AUTONOMOUS SECURITY RESPONSE ORCHESTRATION FOR PROGRAMMABLE NETWORKS

Vibha R Goutham (MSc)

Dr. Piotr Zuraniewski
Ir. Frank Fransen

TNO innovation for life

OUTLINE

- Master Thesis proposal – background
- Use Cases
- TNO Research Cloud
- Research questions
- Implementation – methods
- Preliminary results

MASTER THESIS PROPOSAL

Background:

- Nature & complexity of cyber attacks are growing and tackling them beyond human capabilities is essential
- This necessitates design of an automated security function that can be orchestrated

- Today's networks focus on accelerated deployment of new network functions. Also the aim is to reduce hardware constraints greatly
- This leads to imploring principles of *Network Function Virtualization* (NFV)

- The research is a combination of Automated Security function + NFV to deal with cyber attacks

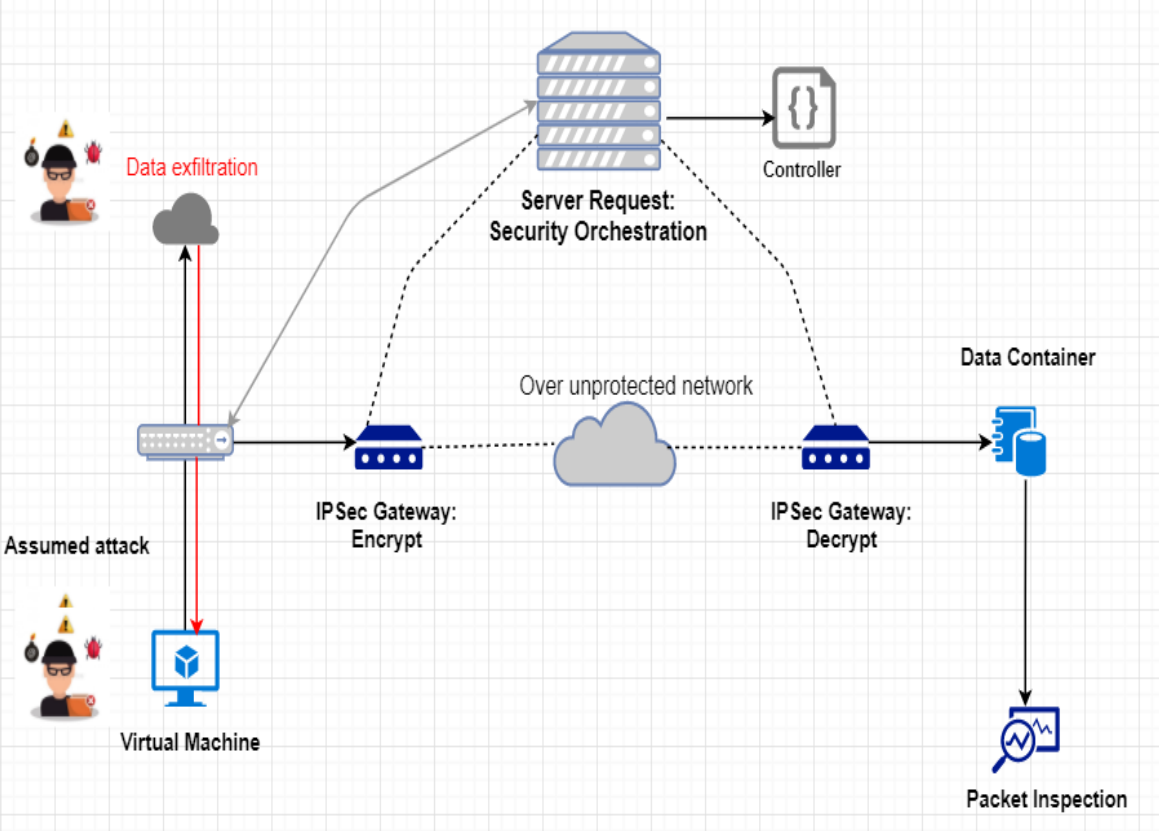
USE CASE I

Assumption: Data exfiltration attack

- Clone/divert suspicious traffic over dynamically established secure gateways (IPSec/GRE tunnel)

- Packets stored in container

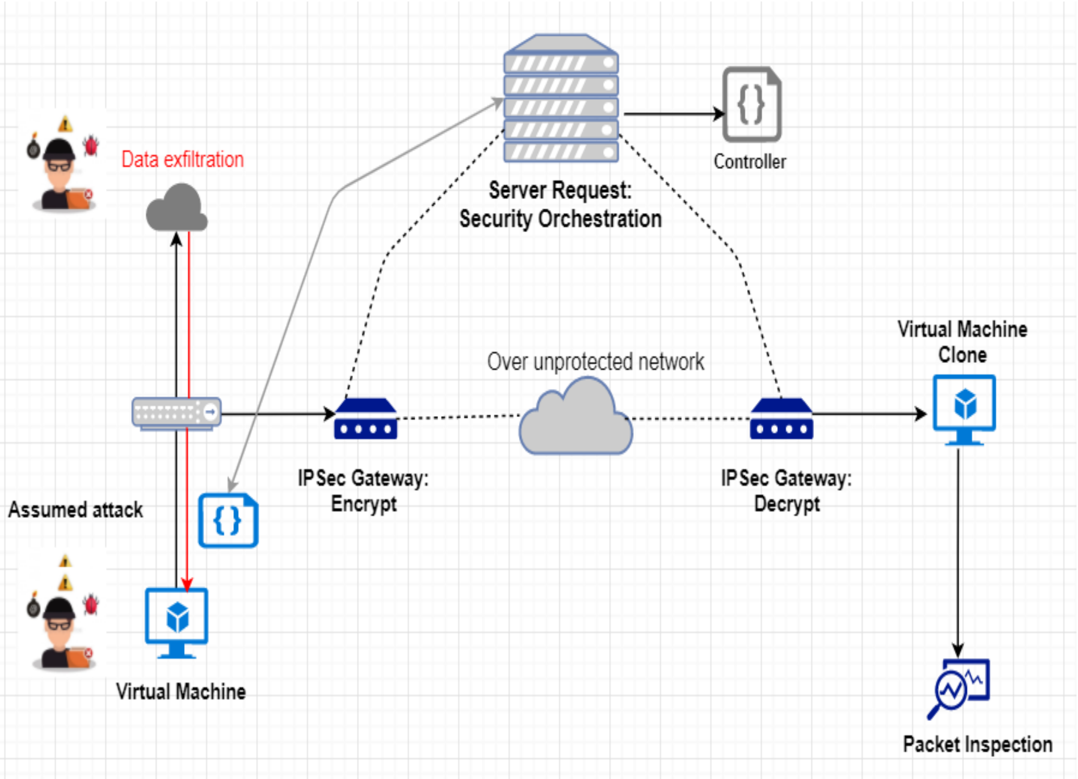
Security monitoring: Packet inspection, security logging and analysis capability



USE CASE II

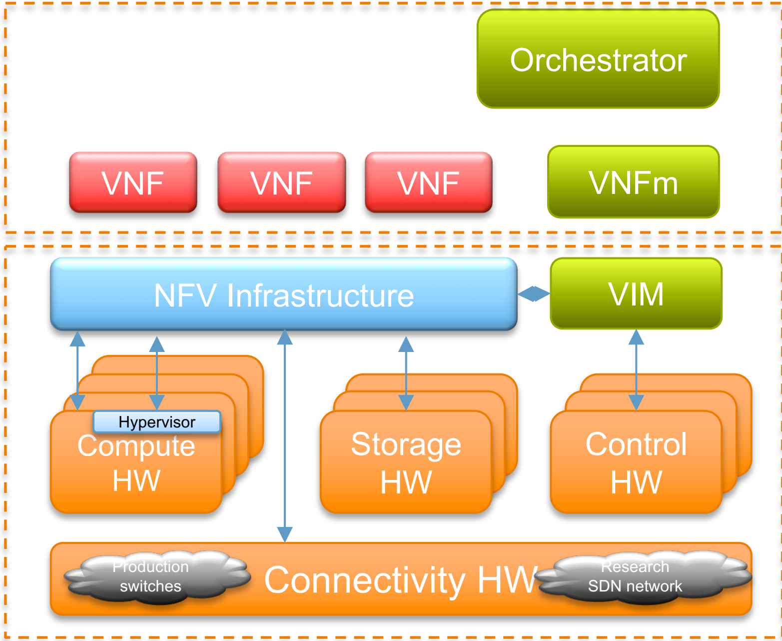
Assumption: Exfiltration attack

- Cloning VM and its evacuation over public network
- Re-route suspicious traffic over dynamically established secure gateways (IPSec/GRE tunnel)
- Security monitoring: Packet inspection, security logging and analysis capability environment for cloned VM



TNO RESEARCH CLOUD

- Platform available in-house:
  - OpenStack/Ceph private cloud infrastructure
  - 10 physical servers (high-availability design)
  - 2 top-of-the-rack switches
  - Programmable NICs, several SDN switches
- Used for prototyping and experimentation
  - Management and orchestration (MANO)
  - 5G
  - Post-quantum crypto, Blockchain, ICN,…



RESEARCH QUESTIONS

How will the deployment of the security function be done via an Orchestrator?

Is it possible to initiate and re-configure the security function via Orchestrator? If yes, how will it be done?

Which security function method is better in terms of ease of set-up, robustness, performance & scalability?

The response window for the security function and its optimization.

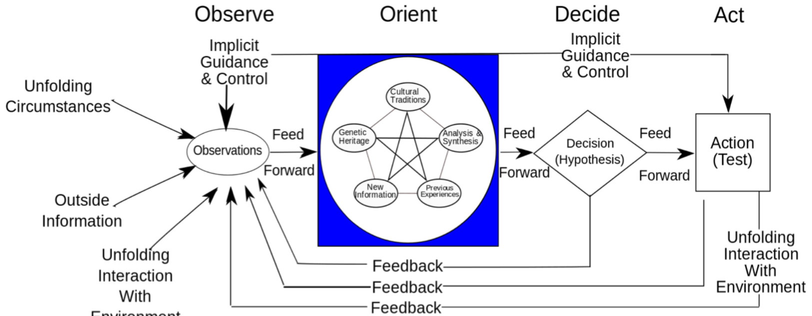
IMPLEMENTATION

<u>Background - OODA Control loop</u>

› The OODA loop decision cycle of

Observe, *Orient*, *Decide*, and *Act*,

developed by military strategist and United States Air Force Colonel John Boyd



John Boyd's OODA Loop

Proposed methods for Secure transfer

<u>Method 1:</u>

Experiment on building an encapsulated IPSec-GRE tunnel over Open vSwitch

- On an unprotected public network, suspicious traffic or an instance of a VM may be transported over the encapsulated tunnel.
- IPSec encryption with AES-SHA shall ensure the secure transfer.

<u>Method 2:</u>

Employ hardware-accelerated data plane function on a smart NIC.

- Smart NICs perform role of secure gateways for transfer over unprotected public network.
- IPSec encryption shall be tested on vendor-specific NICs (ex: hardware accelerated OVS, P4 enabled)

PRELIMINARY RESULTS

Baseline IPSEC without NIC acceleration

- VMs hosted on same Hypervisor environment
- Further Investigation: Available bandwidth for IPSec-GRE tunnel mode

```
ubuntu@gre2:~$ iperf3 -c 192.19.0.1
Connecting to host 192.19.0.1, port 5201
[  4] local 192.19.0.2 port 41298 connected to 192.19.0.1 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec  62.5 MBytes   524 Mbits/sec    0   1.10 MBytes
[  4]   1.00-2.00   sec  67.9 MBytes   570 Mbits/sec    0   1.23 MBytes
[  4]   2.00-3.00   sec  67.4 MBytes   565 Mbits/sec    0   1.30 MBytes
[  4]   3.00-4.00   sec  67.4 MBytes   565 Mbits/sec    0   1.39 MBytes
[  4]   4.00-5.00   sec  66.9 MBytes   561 Mbits/sec    0   1.39 MBytes
[  4]   5.00-6.00   sec  64.9 MBytes   544 Mbits/sec    0   1.39 MBytes
[  4]   6.00-7.00   sec  67.8 MBytes   569 Mbits/sec    0   1.39 MBytes
[  4]   7.00-8.00   sec  61.9 MBytes   519 Mbits/sec    0   1.39 MBytes
[  4]   8.00-9.00   sec  69.7 MBytes   585 Mbits/sec    0   1.65 MBytes
[  4]   9.00-10.00  sec  69.2 MBytes   581 Mbits/sec    0   1.65 MBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec   666 MBytes   558 Mbits/sec    0             sender
[  4]   0.00-10.00  sec   662 MBytes   555 Mbits/sec                  receiver

iperf Done.
```

With GRE tunnelling

```
ubuntu@ipsec1:~$ iperf3 -c 10.100.1.10
Connecting to host 10.100.1.10, port 5201
[  4] local 10.100.1.13 port 42872 connected to 10.100.1.10 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec  1.45 GBytes  12.5 Gbits/sec    4   3.02 MBytes
[  4]   1.00-2.00   sec  1.77 GBytes  15.2 Gbits/sec    0   3.02 MBytes
[  4]   2.00-3.00   sec  1.78 GBytes  15.3 Gbits/sec    0   3.02 MBytes
[  4]   3.00-4.00   sec  1.49 GBytes  12.8 Gbits/sec    0   3.02 MBytes
[  4]   4.00-5.00   sec  1.75 GBytes  15.0 Gbits/sec    0   3.02 MBytes
[  4]   5.00-6.00   sec  1.76 GBytes  15.1 Gbits/sec    0   3.02 MBytes
[  4]   6.00-7.00   sec  1.75 GBytes  15.1 Gbits/sec    0   3.02 MBytes
[  4]   7.00-8.00   sec  1.63 GBytes  14.0 Gbits/sec    0   3.02 MBytes
[  4]   8.00-9.00   sec  1.71 GBytes  14.7 Gbits/sec    0   3.02 MBytes
[  4]   9.00-10.00  sec  1.58 GBytes  13.6 Gbits/sec    0   3.02 MBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec  16.7 GBytes  14.3 Gbits/sec    4             sender
[  4]   0.00-10.00  sec  16.7 GBytes  14.3 Gbits/sec                  receiver

iperf Done.
```

Without tunnelling

```
ubuntu@ipsec1:~$ iperf3 -c 192.19.0.12
Connecting to host 192.19.0.12, port 5201
[  4] local 192.19.0.11 port 54082 connected to 192.19.0.12 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec  67.3 MBytes   565 Mbits/sec    0   1.96 MBytes
[  4]   1.00-2.00   sec  76.2 MBytes   638 Mbits/sec    0   2.98 MBytes
[  4]   2.00-3.00   sec  72.5 MBytes   609 Mbits/sec    0   3.00 MBytes
[  4]   3.00-4.00   sec  73.8 MBytes   619 Mbits/sec    0   3.00 MBytes
[  4]   4.00-5.00   sec  70.0 MBytes   587 Mbits/sec    0   3.00 MBytes
[  4]   5.00-6.00   sec  72.5 MBytes   608 Mbits/sec    0   3.00 MBytes
[  4]   6.00-7.00   sec  73.8 MBytes   619 Mbits/sec    0   3.00 MBytes
[  4]   7.00-8.00   sec  72.5 MBytes   608 Mbits/sec    0   3.00 MBytes
[  4]   8.00-9.00   sec  73.8 MBytes   619 Mbits/sec    0   3.00 MBytes
[  4]   9.00-10.00  sec  73.8 MBytes   619 Mbits/sec    0   3.00 MBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec   726 MBytes   609 Mbits/sec    0             sender
[  4]   0.00-10.00  sec   723 MBytes   607 Mbits/sec                  receiver

iperf Done.
```

With IPSec-GRE


THANK YOU FOR YOUR ATTENTION
Take a look:
TIME.TNO.NL
TNO innovation for life