Using Tetration for application security and policy enforcement in multi-vendor environments.

Joel W. King  Engineering and Innovations  Network Solutions
Abstract

Using Tetration for application security and policy enforcement in multi-vendor environments.

Network engineers increasingly must view the network as one big software system, which streams telemetry data from software sensors and network devices to an analytics engine.

To implement the whitelist-based segmentation and zero-trust policy model generated from the data analysis, automation is a requirement when dealing with tens of thousands of workloads and complex rules.

This session examines how Cisco Tetration Analytics combined with automation can be used to implement a zero-trust policy model on multi-vendor network fabrics, firewalls and application delivery controllers.
$ whoami

- **Joel W. King**
  Principal Architect
  World Wide Technology
  Research Triangle Park, NC

- **Experience**
  AMP Incorporated, Network Architect
  Cisco, Cisco Validated Designs (CVDs)
  NetApp, Big Data: Video Surveillance Storage

- **Contact Info**
  [linkedin.com/in/programmablenetworks](https://linkedin.com/in/programmablenetworks)

  @joelwking

  joel.king@wwt.com
Very topical for us -- talk on implementing Zero Trust with automation and Tetration …

…Personally, I think ZT will replace perimeter security model within 5-7 years, and already we're hearing customers ask about it. …

-----------------------------------------------
Gene Geddes | Chief Scientist, Security Solutions | World Wide Technology
Why Analytics?
Automation
Deploy Sensors
Inventory
Tetration Network Policy Publisher
Under the Hood
Resources

#SiliconValleyinSTL
Analytics across the security landscape

- **SHARE**: Can you collaborate with trusted partners to disrupt adversary campaigns?
- **ACT**: Can you deploy proven countermeasures to evict and recover?
- **TRACE**: During an intrusion, can you observe adversary activity in real time?
- **HUNT**: Can you detect an adversary that is already embedded?
- **BEHAVIORS**: Can you detect adversary activity within your environment?
- **THREATS**: Who are your adversaries? What are their capabilities?
- **TRIAGE**: Can you accurately classify detection results?
- **DETECTION**: Can you detect unauthorized activity?
- **TELEMETRY**: Do you have visibility across your assets?
- **INVENTORY**: Can you name the assets you are defending?

- **What’s on my network?**
- **Should it?**

© World Wide Technology
Automated whitelist policy
Zero-trust, application segmentation

Cisco Tetration Analytics
Illumio
VMware vRNI
telemetry agent installation

inventory

policy

enforcement iptables | firewall

publisher kafka

NETWORK DEVICES
Data Collection Layer

CONTAINER HOST SENSOR
SOFTWARE SENSOR [AND ENFORCEMENT]
EMBEDDED HARDWARE SENSORS
ERSPAN SENSORS
NETFLOW V9/ IPFIX SENSORS
NETWORKING [TELEMETRY ONLY]

Data Consumption Layer

WEB GUI
REST API [DYNAMIC INVENTORY]
KAFKA MESSAGE BUS [NETWORK POLICY PUBLISHER]
CISCO TETRATION APPS

Cisco Tetration Analytics™
Automation #SiliconValleyInSTL
Use Cases

...yes there is code ...

https://github.com/joelwking/ansible-tetration

- Deploy Software Sensors
  setup_tetration_sensor.yml

- Dynamic Inventory
  inventory/sensors.py

- Network Policy Publisher
  library/tetration_network_policy.py
Automation

Deploy Sensors

#SiliconValleyinSTL
Data Collection Layer

SOFTWARE SENSOR
[AND ENFORCEMENT]

ERSPAN SENSORS

NETFLOW V9/ IPFIX SENSORS

Cisco Tetration Analytics™

39-RU 8-RU SaaS

or virtual appliance

25,000 | 5,000 | 1,000

NETWORK INFRASTRUCTURE

NetFlow | ERSPAN VM Appliance

© World Wide Technology
Sensor Agent Install - Pain Points

- Extensive matrix of Windows | Unix | Linux
- Package and version dependencies e.g. rpm (even in Ubuntu/Debian)
- Different agent RPMs for ...
  - Agent type, e.g. enforcement, visibility
  - Target system, e.g. CentOS 6.0 vs 7.0
  - Latest version covers 34 RPMs
- Agent downloaded from GUI
Re-think Documentation

- **Rather than PDF …**

- **./setup_tetration_sensor.yml**

```bash
[administrator@centos-ansible-1 ~]$ uname
Linux
[administrator@centos-ansible-1 ~]$ -r
-bash: -r: command not found
[administrator@centos-ansible-1 ~]$ uname -r
3.10.0-862.el7.x86_64
```

---

```bash
type:$

```

unname -r value: 3.10.0-862.el7.x86_64
cat /etc/shells value: /bin/sh
dmidecode -V value: 3.0
openssl version -a value: OpenSSL 1.0.2k-fips
cpio --version value: cpio (GNU cpio) 2.11
sed --version value: sed (GNU sed) 4.2.2
awk --version value: GNU Awk 4.0.2
flock -V value: flock from util-linux 2.23.2
iptables --version value: iptables v1.4.21
ipset --version value: ipset v6.29,
```

ansible-tetration/setup_tetration_sensor.yml
ansible-tetration/inventory/sensors.py
ansible-inventory --host centos-ansible-1 -i ./inventory/sensors.py

```python
{
    "agent_type": "ENFORCER",
    "auto_upgrade_opt_out": false,
    "cpu_quota_mode": 1,
    "cpu_quota_usec": 30000,
    "current_sw_version": "2.3.1.41-1-enforcer",
    "data_plane_disabled": false,
    "enable_forensics": false,
    "enable_pid_lookup": false,
    "host_name": "centos-ansible-1",
    "interfaces": [
        {
            "family_type": "IPV4",
            "ip": "10.255.40.139",
            "mac": "00:50:56:b9:62:58",
            "name": "ens160",
            "netmask": "255.255.255.0",
            "vrf": "Default",
            "vrf_id": 1
        },
        [snip]
    ],
    "last_config_fetch_at": 1537905092,
    "last_software_update_at": 1535054507,
    "platform": "CentOS-7.5",
    "uuid": "965e77504bf605d62c575231fa3d56463aed38bf"
}
```
Automation

Tetration Network Policy Publisher

#SiliconValleyInSTL
policy

enforcement
iptables | firewall

publisher
kafka

NETWORK DEVICES

INFRASTRUCTURE
Enforcement and Publishing Policy

BROKER

ADD TENANT
ADD VRF
AGENT CONFIG
ENABLE ENFORCEMENT
CREATE INTENT
ADD SCOPE
CREATE APP
START ADM RUN
ENABLE ENFORCEMENT
VERIFY DATATAP CREATION
DOWNLOAD CERTIFICATES

NETWORK PROGRAMMABILITY
DEVELOPER

ADM ANALYST

Tnp-12 10.253.239.14:9093

 certificates
 ├── producer-tnp-12.cert/
 │     └── kafkaBrokerIps.txt
 │     └── KafkaCA.cert
 │     └── KafkaConsumerCA.cert
 │     └── KafkaConsumerPrivateKey.key
 └── topic.txt

© World Wide Technology
Network Policy Publisher

- **message publisher**: kafka
- **policy subscription**: BROKER

**Modules**
- tetratlon_network_policy.py

**Ansible Playbook**
- aci_create_filters.yml

Alerts every minute for enforcement

Released in 2.3.1.41 April 2018
- name: Tetration Network Policy
  tetration_network_policy:
  broker: "192.0.2.1:9093"
  topic: "Tnp-2"
  cert_directory: "{{ playbook_dir }}/files/certificates/producer-tnp-2.cert/"

https://github.com/joelwking/ansible-tetration/blob/master/aci_create_filters.yml
Why these tools?

- Protocol Buffers are a way of encoding structured data in an efficient yet extensible format.
- Google open source and supported for popular programming languages.
- Fast and efficient (than JSON or XML).

- … designed to deal with millions of firehose-style events generated in rapid succession…
- … clients will never receive messages automatically. They have to explicitly ask for a message ...

https://codeclimate.com/blog/choose-protocol-buffers/
https://thenewstack.io/apache-kafka-primer/
Architecture

TETRATION

KAFKA

producer

key : value
key : value

messages

key : value
key : value

partition

topic = "Tnp-12", partition = 0

consumer

-.proto

USER CODE

certs

NETWORK DEVICES

Application dependency mapping [with enforcement]
Tetration Network Policy

Kafka message(s)
- topic
- partition
- offset
- key
- value

Google Protocol Buffer

UPDATE
- len( value ) == 8
- UPDATE_START
- UPDATE_END

EARLIEST — Tetration Network Policy — LATEST

© World Wide Technology
Google Protocol Buffer Basics: Python

- Also know as: “GPB” or “protobufs”
- What are they?
  - Method of serializing structured data
  - XML | JSON uses strings to identify the key
  - Protobufs uses integers to represent the key
  - Sender and receiver share a .proto definition file
- Why Use Protocol Buffers?
  - Performance: Smaller and faster than XML
  - More compact (smaller packets, messages)
  - Faster, less CPU to encode / decode

For Reference

https://developers.google.com/protocol-buffers/docs/pythontutorial
Resources

- **AnsibleFest 2018**: Using Ansible Tower to implement security policies and telemetry streaming for hybrid clouds
  https://github.com/joelwking/ansible-tetration

- **DevNetCreate 2018**: Applying a whitelist policy generated by Cisco Tetration to an ACI network fabric.
  https://www.wwt.com/all-blog/devnet-create-2018/

- **Cisco Tetration Light-board**: Cloud Workload Protection
  https://youtu.be/Hd56GVVr_AE

- **Cisco Code Exchange**
  https://developer.cisco.com/codeexchange/#search=tetration
… turning the whole network into essentially a big software system where you define your policy in one place …

That policy gets translated into what you want the network to do, and then you have an automation layer that activates all of those changes across your network fabric.

David Goeckler, EVP / GM of Cisco's Networking and Security
Key Points

- Traditional Firewalls as perimeter security are becoming obsolete
- Future is white-list segmentation, Zero Trust model
- View the network as a software system, use automation to apply policy
