VISUALIZING SECURITY BOUNDARIES IN DOCKER SWARM OVERLAY NETWORKS

Marcel Brouwers
July 3, 2017

Master of System and Network Engineering
University of Amsterdam
Supervisor: Esan Wit
Docker Swarm

- Mode for managing a cluster of docker nodes
- The Swarm keeps services running and distributes containers over the nodes
- Has a feature for overlay networks between containers
Docker Swarm overlay network

- VxLAN \(^1\) based overlay networks. (Layer 2 over Layer 3)
- Containers can be connected to multiple Swarm overlay networks
- Networks are created from the manager nodes
- Serf used for mapping \(^2\)

\(^1\)https://tools.ietf.org/html/rfc7348
\(^2\)https://github.com/docker/libnetwork/blob/master/drivers/overlay/ov_serf.go
- RFC 7348
- Layer 2 over layer 3
- 24 bits Virtual Network Identified (VNI)
- UDP port 4789
Research Question

- What gets exposed when using Docker Swarm overlay networks and is there a way to visualize what gets exposed?
What gets exposed when using Docker Swarm overlay networks and is there a way to visualize what gets exposed?

Which security measures are there for Docker Swarm overlay networks and what can be done on the overlay network if a container or host gets compromised?
Research Question

- What gets exposed when using Docker Swarm overlay networks and is there a way to visualize what gets exposed?
  - Which security measures are there for Docker Swarm overlay networks and what can be done on the overlay network if a container or host gets compromised?
  - Which strategies are there to find out what gets exposed by containers and hosts in (overlay) networks?
What gets exposed when using Docker Swarm overlay networks and is there a way to visualize what gets exposed?

- Which security measures are there for Docker Swarm overlay networks and what can be done on the overlay network if a container or host gets compromised?
- Which strategies are there to find out what gets exposed by containers and hosts in (overlay) networks?
- Is it feasible to consolidate all the information about exposure and visualize it in a comprehensible way?
Related work

- Layer 2 attacks on a VxLAN overlay network, Author: G. Peneda, March 11, 2014
- Secure Virtual Network Configuration for Virtual Machine (VM) Protection Author: NIST, March 2016
- Docker swarm mode overlay network security model Author: Docker Project, 2017

https://docs.docker.com/engine/userguide/networking/overlay-security-model/
Security measures for Swarm overlays

- Encryption possible: IPSEC tunnel
- Encryption for overlay network not used by default
· Tested: ARP spoofing, MAC flooding
  · Tested using: Arpspoof tool (Dsniff), Ettercap, Macof (Dsniff)
  · Using non-privileged containers and privileged containers
  · Monitored ARP tables and sniffed network traffic
What’s possible?

- Tested: ARP spoofing, MAC flooding
  - Tested using: Arpspoof tool (Dsniff), Ettercap, Macof (Dsniff)
  - Using non-privileged containers and privileged containers
  - Monitored ARP tables and sniffed network traffic
  - Result: Not possible.
Why was that not possible?

Listing 1: Proxy ARP configured on VTEP

```
root@manager1:~# ip netns exec 1-7x3gglxlba ip -d link show vxlan1
11: vxlan1: <BROADCAST, MULTICAST, UP, LOWER_UP> mtu 1450 qdisc noqueue master br0 state UNKNOWN mode DEFAULT group default
   link/ether 46: e6: 48: 5d: dd: 92 brd ff: ff: ff: ff: ff: ff link—netnsid 0 promiscuity 1
   vxlax id 4097 srcport 0 0 dstport 4789 proxy l2miss l3miss ageing 300
```

“In addition to a learning-based control plane, there are other schemes possible for the distribution of the VTEP IP to VM MAC mapping information” 4

FDB gets populated using a gossip protocol “Serf”.

4https://tools.ietf.org/html/rfc7348#page-21
What’s possible?

- Tested: Replay of packets
  - Using Tcpreplay
  - ICMP from container A to container B on host A and B
  - Replayed ICMP request from node C
What’s possible?

- Tested: Replay of packets
  - Using Tcpreplay
  - ICMP from container A to container B on host A and B
  - Replayed ICMP request from node C
  - Works, ICMP reply arrives at container A

Also works when source ip is changed
Replay also works for an encrypted Swarm overlay network
VNIs predictable: start at 4096
UDP port 4789 (and tcp/udp 7946 for Serf)
What’s possible?

- Tested: Replay of packets
  - Using Tcpreplay
  - ICMP from container A to container B on host A and B
  - Replayed ICMP request from node C
  - Works, ICMP reply arrives at container A
  - Also works when source ip is changed
What’s possible?

- Tested: Replay of packets
  - Using Tcpreplay
  - ICMP from container A to container B on host A and B
  - Replayed ICMP request from node C
  - Works, ICMP reply arrives at container A
  - Also works when source ip is changed
  - Replay also works for an encrypted Swarm overlay network
What’s possible?

- Tested: Replay of packets
  - Using Tcpreplay
  - ICMP from container A to container B on host A and B
  - Replayed ICMP request from node C
  - Works, ICMP reply arrives at container A
  - Also works when source ip is changed
  - Replay also works for an encrypted Swarm overlay network

- VNIs predictable: start at 4096
- UDP port 4789 (and tcp/udp 7946 for Serf)
Strategies for finding out what gets exposed

- Have each container report netstat output and firewall status
  - Pro: Can be fast and complete
  - Con: Overhead by running on each container
  - Con: Required adapting docker files and redeploying.

- Scan the network
  - Pro: One container that runs a scanner
  - Con: Should be connected to all overlay networks
  - Con: Scan can take a long time
Strategies for finding out what gets exposed

· Have each container report netstat output and firewall status
  · Pro: Can be fast and complete
  · Con: Overhead by running on each container
  · Con: Required adapting docker files and redeploying.

· Scan the network
  · Pro: One container that runs a scanner
  · Con: Should be connected to all overlay networks
  · Con: Scan can take a long time
Strategies for finding out what gets exposed

- Have each container report netstat output and firewall status
  - Pro: Can be fast and complete
  - Con: Overhead by running on each container
  - Con: Required adapting docker files and redeploying.

- Scan the network
  - Pro: One container that runs a scanner
  - Con: Should be connected to all overlay networks
  - Con: Scan can take a long time

- Have each host report netstat output and firewall status for the containers
  - Pro: Containers can not be overlooked
  - Pro: Can be relatively fast
Visualizing

- D3.js
- Visualizations in the browser
- Collected data using Swarm API and scripts on hosts
Visualizing
Visualizing
Visualizing
Demo
CONCLUSION

- Layer 2 attacks based on ARP injecting seems not possible on a Swarm overlay network
- It is possible to inject something in a Swarm overlay network when standard configuration is used
- Encrypted Swarm overlay traffic can be successfully replayed
- Creating visualizations of the Swarm overlay networks taking security boundaries into account is possible
Future work

- Research the mechanism that updates the mapping for the VTEPs
- Work on visualizations for single nodes showing more detail for firewall configuration
Questions?