OPEN SOURCE HONEYPOT MANAGEMENT SYSTEM

NUTANIX YOUR ENTERPRISE CLOUD

01

STUDENTS: ADIL ISLAM, KALYANI MARATHE, NING WANG, SHAWN HSIAO

Security & Honeypots

Over a third of Americans have been the victim of a hack or stolen identity [1]. As our data becomes increasingly valuable in a world of continuously improving technology, security and preventative measures are critical to modern safety. Thus, our capstone project works to improve user security and safety in the situation of a hack. We are working to create a honeypot management system which can utilize data from various honeypot sensors and provide logs of any occurring attacks and provide continuous feedback to an administrator. Furthermore, our goal is to make this system open source and user friendly!

Honeypots Used

Heralding:

- Honeypot that collects credentials
- Protocols that are supported: ftp, telnet, ssh, rdp, http, https, pop3, pop3s, imap, imaps, smtp, vnc, postgresql and socks5.
- Need Python 3.6.0 or higher
- Logs:Log_session.json, log_auth.csv, log_session.csv

Zeek:

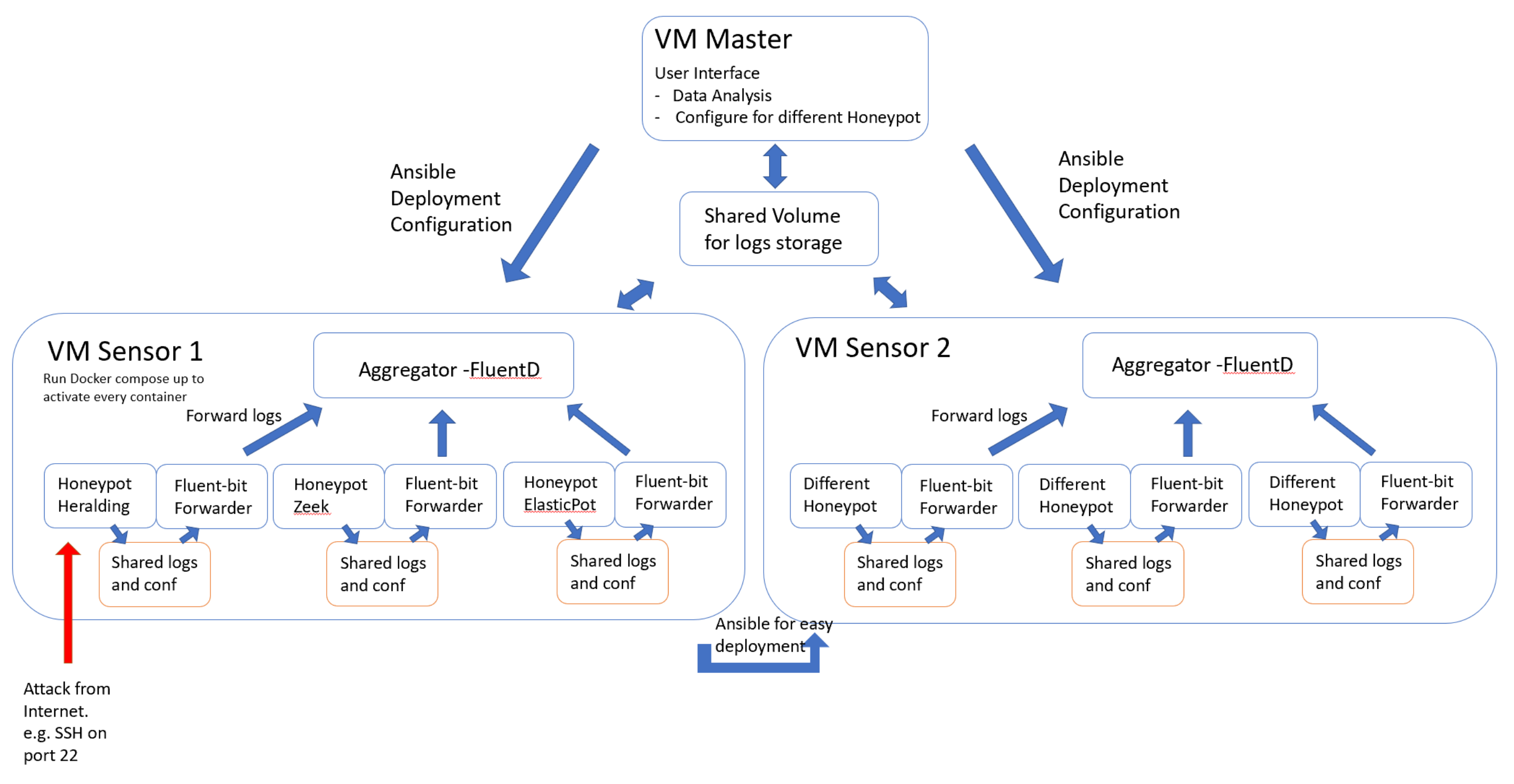
- Free, open-source network analysis framework
- Analyzes network traffic and generates event logs when "something" happens
  - Zeek action
  - Network action
- Efficient and can be used at high-performance / large-scale
- Flexible towards a variety of security approaches

ElasticPot:

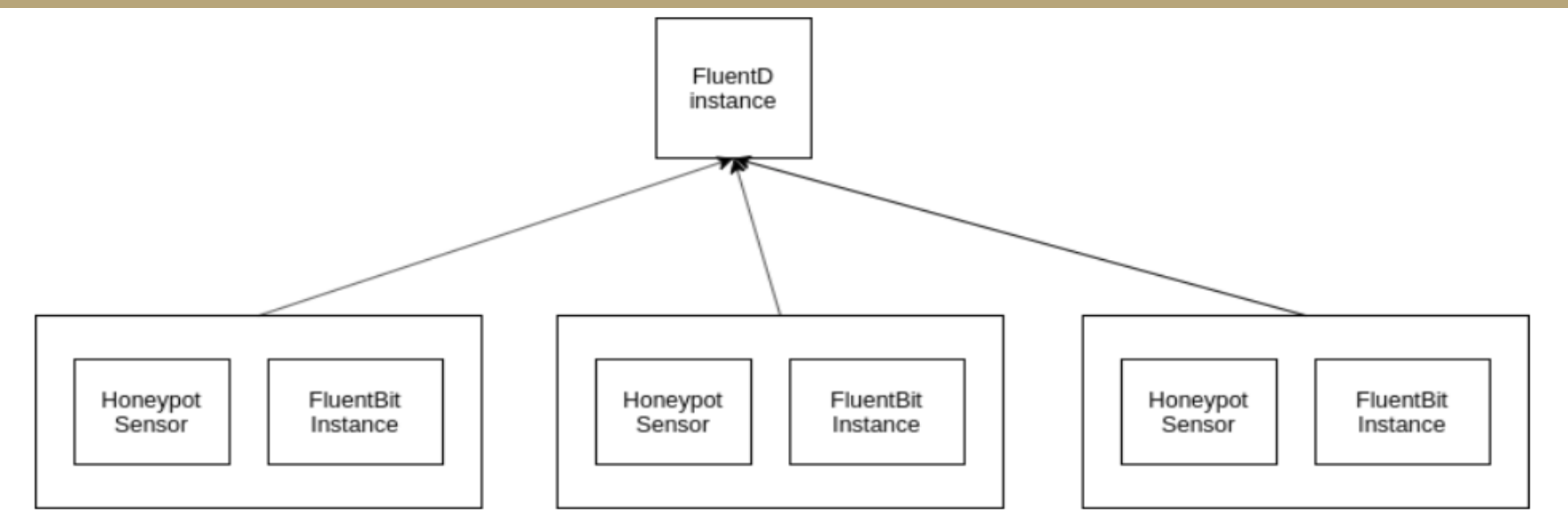
- Elastic Search: Application search, Website search, Logging and log analytics, etc.
- Exp: Elasticpot blocks unauthenticated users who are trying to log in to the LogFile of Elastic Search;
- Elasticpot simulates the Remote Code Execution(RCE) loophole, uses fake functions to answer JSON format message generated by vulnerable ES nodes.
- Need Python 3.
- Generates JSON log file, also write to the original Elastic Search's log file.

Technical Approach & System Architecture

A running honeypot sensor is composed of three core components: console, sensor, and honeypot. The interaction between these components is diagrammed below.



Log Collection Framework



The log structure stores collected logs on the master node and sorts them. This structure resembles a tree, where the top layer is a subdirectory within the master node and the second layer contains folders of various honeypots. Then, within each honeypot is a third layer which contains a folder for each month. Finally, within each month is the fourth layer which redirects the output logs of fluentd daily.

Fluentbit/Fluentd

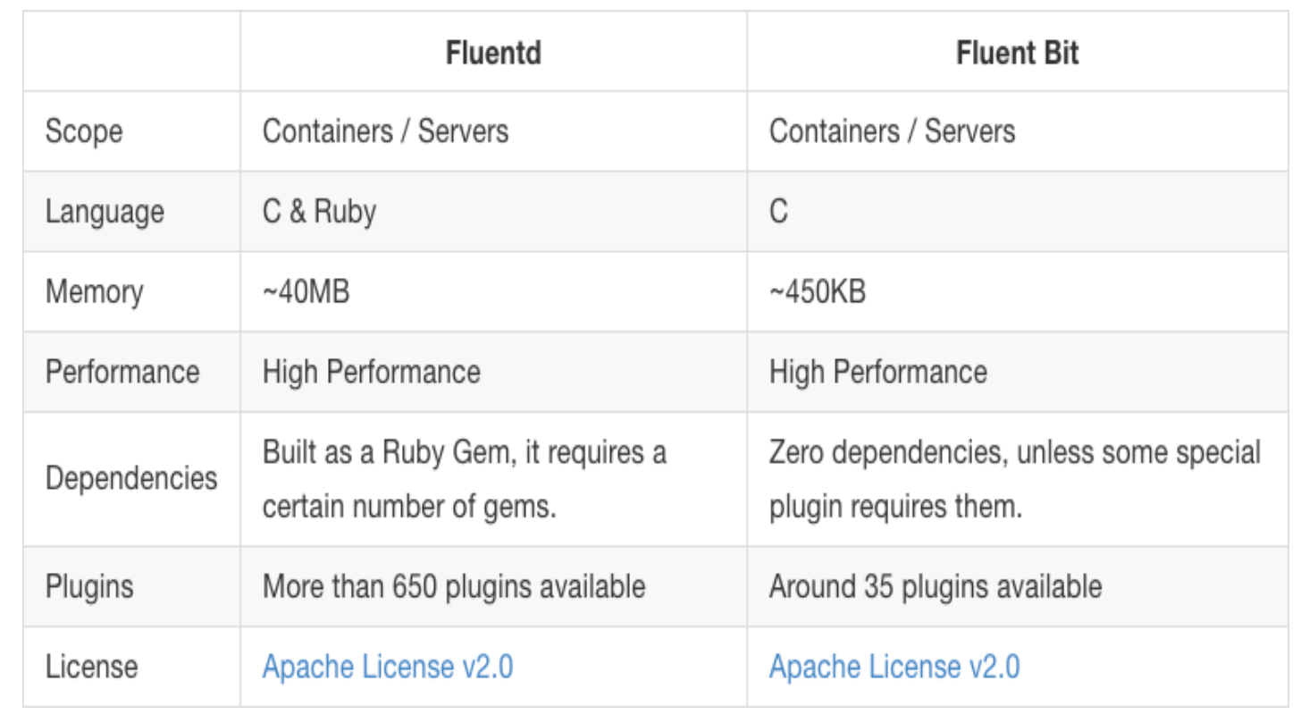
Fluent Bit :
- To collect the logs from each honeypot sensor and forwarding those logs to fluentD.
-A lightweight log processor
FluentD :
-Used for large scale applications and to handle heavy throughput, aggregating multiple inputs.
-FluentBit is not as flexible as fluentD but has many different input and output plugins.

Configuring Fluentbit:

```
[INPUT]
    Name       tail
    Tag  log.local
    Path       path/to/log_session.json


[OUTPUT]

    Name          forward
    Match         *
    Host          127.0.0.1
    Port          24224
```

Configuring FluentD:

```
<source>
  @type forward
  @id input_forward
  bind 0.0.0.0
  port 24224
</source>

# File output
# match tag=local.** and write to file
<match log.local.**>
  @type file
  @id output_file
  path /var/log/td-agent/access/output.txt
</match>
```



	Fluentd	Fluent Bit
Scope	Containers / Servers	Containers / Servers
Language	C & Ruby	C
Memory	~40MB	~450KB
Performance	High Performance	High Performance
Dependencies	Built as a Ruby Gem, it requires a certain number of gems.	Zero dependencies, unless some special plugin requires them.
Plugins	More than 650 plugins available	Around 35 plugins available
License	Apache License v2.0	Apache License v2.0

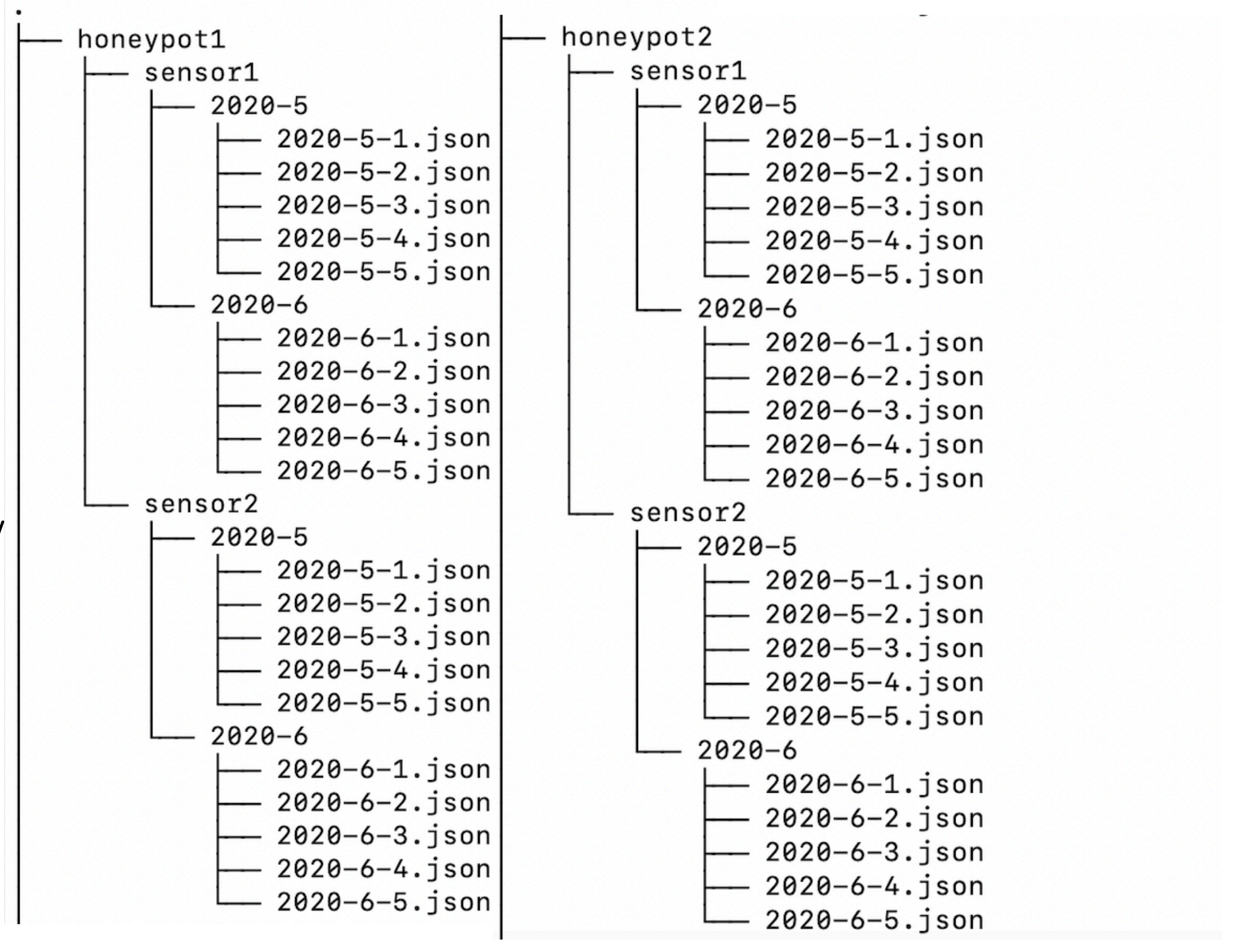
File Structure

The log file structure can be seen within the image on the right. We designed the structure through FluentD and Ansible, and the actual files are stored in the master VM. Here is a description of some of the components within it:

•honeypots:
Each honeypot own a directory for log storage in master VM.

•sensors:
Within each honeypot the logs are stored respectively according to from which sensors the logs were generated.

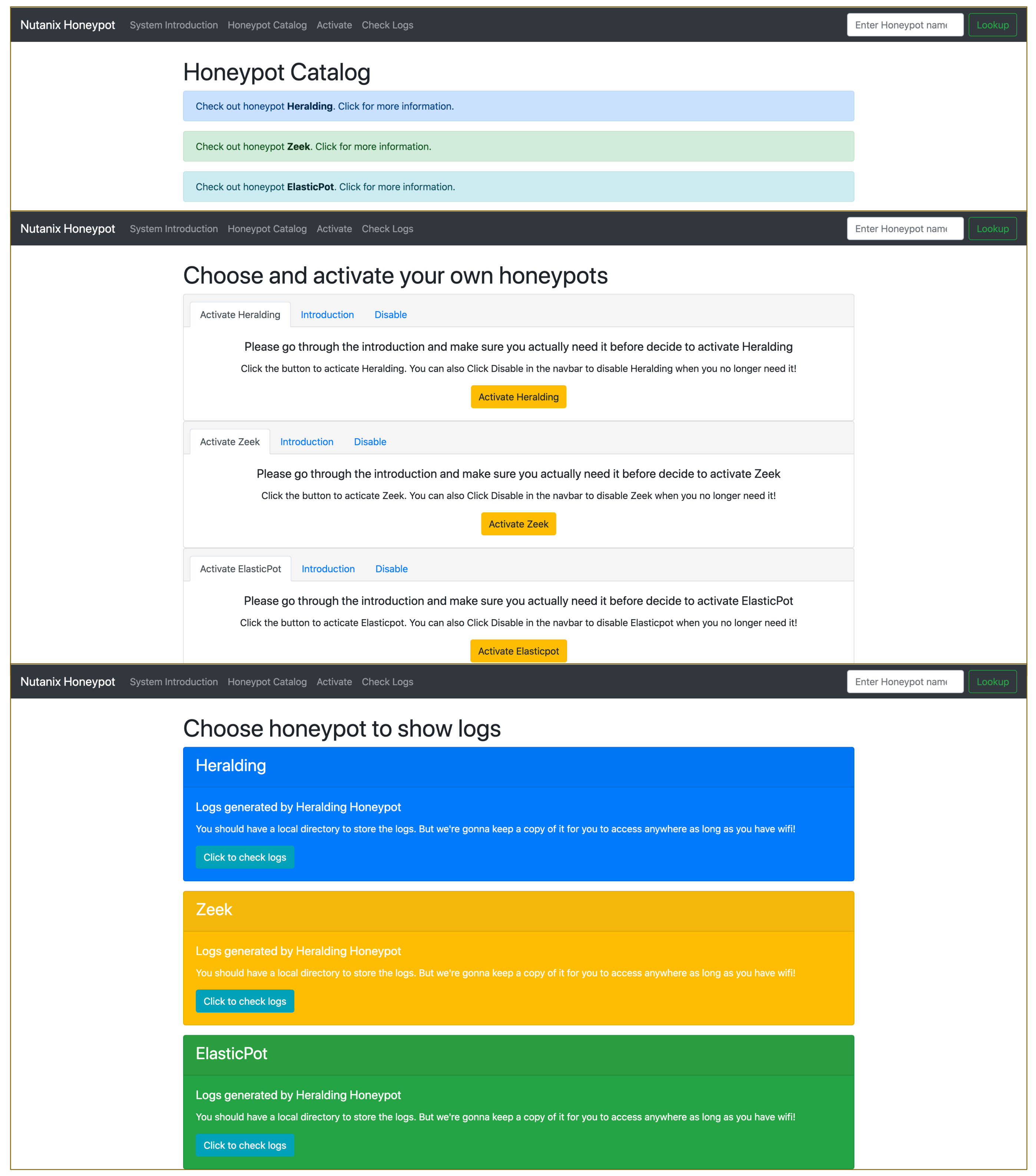
•months:
For each sensor the .json log files generated each day are stored month by month, so the users would have an overall idea about the distribution of the attacks.



```
.
├── honeypot1
│   ├── sensor1
│   │   ├── 2020-5
│   │   │   ├── 2020-5-1.json
│   │   │   ├── 2020-5-2.json
│   │   │   ├── 2020-5-3.json
│   │   │   ├── 2020-5-4.json
│   │   │   └── 2020-5-5.json
│   │   └── 2020-6
│   │       ├── 2020-6-1.json
│   │       ├── 2020-6-2.json
│   │       ├── 2020-6-3.json
│   │       ├── 2020-6-4.json
│   │       └── 2020-6-5.json
│   └── sensor2
│       ├── 2020-5
│       │   ├── 2020-5-1.json
│       │   ├── 2020-5-2.json
│       │   ├── 2020-5-3.json
│       │   ├── 2020-5-4.json
│       │   └── 2020-5-5.json
│       └── 2020-6
│           ├── 2020-6-1.json
│           ├── 2020-6-2.json
│           ├── 2020-6-3.json
│           ├── 2020-6-4.json
│           └── 2020-6-5.json
├── honeypot2
│   ├── sensor1
│   │   ├── 2020-5
│   │   │   ├── 2020-5-1.json
│   │   │   ├── 2020-5-2.json
│   │   │   ├── 2020-5-3.json
│   │   │   ├── 2020-5-4.json
│   │   │   └── 2020-5-5.json
│   │   └── 2020-6
│   │       ├── 2020-6-1.json
│   │       ├── 2020-6-2.json
│   │       ├── 2020-6-3.json
│   │       ├── 2020-6-4.json
│   │       └── 2020-6-5.json
│   └── sensor2
│       ├── 2020-5
│       │   ├── 2020-5-1.json
│       │   ├── 2020-5-2.json
│       │   ├── 2020-5-3.json
│       │   ├── 2020-5-4.json
│       │   └── 2020-5-5.json
│       └── 2020-6
│           ├── 2020-6-1.json
│           ├── 2020-6-2.json
│           ├── 2020-6-3.json
│           ├── 2020-6-4.json
│           └── 2020-6-5.json
```

Front End / UI

The UI was built to give users to have an intuitive panel to pick, setup, and monitor the honeypots according to their needs.
The welcome homepage and introduction page give users a sense of what the honeypot management system is and how they can benefit from it.
The Honeypot Catalog allows users to choose which honeypots they wish to deploy, and buttons in the catalog lead to the activate page where users can activate desired honeypots.
Finally the Check Logs page shows the attack logs generated from each honeypot.



Honeypot Catalog

Check out honeypot **Heralding**. Click for more information.

Check out honeypot **Zeek**. Click for more information.

Check out honeypot **ElasticPot**. Click for more information.

Choose and activate your own honeypots

Please go through the introduction and make sure you actually need it before decide to activate Heralding

Click the button to acticate Heralding. You can also Click Disable in the navbar to disable Heralding when you no longer need it!

Please go through the introduction and make sure you actually need it before decide to activate Zeek

Click the button to acticate Zeek. You can also Click Disable in the navbar to disable Zeek when you no longer need it!

Please go through the introduction and make sure you actually need it before decide to activate ElasticPot

Click the button to acticate Elasticpot. You can also Click Disable in the navbar to disable Elasticpot when you no longer need it!

Choose honeypot to show logs

Heralding

Logs generated by Heralding Honeypot

You should have a local directory to store the logs. But we're gonna keep a copy of it for you to access anywhere as long as you have wifi!

Zeek

Logs generated by Heralding Honeypot

You should have a local directory to store the logs. But we're gonna keep a copy of it for you to access anywhere as long as you have wifi!

ElasticPot

Logs generated by Heralding Honeypot

You should have a local directory to store the logs. But we're gonna keep a copy of it for you to access anywhere as long as you have wifi!

Future Work, References, and Acknowledgments

- Complete integration with Docker via docker compose.
- Improve front end UI to include a flow chart/trend of attacks and predictions based upon past events.
- Connect to cloud service to run remotely.
- Comprehensive API to interact with management console.
- Integrate raspberry pi support.

Faculty: Radha Poovendran, Bhaskar Ramasubramanian
Graduate Students: Kalyani Marathe, Ning Wang, Shawn Hsiao
Undergraduate Students: Adil Islam

[1] https://nypost.com/2019/08/15/more-than-1-in-3-americans-have-been-hacked-or-had-their-identity-stolen-survey/

ELECTRICAL & COMPUTER ENGINEERING
UNIVERSITY *of* WASHINGTON

ADVISORS: YUHUI HUANG, BALA NEERUMALLA, BHASKAR RAMASUBRAMANIAN, RADHA POOVENDRAN

SPONSOR: NUTANIX