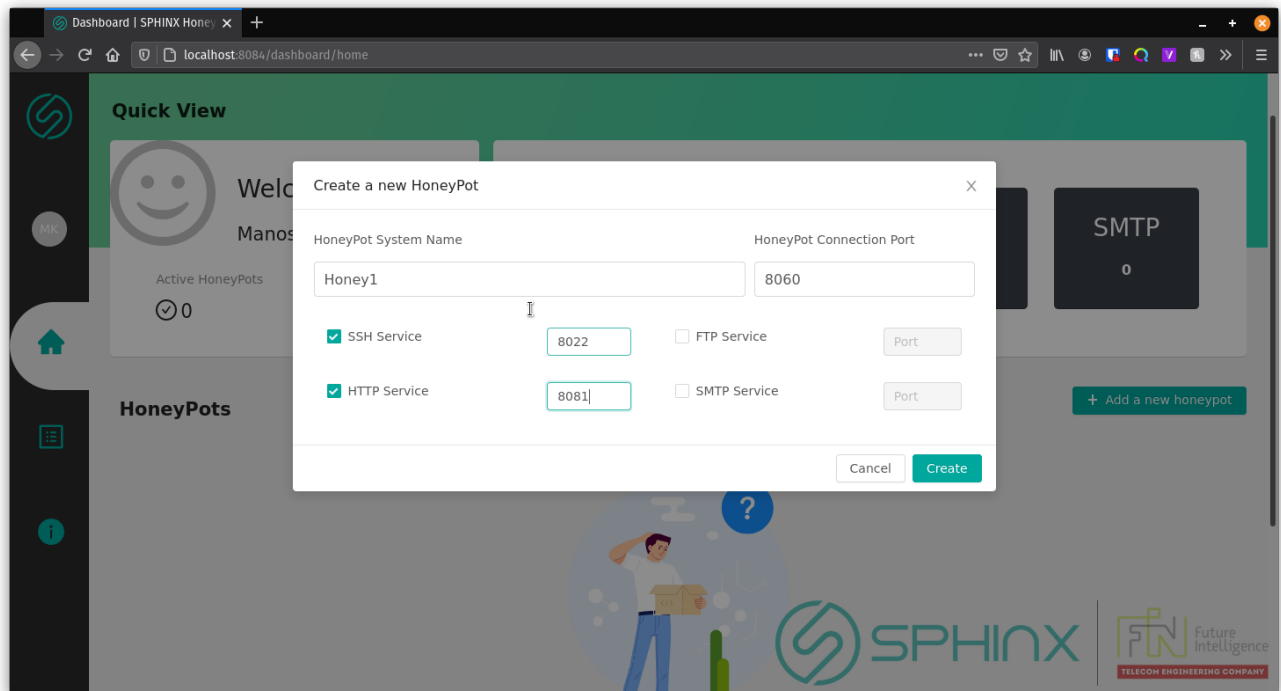
The actor for this procedure will be the Main IT Manager of a Hospital. The actor should have access to the SPHINX AI Honeypot's Dashboard and the permission to create new Honeypots and the ability to read the logs of the Honeypots.

5.2 Instructions

For this case example you should enter the SPHINX AI Honeypot's Dashboard and deploy a new Honeypot with SSH and HTTP Services in ports 8022 and 8081. The Honeypots API Port should be 8060. You should then open a new browser tab and go to <http://localhost:8081>. A blank page should appear. Then you should go to the logs and see the http service logs.

5.3 Expected Outcome

If everything has been followed correctly, the values in the New Honeypot modal should match the values in the following image.



The screenshot shows a web browser window with the URL `localhost:8084/dashboard/home`. The dashboard has a sidebar with navigation icons and a main content area. A modal window titled "Create a new HoneyPot" is open in the center. The modal contains the following fields and options:

- HoneyPot System Name:** A text input field containing "Honey1".
- HoneyPot Connection Port:** A text input field containing "8060".
- SSH Service:** A checked checkbox with a corresponding port input field containing "8022".
- FTP Service:** An unchecked checkbox with a corresponding "Port" input field.
- HTTP Service:** A checked checkbox with a corresponding port input field containing "8081".
- SMTP Service:** An unchecked checkbox with a corresponding "Port" input field.

At the bottom of the modal are "Cancel" and "Create" buttons. The background dashboard shows a "Quick View" section with a welcome message, "Active HoneyPots" count (0), and a "HoneyPots" section. There is also an "SMTP" status card showing "0" and an "Add a new honeypot" button.

Figure 5.1 Values that should be inputed

If the docker containers have been deployed successfully, the dashboard should now present your new honeypot in the Home Page. By pressing the See More Info button you should see the exact values as the following image.

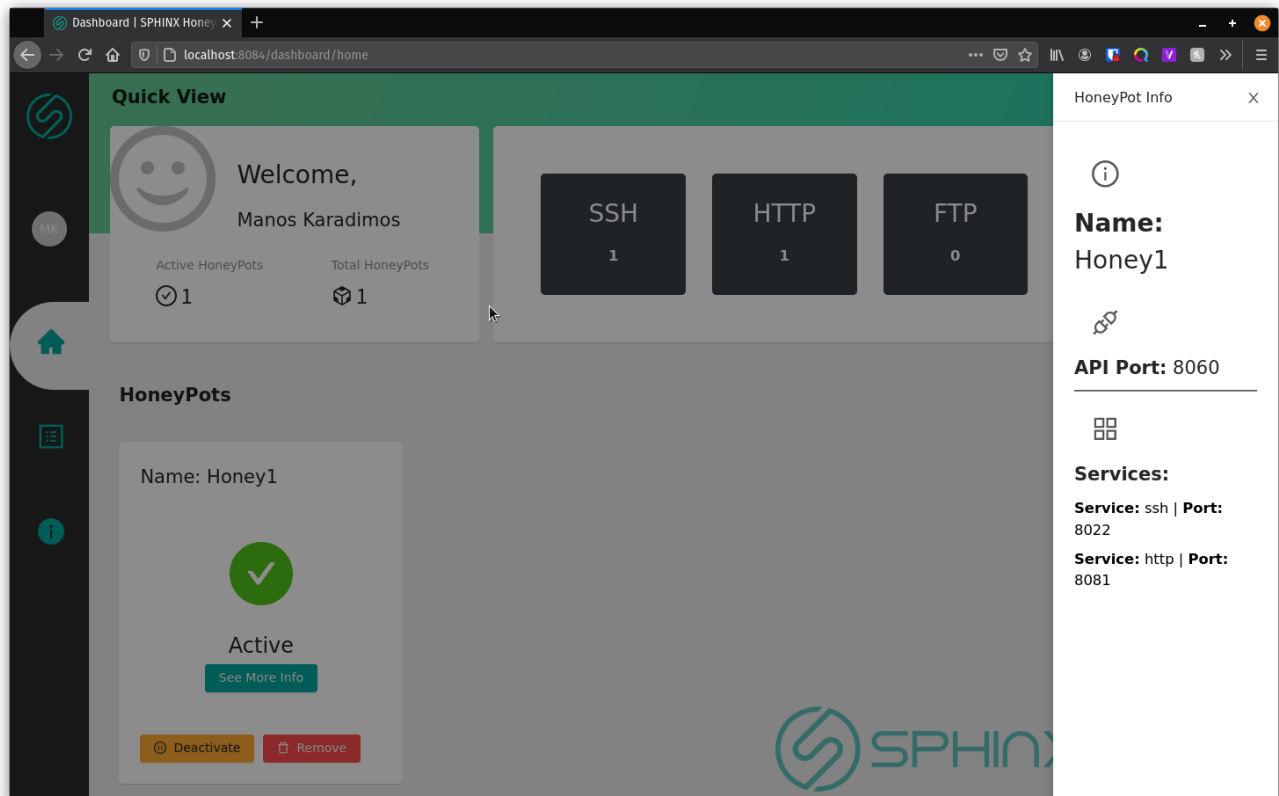


Figure 5.2 The expected Result

If you now open a new tab and go to the HTTP Service's url you should see a blank page. This is normal.

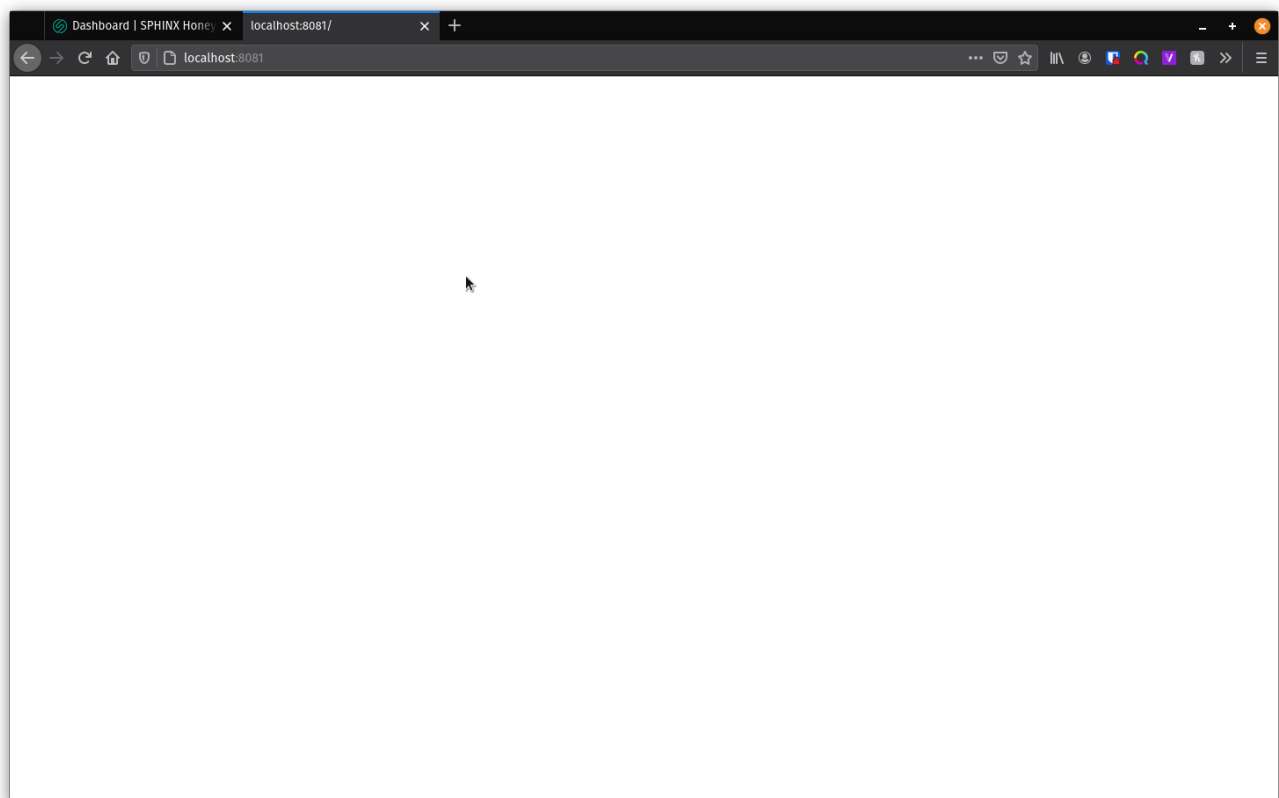


Figure 5.3 Blank page from HTTP Honeypot



Now go to the SPHINX AI HoneyPot's Dashboard and go to the Logs page. Select the previously made honeypot and the HTTP Service. You should now see the logs for the HTTP Service as well as the MLIDs for that service

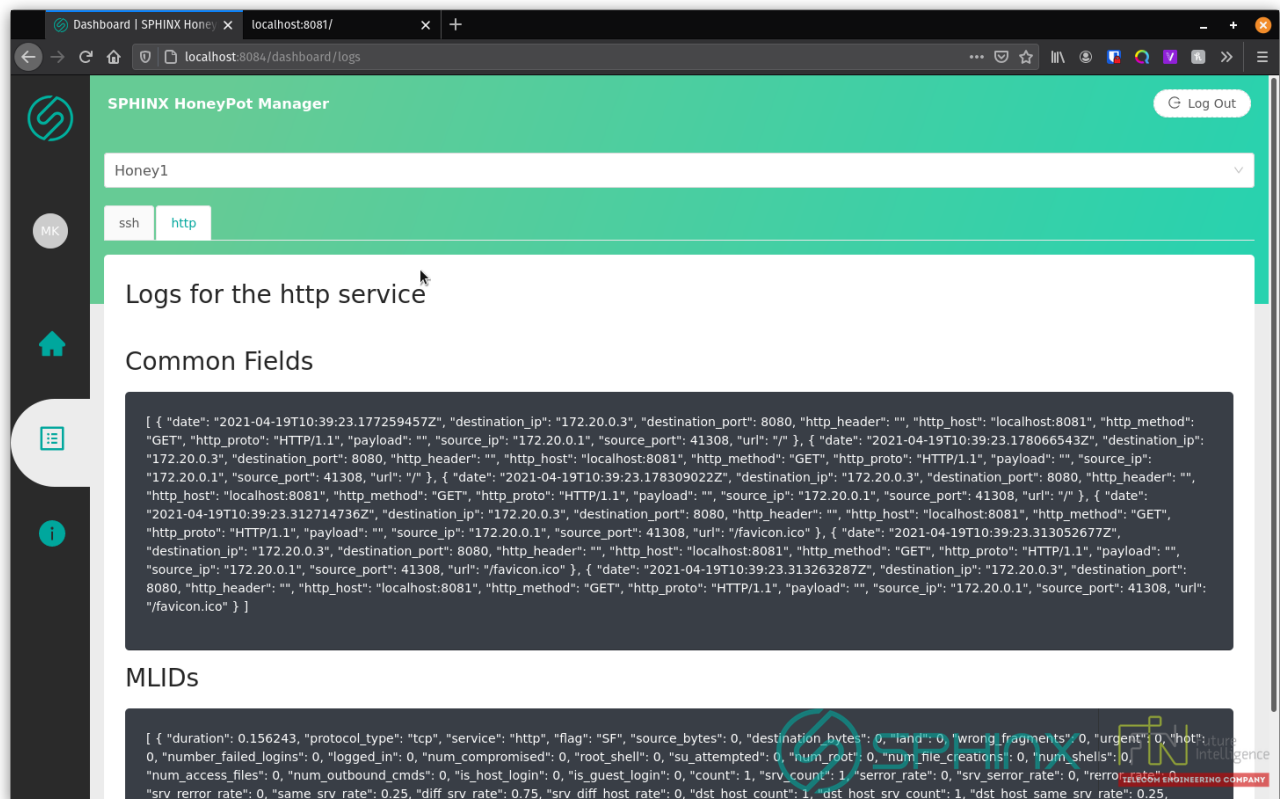


Figure 5.4 Logs from HTTP Service