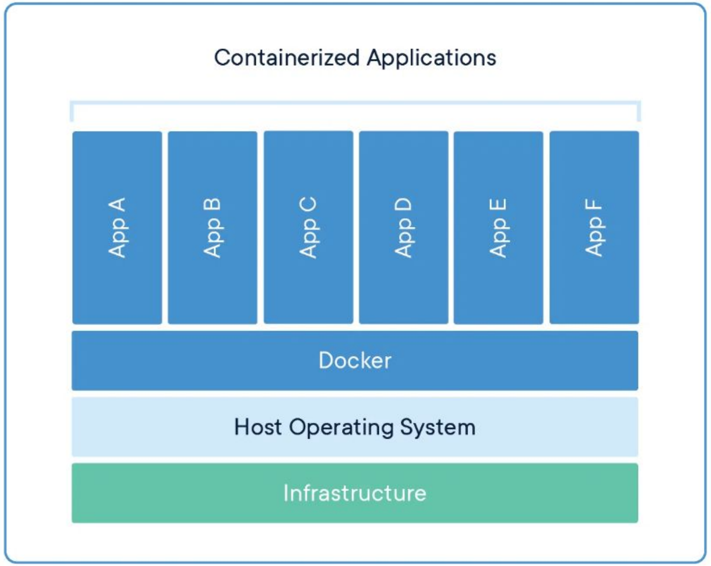
NYU

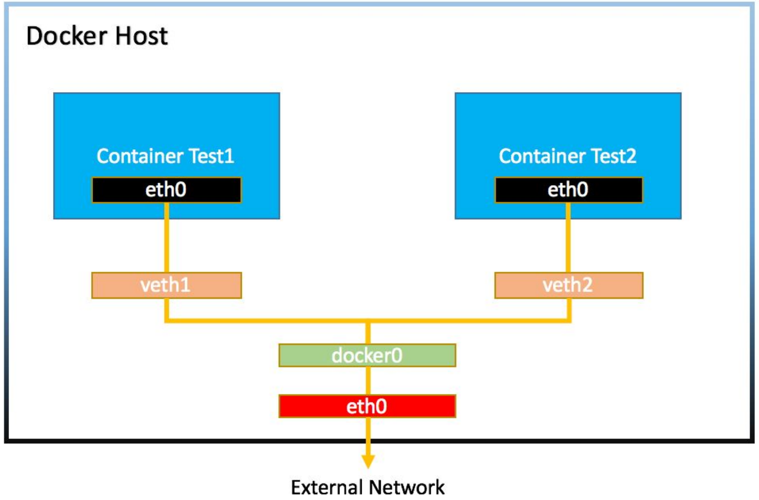
Smart Bridge for Automated DDoS Detection in Docker Networks

Sakshi Kulkarni (smk8939)
Chandana Srinivasa Yatisha (cs7074)
Amrutha Patil (ap7982)

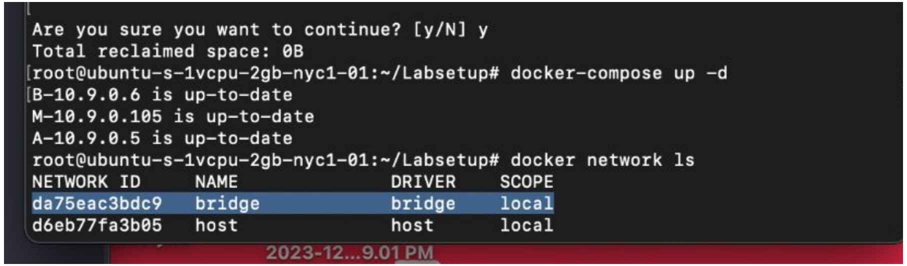
Dockers and Containers

- What is a Docker and what are Containers?
- How do containers interact with each other?

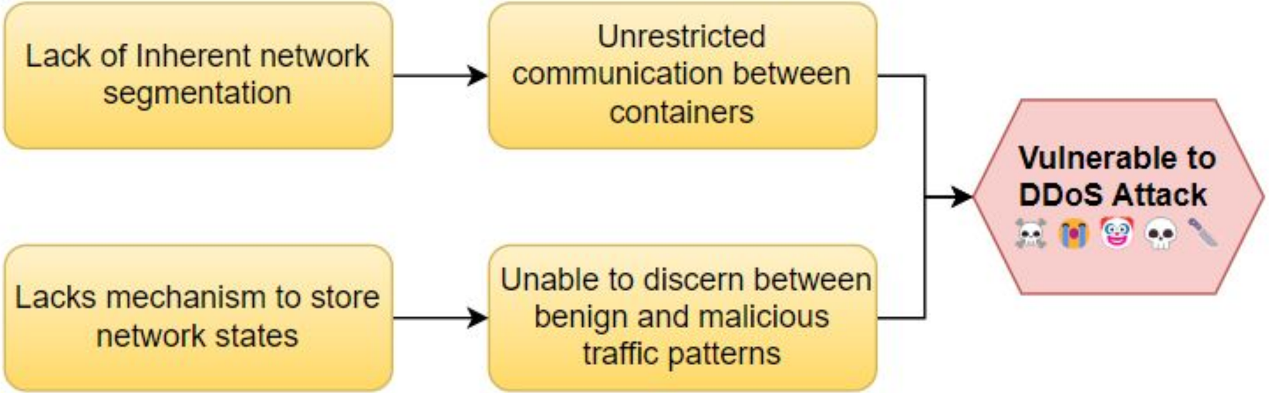




Default Bridge Docker0



```
[
 Are you sure you want to continue? [y/N] y
 Total reclaimed space: 0B
[root@ubuntu-s-1vcpu-2gb-nyc1-01:~/Labsetup# docker-compose up -d
[B-10.9.0.6 is up-to-date
 M-10.9.0.105 is up-to-date
 A-10.9.0.5 is up-to-date
 root@ubuntu-s-1vcpu-2gb-nyc1-01:~/Labsetup# docker network ls
 NETWORK ID     NAME                 DRIVER    SCOPE
 da75eac3bdc9   bridge               bridge    local
 d6eb77fa3b05   host                 host      local
```



Lack of Inherent network segmentation

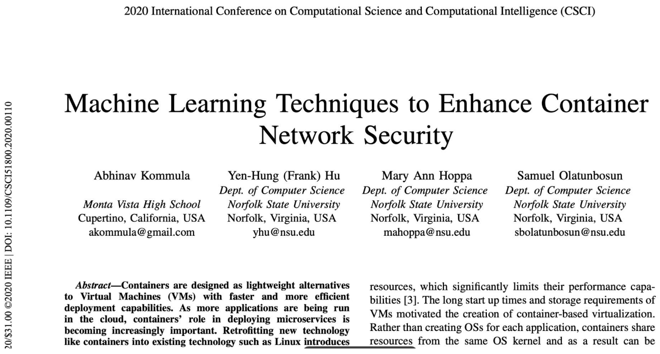
Unrestricted communication between containers

Lacks mechanism to store network states

Unable to discern between benign and malicious traffic patterns

Vulnerable to DDoS Attack ☠️ 😭 🤡 💀 🔪

Existing Solutions

- 

2020 International Conference on Computational Science and Computational Intelligence (CSCI)

20/$31.00 ©2020 IEEE | DOI: 10.1109/CSCI51800.2020.00110

Machine Learning Techniques to Enhance Container Network Security

Abhinav Kommula
Monta Vista High School
Cupertino, California, USA
akommula@gmail.com

Yen-Hung (Frank) Hu
Dept. of Computer Science
Norfolk State University
Norfolk, Virginia, USA
yhu@nsu.edu

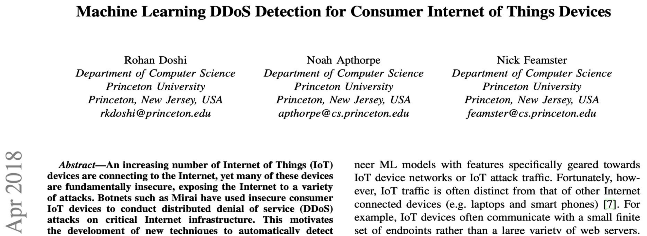
Mary Ann Hoppa
Dept. of Computer Science
Norfolk State University
Norfolk, Virginia, USA
mahoppa@nsu.edu

Samuel Olatunbosun
Dept. of Computer Science
Norfolk State University
Norfolk, Virginia, USA
sbolatunbosun@nsu.edu

***Abstract*—Containers are designed as lightweight alternatives to Virtual Machines (VMs) with faster and more efficient deployment capabilities. As more applications are being run in the cloud, containers' role in deploying microservices is becoming increasingly important. Retrofitting new technology like containers into existing technology such as Linux introduces**

resources, which significantly limits their performance capabilities [3]. The long start up times and storage requirements of VMs motivated the creation of container-based virtualization. Rather than creating OSs for each application, containers share resources from the same OS kernel and as a result can be

Key Points: Stateful Bridge; ARP Spoofing; Stores MAC Addresses

- 

Machine Learning DDoS Detection for Consumer Internet of Things Devices

Rohan Doshi
Department of Computer Science
Princeton University
Princeton, New Jersey, USA
rkdoshi@princeton.edu

Noah Apthorpe
Department of Computer Science
Princeton University
Princeton, New Jersey, USA
apthorpe@cs.princeton.edu

Nick Feamster
Department of Computer Science
Princeton University
Princeton, New Jersey, USA
feamster@cs.princeton.edu

Apr 2018

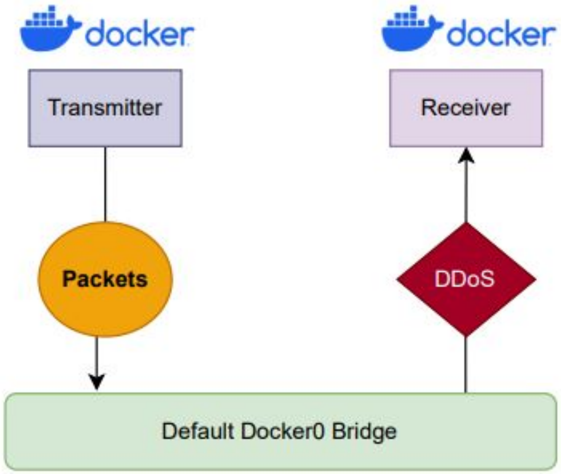
***Abstract*—An increasing number of Internet of Things (IoT) devices are connecting to the Internet, yet many of these devices are fundamentally insecure, exposing the Internet to a variety of attacks. Botnets such as Mirai have used insecure consumer IoT devices to conduct distributed denial of service (DDoS) attacks on critical Internet infrastructure. This motivates the development of new techniques to automatically detect**

neer ML models with features specifically geared towards IoT device networks or IoT attack traffic. Fortunately, however, IoT traffic is often distinct from that of other Internet connected devices (e.g. laptops and smart phones) [7]. For example, IoT devices often communicate with a small finite set of endpoints rather than a large variety of web servers.

Key Points: Packet Level; ML based binary classification of DDoS; KNN

Hypothesis: Auto-learning Stateful Bridge

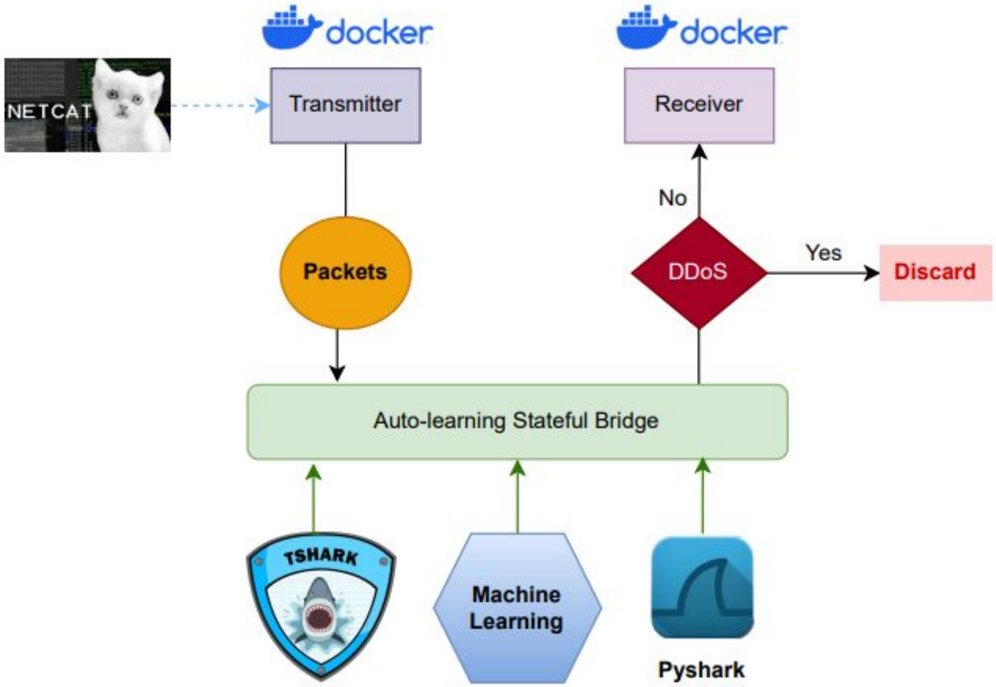
Hypothesis: Introducing auto-learning and state awareness using Machine Learning within a docker network bridge to analyze network traffic patterns could significantly enhance DDoS attack defense compared to the default docker0 bridge.



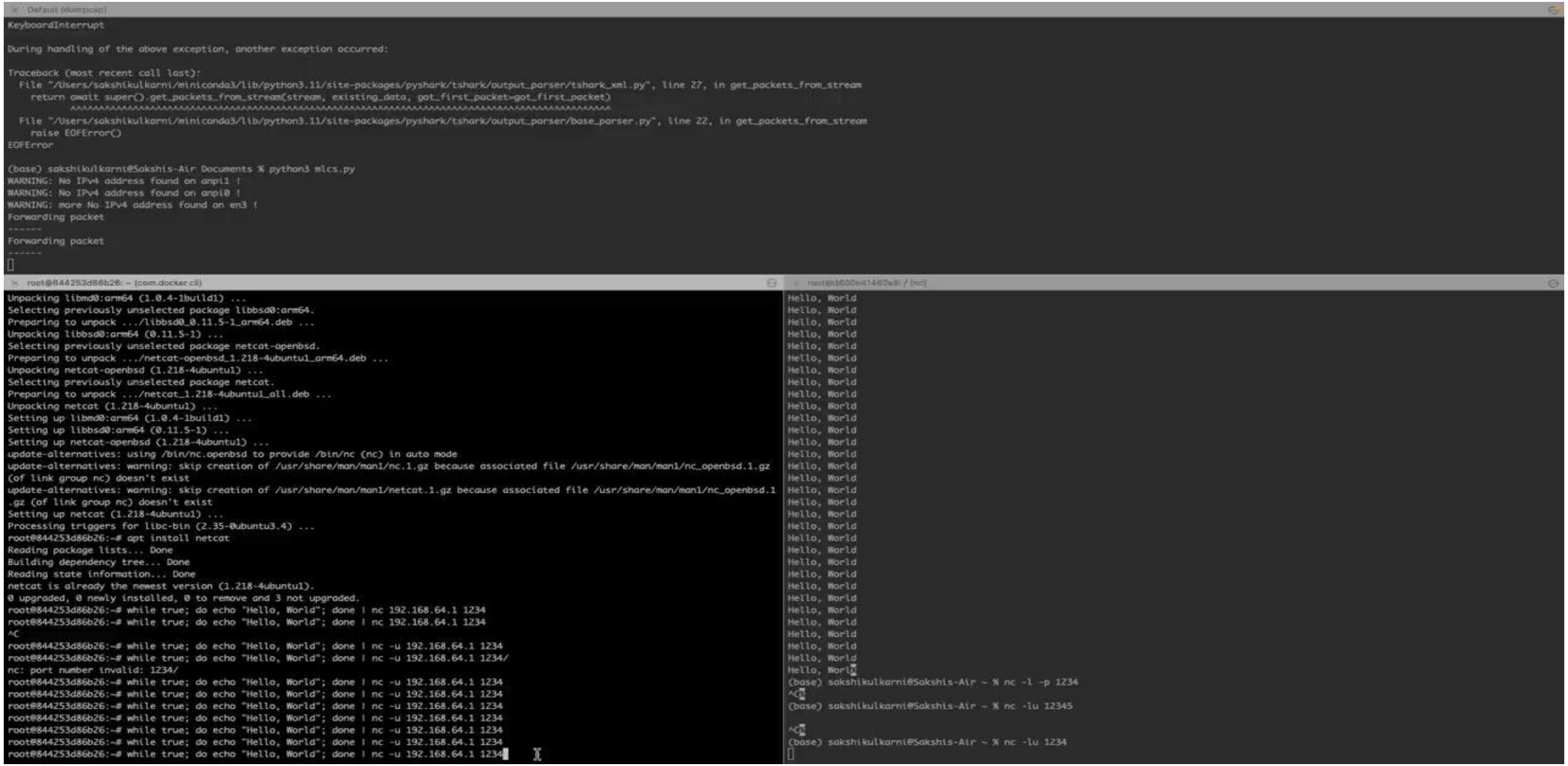
Hypothesis: Auto-learning Stateful Bridge

Our Custom Bridge can:

- Learn packet details using ML.
- Maintain state of network pack
- Detect malicious DDoS packet
- Offer a dynamic response.



Evaluation & Results



Default (dumpcap)
root@844253d86b26: ~ (com.docker.cli)
root@d600e41460e8: / (nc)

```
KeyboardInterrupt

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/Users/sakshikulkarni/miniconda3/lib/python3.11/site-packages/pyshark/tshark/output_parser/tshark_xml.py", line 27, in get_packets_from_stream
    return await super().get_packets_from_stream(stream, existing_data, got_first_packet=got_first_packet)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/Users/sakshikulkarni/miniconda3/lib/python3.11/site-packages/pyshark/tshark/output_parser/base_parser.py", line 22, in get_packets_from_stream
    raise EOFError()
EOFError

(base) sakshikulkarni@Sakshis-Air Documents % python3 mlcs.py
WARNING: No IPv4 address found on anpi1 !
WARNING: No IPv4 address found on anpi0 !
WARNING: more No IPv4 address found on en3 !
Forwarding packet
------
Forwarding packet
------
▯
```

```
Unpacking libmd0:arm64 (1.0.4-1build1) ...
Selecting previously unselected package libbsd0:arm64.
Preparing to unpack .../libbsd0_0.11.5-1_arm64.deb ...
Unpacking libbsd0:arm64 (0.11.5-1) ...
Selecting previously unselected package netcat-openbsd.
Preparing to unpack .../netcat-openbsd_1.218-4ubuntu1_arm64.deb ...
Unpacking netcat-openbsd (1.218-4ubuntu1) ...
Selecting previously unselected package netcat.
Preparing to unpack .../netcat_1.218-4ubuntu1_all.deb ...
Unpacking netcat (1.218-4ubuntu1) ...
Setting up libmd0:arm64 (1.0.4-1build1) ...
Setting up libbsd0:arm64 (0.11.5-1) ...
Setting up netcat-openbsd (1.218-4ubuntu1) ...
update-alternatives: using /bin/nc.openbsd to provide /bin/nc (nc) in auto mode
update-alternatives: warning: skip creation of /usr/share/man/man1/nc.1.gz because associated file /usr/share/man/man1/nc_openbsd.1.gz
(of link group nc) doesn't exist
update-alternatives: warning: skip creation of /usr/share/man/man1/netcat.1.gz because associated file /usr/share/man/man1/nc_openbsd.1
.gz (of link group nc) doesn't exist
Setting up netcat (1.218-4ubuntu1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.4) ...
root@844253d86b26:~# apt install netcat
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
netcat is already the newest version (1.218-4ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
root@844253d86b26:~# while true; do echo "Hello, World"; done | nc 192.168.64.1 1234
root@844253d86b26:~# while true; do echo "Hello, World"; done | nc 192.168.64.1 1234
^C
root@844253d86b26:~# while true; do echo "Hello, World"; done | nc -u 192.168.64.1 1234
root@844253d86b26:~# while true; do echo "Hello, World"; done | nc -u 192.168.64.1 1234/
nc: port number invalid: 1234/
root@844253d86b26:~# while true; do echo "Hello, World"; done | nc -u 192.168.64.1 1234
root@844253d86b26:~# while true; do echo "Hello, World"; done | nc -u 192.168.64.1 1234
root@844253d86b26:~# while true; do echo "Hello, World"; done | nc -u 192.168.64.1 1234
root@844253d86b26:~# while true; do echo "Hello, World"; done | nc -u 192.168.64.1 1234
root@844253d86b26:~# while true; do echo "Hello, World"; done | nc -u 192.168.64.1 1234
root@844253d86b26:~# while true; do echo "Hello, World"; done | nc -u 192.168.64.1 1234
root@844253d86b26:~# while true; do echo "Hello, World"; done | nc -u 192.168.64.1 1234
```

```
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, Worl
(base) sakshikulkarni@Sakshis-Air ~ % nc -l -p 1234
^C
(base) sakshikulkarni@Sakshis-Air ~ % nc -lu 12345

^C
(base) sakshikulkarni@Sakshis-Air ~ % nc -lu 1234
▯
```

Evaluation & Results

Type	Latency increase
Regular traffic on the Bridge	28.94%
DDoS attack on Bridge	3538.78%

Although these latency figures are notable, implementing such a bridge can be beneficial for system protection during potential DDoS attacks.

Limitations & Future Scope

- Integrating the bridge into Dockers.
- Adding support for TCP Reset etc.
- python™ → C