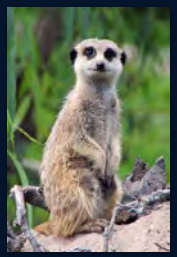
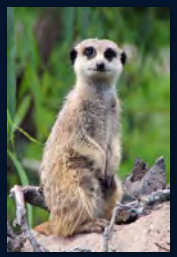


Virtual Meerkats

A brief, entry level expedition through setting up Suricata within a virtualized environment. Will cover planning, implementation, common problems, tuning, and tips.

Presented by Jeremy MountainJohnson

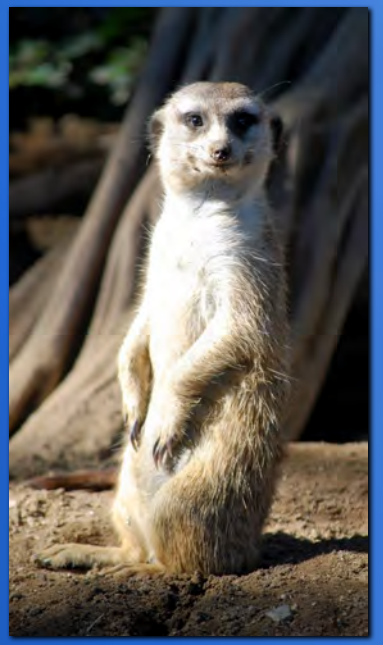


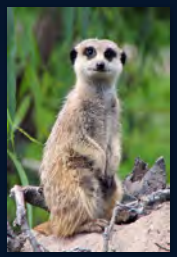
Greetings!

It is good to meet everyone, albeit virtually!

A little about me:

- Jeremy MountainJohnson
- Minneapolis, Minnesota, US
- Computer Forensics BaS, Computer Networking AAS
- Former Digital Forensics / Incident Response
- Former Adjunct Professor, UNIX
- Currently Security Analyst

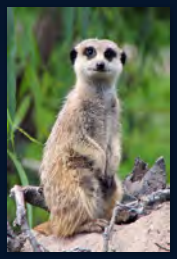




Our Agenda

Introduction to suricata sensors in a virtual environment

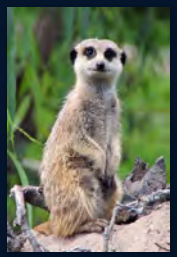
- Recommended settings in a virtual environment
- Operating systems
- Planning your setup
- Common virtual methods of getting traffic flow to your sensors
- Future scaling considerations
- Drivers, suricata running mode
- Network segmentation
- Suricata bpf, pass rules
- Typical problems, tuning
- Overview of virtualized sensors in production setups
- Discussion and Q&A



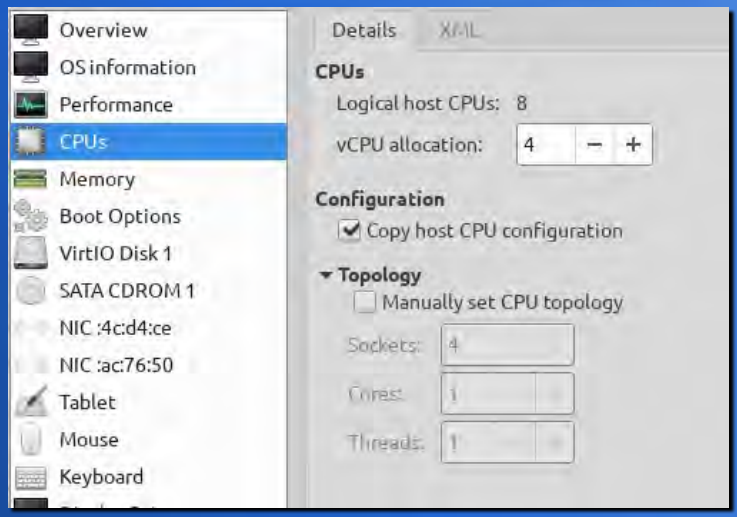
Before we get started...

- All information presented is my opinion and personal recommendations only, not those of my employer, OISF, or anyone else
- Due to variations hypervisor hardware/software, OS versions highly recommend testing in similar lab environment
- *NIX based operating systems

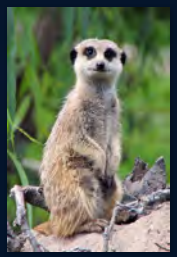
Suricon Code of Conduct located on Discord



Recommend Settings

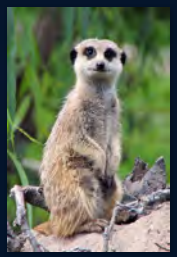


- ***Dedicated*** resource allocation
  - vCPU
  - vRAM
  - vNIC adequate speed
  - Storage pool I/O
- GPU acceleration not needed; CUDA support deprecated?
- More rules + traffic, more v-resources
- Storage space needs varies based on sensor setup
  - Saving pcaps
  - Eve logs, redis



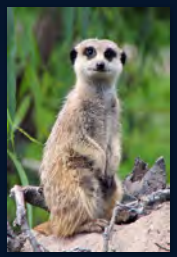
Operating Systems

- Base VM image for sensors
- Kernel modules for vnics
- Maintained port / package available for:
  - open-vm-tools, vbox
  - tcpdump or tshark, pcap
  - sshd
- Gotchas with distros
  - Suricata version dependencies
- Consider running sensors containerized
  - More upfront work, less maintenance long-term
  - Easier to use your preferred distro, less OS dependency problems



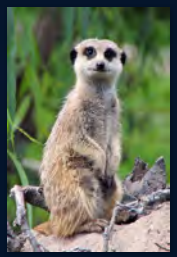
Planning & Scaling

- IPS or IDS now or future?
- Environment: all virtual, physical, or both?
- Internal, External traffic
  - Tap placements, span points
  - Hairpinning
  - vlans
- Traffic volume, sensors
  - 10/100/1000/10000+
  - How many sensors
    - By physical location, net segment, devices



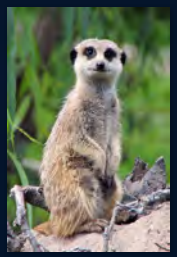
Getting Traffic In

- Getting traffic in
  - Virtual Switches, routers
  - Span
  - Tap
  - NIC device pass-thru (PCIE, USB)
- Testing receipt
  - Tcpdump, tshark, pcap
- Monitoring for outages



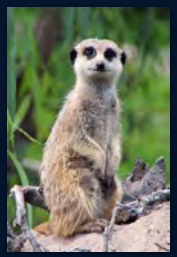
Drivers, run mode

- Kernel drivers for device pass-thru
- vNIC drivers
  - vmx#, virtio
- Emulated Intel, BroadCom, Realtek...
  - Improving
- Default run modes, single or workers
- Hyperscan has worked well in virtual environment



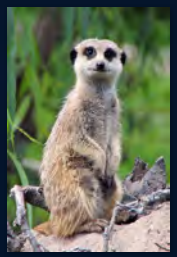
Network Segmentation

- Like physical, important to organize sensors
  - Physical site
  - vlan
  - Subnet
- Back of mind growth and scalability



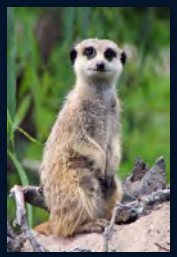
Suricata Rules and BPF

- Rules included in default rule set
  - erspan
  - General traffic rules
- Craft BPF syntax to environment
  - Added efficiency
  - Avoid switch loops
- Pass rules
- Custom rules, vlan



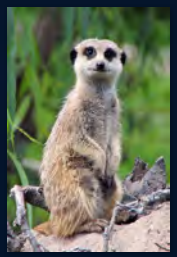
Packet Capture

- Erspan, rpsan, port mirroring
  - Suricata has erspan support
  - HP ERM, may need third party tool
    - rcdcap
      https://sourceforge.net/projects/rcdcap/
- Packet capture
  - AF-Packet
  - PF_RING



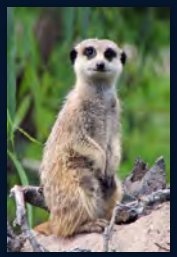
Typical Problems

- Non-dedicated resources latency
- Under-speced resources
- OS vNIC drivers, old or not using
- Improper MTU, jumbo frames
- Hypervisor
  - Proper CPU extensions
  - vNIC support
  - Bugs



Overview of Production

- Tested base sensor image deployed across multiple sites
- rspan with vlan tagging for internal traffic
- External traffic sensors downstream from firewall
- Sensors updated and managed by scripts, Ansible
- Alerting and monitoring with SIEM

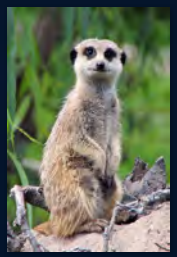


Credit for Digital Art Work

Images were used for non-commercial purposes, in accordance with their license, and not modified. Most licenses are under Creative Commons.

Original sources or creators credited below:

- Meerkat - https://www.flickr.com/photos/ianvoyce/
- Meerkats - https://www.flickr.com/photos/ullisandersson/
- Smiling Meerkat - https://www.flickr.com/photos/mackenzieandjohn/
- Beastie logo - https://upload.wikimedia.org/wikipedia/en/5/55/Bsd_daemon.jpg
- Tux logo - https://upload.wikimedia.org/wikipedia/commons/thumb/5/55/Tux_Enhanced.svg/878px-Tux_Enhanced.svg.png
- Looking Meerkat - https://www.flickr.com/photos/hollyoak-smith/
- Containers - https://www.flickr.com/photos/volvob12b/
- Phone switch - https://www.flickr.com/photos/glenbledsoe/
- Network switch - https://www.flickr.com/photos/christiaancolen/



Discussion & Q&A

Questions? Comments? Discussions? We're here!

Thank you for attending Suricon 2021, and also my talk!

You can find me:

- Twitter: @JSkier21
- Suricata forums: @JSkier
- Discord Suricata: Jeremy MountainJohnson