Collaborative Network Security in Multi-Tenant Data Center for Cloud Computing

Zhen Chen*‡, Wenyu Dong§, Hang Li§, Junwei Cao*‡, Peng Zhang § and Xinming Chen §
§ Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China
*Research Institute of Information Technology, Tsinghua University, Beijing, 100084, China
‡Tsinghua National Lab for Information Science and Technology, Beijing, 100084, China
{zhenchen, dwy13, jcao}@tsinghua.edu.cn, luckyyyewyz@gmail.com

Abstract-Data center is the infrastructure of supporting Internet service. Cloud computing is rapidly changing the face of Web Internet service infrastructure, enabling even small organizations to quickly create Web and mobile applications for millions of users by taking advantage of the scale and flexibility of the shared physical infrastructures provided by cloud providers. In this scenario, multiple tenants saved their data and applications in the same data centers making the network boundaries between each tenant become blurred. Also different tenants have different security requirements, it needs to create different security policies for them. Network virtualization is to "compile" a diverse set of tenant-specific requirements into a single configuration of the underlying physical cloud network, enabling multi-tenant datacenters to automatically address a large and diverse set of tenants' requirements. In this paper, we propose architecture, mechanism design and system implementation of vCNSMS, a collaborative network security prototype system in multiple tenant's data center network. We demonstrates vCNSMS with a centralized collaborative scheme and deep packet security check in peer-UTMs among vCNSMS with open source peer-UTM system. A security level based protection policy is proposed for simplifying the security rule management for vCNSMS. Different security level has different packet inspection scheme and enforced with different security plugins. A smart packet verdict scheme is also integrated into vCNSMS for intelligence flow processing to defense possible network attack inside data center network.

Keywords: data center network, network security, software defined network, collaborative network security, multi-tenant, network virtualization, intelligent flow processing.

1. Introduction

Cloud data center is the infrastructure of supporting Internet service. The definition of cloud data center can have a variety of perspective, the most popular one are categorized by IAAS, PAAS, SAAS proposed by the U.S. NIST [X] and public cloud, private cloud, hybrid cloud and some other different categories, but also computing, networking, storage of a system perspective, or using (in use), archiving (at rest), transmission (in motion) from a data perspective. Specific to the cloud network, there are different characteristics of cloud (terminal access), within a cloud, between clouds network. VMware NSX [23] is aimed at the virtualization of networks inside the cloud data center, consolidating its server virtualization and achieving the blueprint of a virtual software defined data center. In [29], Google B4 network also use OpenFlow based SDN [30-31] to implement all the interconnection between the cloud data center.

Starting from the actual demands of virtual private cloud of tenants, compared the differences between enterprise networks and virtual private clouds, it is easy to find that requirements of network security are still through flexible strategy tailored to realize inner network (trusted network or the internal network) and external network (not trusted network or public network) secure connections. The biggest challenge posed by SDN is dynamic characteristic of network boundaries which is flexibility provided by virtualization. In other words, the original static, natural physical boundaries within the network, is replaced by dynamic and virtual logical boundaries of SDN. So network security within the cloud data center will be more dependent on dynamic deployment, configuration and management of security policies and security components, and more dependent on the network security system for traffic awareness, flow management, decision-making and quick response.

1.1 State-of-art network security in data center network

Traditional network security devices such as Firewalls, IDS, WAF etc. are deployed as Middleboxes in-between the inside and outside networks. With the development of cloud computing technology, more and more enterprises are moved in to cloud as a tenant to take the advantage of scale and flexibility of cloud computing. The deployment of Middleboxes is

facing new challenges in the large-scale data center network environment.

(1) In multi-tenant cases, the network boundaries are blurring.

With the increase of tenants, the data center network topology along becomes complicated. Multiple tenants put their data in cloud and the same tenant may utilize different servers with multiple backups, making the network boundaries between each tenant become blurred, and virtualized rather than traditional physical isolation. The original static, natural physical boundaries within the network, is replaced by dynamic and virtual logical boundaries. So network security within the cloud will be more dependent on dynamic deployment, configuration and management of security policies and security components, and more dependent on the network security system for flow and traffic awareness, decision-making and response. Undoubtedly, this makes network security management become more complex. How to ensure network security is also a problem in such a complex network environment.

(2) The deployment of Middleboxes need repositioning.

In traditional enterprise networks, the traffic of hosts are protected from the same gateway, and the entire enterprise network may have several gateways. In the data center network, the physical gateway have been replaced by virtual logical gateways. In order to protect the security of virtual logical boundaries between tenants, it requires a Firewall, IDS/IPS and other devices to collaborate with the traffic controller, to adapt to performance and safety requirements of each tenant or security domain, security boundary dynamics caused by virtual machine migration, as well as the dynamic security requirements of virtual machines on the demand. Therefore, to meet these requirements under the premise of how to properly deploy these security devices is also an important issue.

(3) Security requirements for different tenants are different.

Since different tenants have different security requirements, it needs to create different security policies for these tenants, which is undoubtedly a big challenge for traditional security devices. The traditional approach is to set rules for flows passed through the device which can be filtered and monitored. Obviously, in the data center network, this approach cannot work anymore, and now we have to face the question that how to meet all the individual requirements for security and effectively enforce all the security policy when multiple tenants' network traffic pass through the same security device.

(4) The migration of virtual machines results in security domain changed.

Considered to meet the security needs of a single tenant, it need to configure the security rules within multiple security devices to control traffic and thus completely implement the tenant's security policy. When the server node needs to migrate to other locations in the network, the topology of the network node changes and appropriate security policies also migrate with the tenant data. Original security configuration on the security device will no

longer work, the migration to the new security domain also requires tenants on that server to do some configuration adjustments. Therefore, to protect the mapping of the security policies from the logical network to the physical network and its correctness and consistency will be a big challenge to the control of data center network.

1.2 Software-defined network in data center network

In the cloud data center networks, SDN becomes one of the supporting technologies to build cloud data center networks. Cloud data center networks need to ensure tenants or the security domains with complete and isolated network boundaries. This is not a simple network security technology, but the virtual network itself provides services for multiple tenants or basic virtual private cloud services which should be guaranteed in the design and implementation of the network. Of course, due to the different virtual machines of different tenants shared the same physical resources, system security guarantees such as preventing the virtual machine escaping are also the foundation of network security.

Compared with the traditional network, the core change of SDN is switches, routers and some other forwarding devices only forward according to flow table created by controller, thereby becoming more efficient and less costly. On the other hand, the controller collects network status information, discovers network topologies, checks the network forwarding policies, generates and updates the flow table.

Thus, according to the handling of the first packet's header of a flow in the controller, it should not be forwarded to controller again. So the traditional ACL (Access Control List) or packet filtering Firewall, it should be deployed in the controller. However, the controller usually only receives the first packet header and cannot do stateful inspection. So stateful Firewalls and Deep Inspection Firewalls which need to filter packet payload should be deployed in the data plane or in the intersection of data plane and control plane, such as switches, hypervisors, but also can be virtual machines or servers. Therefore, network access control deployed on physical gateway should still arrange to "logical" gateway.

Through Openflow protocols and controllers, network security based on SDN in cloud data center by setting flow table and guiding traffic that matches the policies, address the problems of Middleboxes deployment and "fragmentation" of security rules.

It is possible to optimize the controller by configuration and management, and make rules for dynamic migration and reconfiguration to solve security policy inconsistencies caused by the Middleboxes reposition and the virtual machine migration.

2. Related work

2.1 Research on Network Security in Data Center Network

Among the network security research in data center network, SDN based security is a hot topic both for academia and industry. So far as we known, the latest interesting research work in academia is concluded in the following.

FRESCO [15] introduces a new network security application development framework in SDN. FRESCO is to solve several key issues during combining security service components on demand. As OpenFlow is an open standard which can be able to provide greatly simplified and convenient when designing complex network security applications and integrating into large networks. However, so far there is still a lack of applications about the security aspects of the OpenFlow. FRESCO provides a scripting language API, security practitioners will be able to use these API to write security monitoring and threat monitoring logic into a modular library, which represents one basic processing unit in FRESCO, and can be connected together to provide a shared complex network security applications. FRESCO module can also customize flow processing rules, so as in response to the detected networks threats, it provides an effective workable method.

sLICK[16] proposes a network programming framework to separates the controller and Middleboxes, and provides an communication interface for them. There are more and more programs running on the controller want to across data plane and control plane to call functions, and still without a complete solution. While OpenFlow provides a rich programmable control plane, and it has a rather simple data plane. In contrast, the Middleboxes can effectively extend the data plane and provide a complex data plane, but cannot do effective integration in the control plane of SDN.

sLICK extends data plane programmability in two aspects which is different from existed systems. First, Slick can arrange complex data plane functions dynamically in the network, and can also guide a subset of data transmission through the appropriate function processing queue, which can adapt to the layout and response changed over time, in order to meet the changeable network conditions and transmission mode. Secondly, in order to achieve modularity, reusability, and integration strategy across multiple applications and network resources, Slick allows programmers to coordinate multiple actions between different entities in the data plane.

SIMPLE [17] can direct traffic to specific Middleboxes. Today's networks need to rely on Middleboxes to provide highly performance, highly security and efficient decision-processing capabilities. To achieve these goals, we need to ensure that network

flows directly goes through the required Middleboxes, which requires a lot of manual effort and the expertise of operator. In this respect, software-defined network (SDN) provides a promising option, but also introduces some layer 2 and layer 3 functions which do not belong to the traditional network model, such as policy components, resource management, and packet manipulation.

SIMPLE allows network operators to specify routing policies of a logical Middlebox, and translates the situation into the packet forwarding rules automatically according to the physical topologies, switching capacity and resource constraints of Middleboxes, to make data flows go through the required Middleboxes in specified order. Under the premise of existed SDN functions and without modifying the implementation of Middleboxes, SIMPLE can increase the flexibility of Middleboxes deployment, and can also generate and load new rules to maintain network stability when the Middleboxes fail and the network transmission is overload.

2.2 Network Virtualization with VMware NSX

VMware NSX [22] is a network virtualization platform for data center network. VMware NSX provides network management its software-defined data center that combines software-defined network (SDN) and network functions virtualization (NFV) [23], which is designed to achieve the rapid deployment of multiple layers of logical network without complicated configuration of physical devices, which greatly improves the flexibility of network function deployment. It provides an overall network for layer 2-7 and security model in a software way to achieve the decoupling of the underlying network hardware, and makes full use of the existing network infrastructure without making changes to improve the speed and agility of service delivery, and reduce costs.

The core components of VMware NSX are logic switches, logical routers, NSX API, logical Firewalls, logical load balancing, logical VPN etc. It can be said that NSX provides a new network virtualization approach allows data center operators to treat the physical network as an on-demand resource pool, which can break down the current network barriers and achieve agility and great economic benefits for the data center operators. Figure 1 depicts the overall structure of NSX.

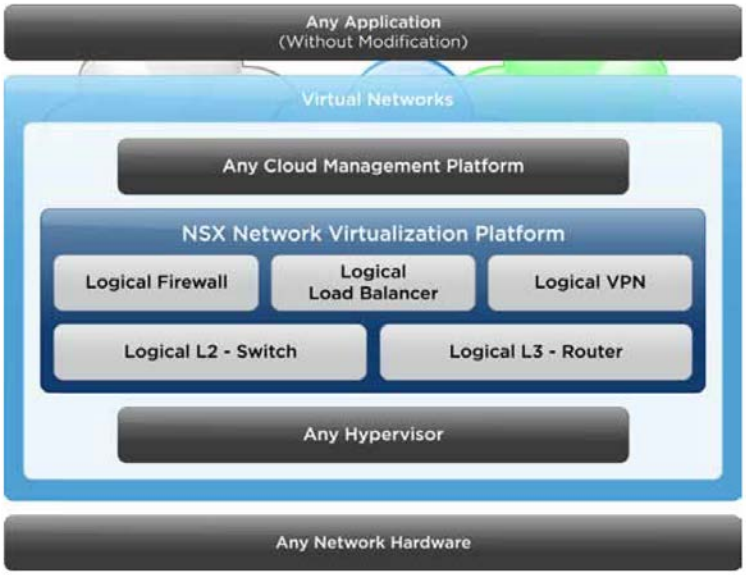

Figure 1. Network virtualization platform VMware NSX.

The scalability in NSX can provide complete services for VMware cloud service, and other network security vendors can optimize their deployment in virtual network. NSX platform uses distributed service framework which allows multiple hosts to integrate the network service, and a partner service modules can be easily inserted in by NSX API. NSX offers a variety of logical devices' interfaces, but it also need the hardware to achieve the mapping of logical-to-physical when face with a wide variety of hardware devices.

2.3 NSX and network security

NSX also provides a lot of common security resources, such as Firewalls and threat prevention tools, security services including Anti-malwares, Vulnerability management and IDS / IPS. In addition, the network service gateways bridge the physical and virtual environments, and application delivery services include load balancing, application delivery and WAN optimization.

NSX aims to integrate security appliance offered by other vendors. To simplify the third-party security products and integrate services, other vendors can use a single specific application programming interface (API). Once this API and the resources are ready, NSX's service composer tool can be used to deploy third-party Firewalls, Anti-malwares, Vulnerability management, Data protection, Intrusion detection and prevention (IDS/IPS) platform respectively.

There are internal stateful Firewalls in NSX management which can provide distributed Firewall for each virtual port. The Firewall also uses a variety of rules, audit and inspection technologies. It also provides the logic state, partner equipment/proxy connection, and can

also integrates third-party security platform.

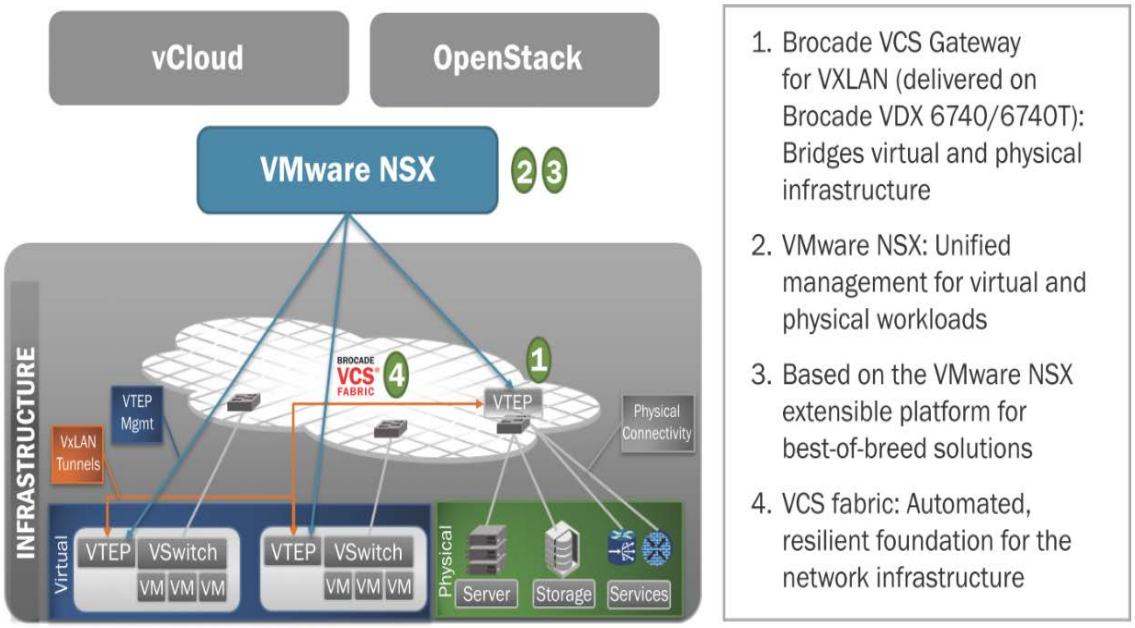

Figure 2. Brocade support NSX switch gateway.

Brocade VCS Gateway associates the virtual facilities with physical facilities, which allows physical devices to connect the virtual network. It provides data center operator with all types of applications in a unified network operating modes. By Brocade VCS Fabric technology, an organization can effectively use the current infrastructure to implement network management deployment. The implementation of Brocade VCS Gateway is shown in Figure 2.

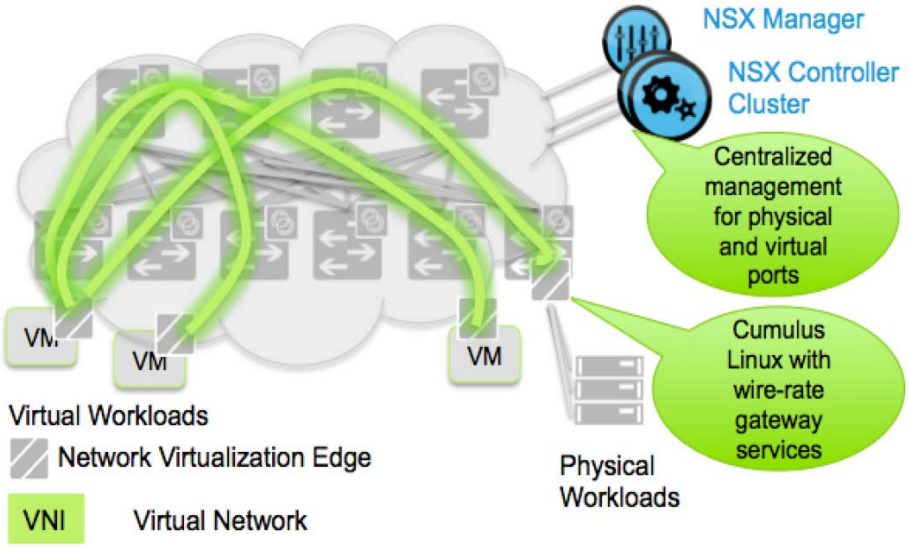

Figure 3. The integrated structure of Cumulus's NSX.

Cumulus Network proposes Cumulus Linux systems to provide rich programmable features and automated tools which used for the same computing environment on network

devices, which can quickly configure the physical network, and increase cost-efficiency when adding new network capacity. It provides support for network virtualization boundary functions by realizing coverage of the second layer gateway and stopping using the virtual network of VXLAN tunnel endpoint (VTEP), and registers NSX gateway service to further simplify management to the controller. Combined with the NSX, Cumulus Linux provides a network virtualization boundary to physical network at high connectivity. The integrated solutions of Cumulus's NSX is shown in Figure 3.

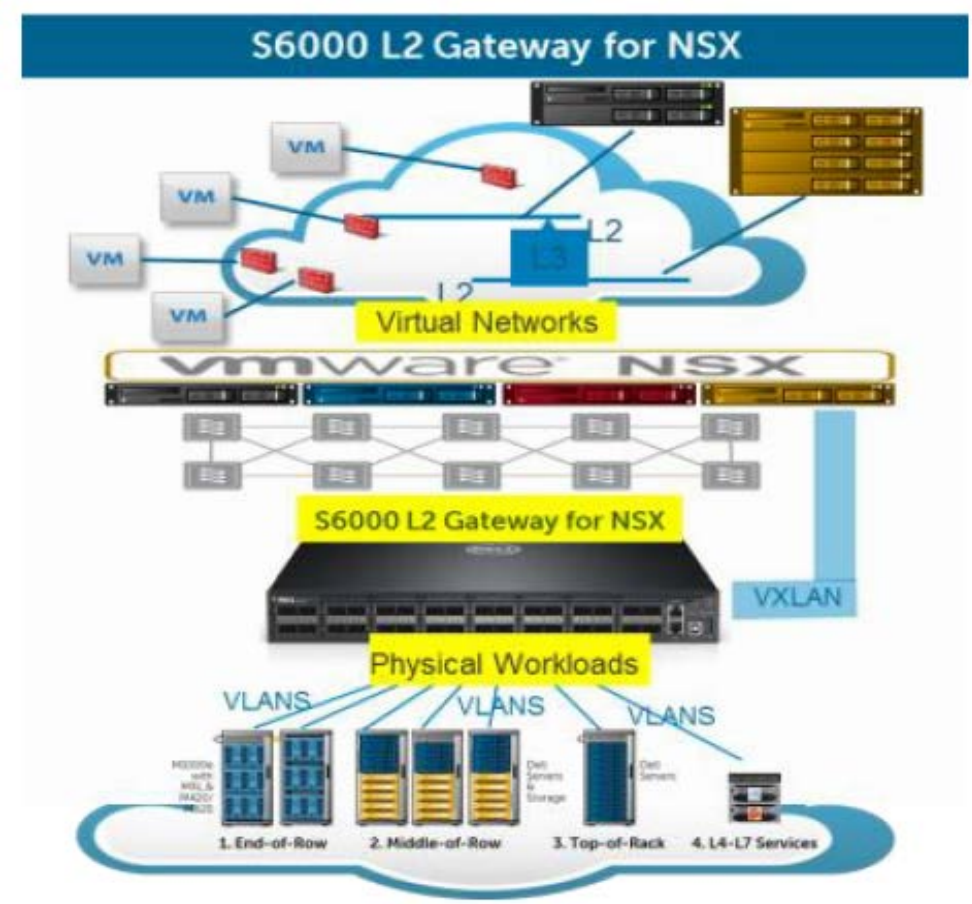

Figure 4. The integration of Dell S6000 switching gateway and NSX.

Dell S6000 data center switching gateways is proposed for the NSX, which can provide high programmability, automation features and scalability. S6000 as a high-performance gateway for the NSX can extend a virtual network to the physical server, or connect the physical workload which can be accessed by virtual LANs to the logical network via the second layer network services, also provide the migration from existing virtual environment to the public cloud and create a hybrid cloud etc. S6000 data center switching gateway for NSX is shown in Figure 4 with the integration of NSX.

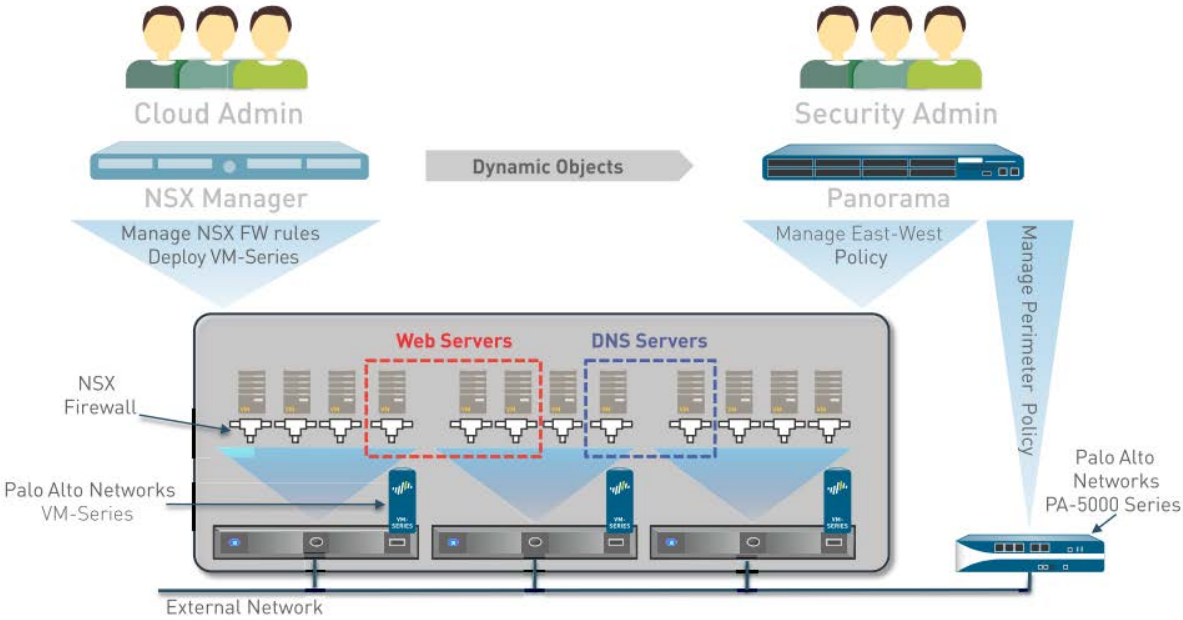

Figure 5. The integration of Palo Alto network's security platform and NSX.

Palo Alto Network [21] presents its security platform for the network security of data center, which provides identification and control functions, and can detect threats while ensuring the safe operation of the apps. It offers a variety means of defense for network threats, such as IPS and Anti-malware softwares to address the known threats, and through automated sandbox analysis of suspicious files to detect unknown malicious programs or APT (Advanced Persistent Threat) attacks. The deployment of its security platform is also shown in Figure 5 with the integration of NSX.

3. vCNSMS in motion

In this section, we propose vCNSMS, a collaborative network security prototype system. Compared with CNSMS[2], vCNSMS is designed and deployed for multi-tenant data center network, a similar implementation of Palo Alto network's security platform. In the following experiments, we demonstrate vCNSMS can be integrated into data center network environments. vCNSMS is constructed with home-brewed version of untangle open source multi-function gateway [11][25].

3.1 The principle of collaborative network security in DCN

3.1.1 Basic network topology

The Security Center and peer-UTMs are deployed in data center network as a cloud services or virtual appliances. In the bootstrap stages, the pee-UTM is running registration

process for Security Center.

Security Center for peer-UTM registration: There is a registration process in peer-UTM collaborative security module, and click the button to register. Security Center received the registration information, displaying the registered peer-UTM. Figure 6 shows the vCNSMS configuration and the registration process in bootstrap stage.

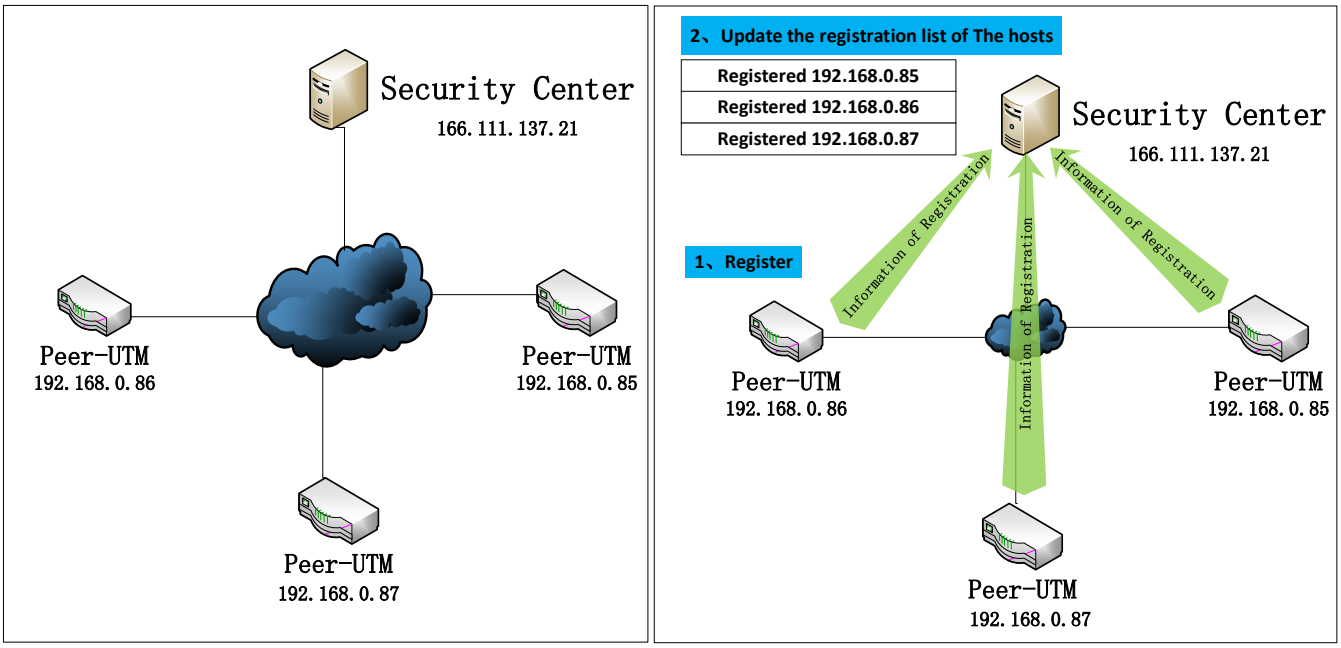

Figure 6. vCNSMS configuration settings and registration process in bootstrap stage.

3.1.2 Collaborative security in DCN

3.1.2.1 Security Center interacts with peer-UTMs

We assume each peer-UTM manage a virtual domain of a tenant in Data center, and the peer-UTM is obliged to the commands of Security Center.

1) Security Center issues rules: There are an option of the rule creation and an option of the rule issued.
2) Peer-UTMs report events: Security Center has a web interface to display the security incidents reported by peer-UTMs.
3) Events informed among peer-UTMs: There is a web panel in peer-UTM collaborative security module. The web panel shows the different peer-UTMs from the Security Center and security events detected by peer-UTM or Security Center.

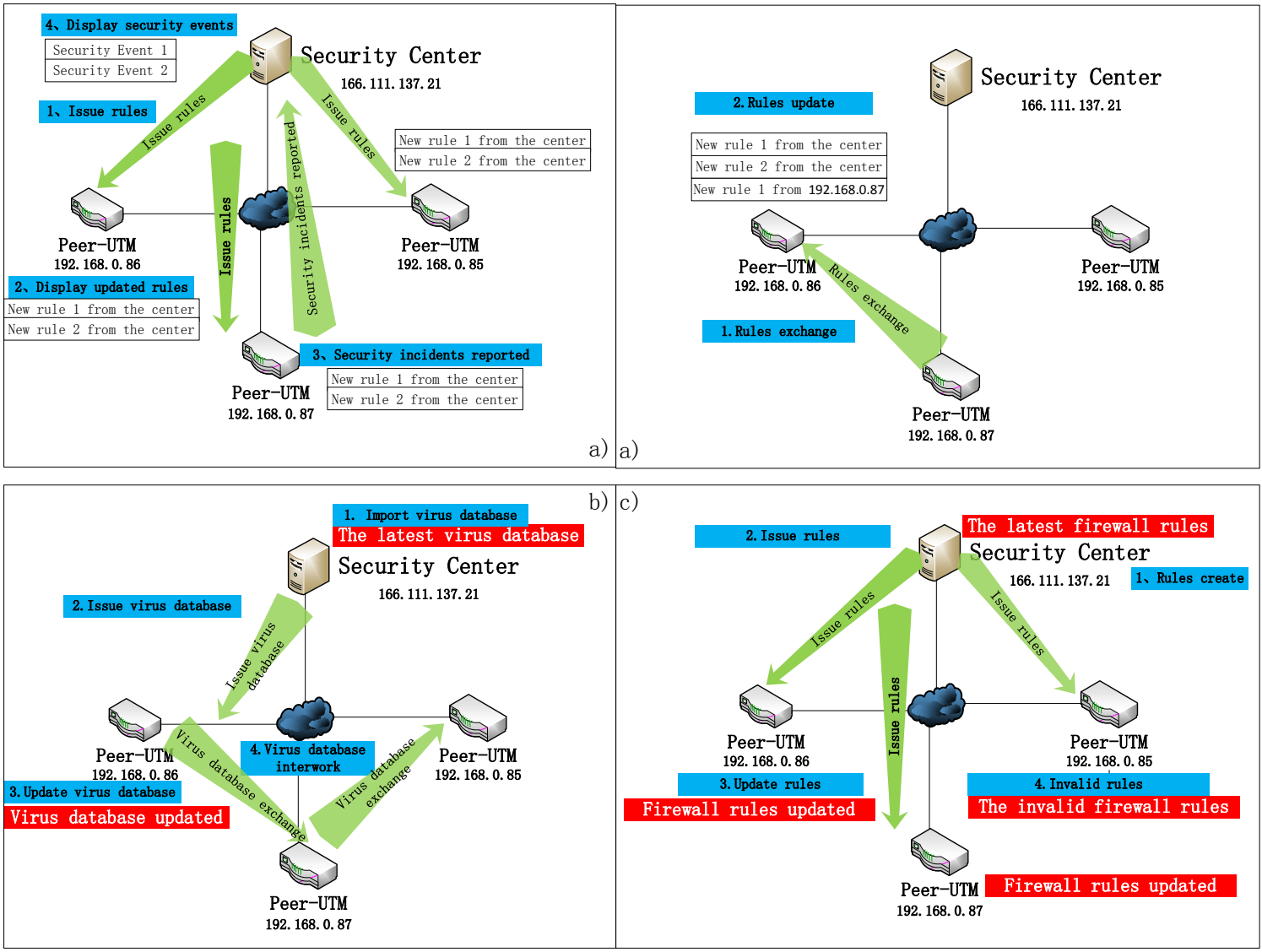

Figure 7. Collaborative network security Process in vCNSMS prototype.

3.1.2.2 Collaborative security for Antivirus

1) Security Center imports the virus signature database: There is an option to import the virus signature database in Security Center and disseminate them to peer-UTMs.
2) Security Center issues the virus signature database: There is an option to issue the virus signature database in Security Center.
3) Interface displays the virus database has been updated in PEER-UTM: The time stamp or version number changes.
4) Virus signature database synchronization among peer-UTMs, which use p2p mode.

3.1.2.3 Collaborative security for Firewall

1) Importing Firewall rules to Security Center: The option of importing Firewall rules to

Security Center.

2) Security Center issues the Firewall rules: The option of issuing the Firewall rules.
3) Interface displays the Firewall rules have been updated: There is a web panel in peer-UTM collaborative security module. The web panel shows the new Firewall rules have loaded in.
4) The peer-UTM choose to apply the rules which is updated by Security Center: In the display panel of the update rules, the peer-UTM can also invalid some rules as required.

3.1.2.4 Collaborative security for Protocol Control

Function and principle are same as above.

3.1.3 Security Center

Peer-UTM's Security Center runs on Debian Linux, which includes security rule distribution and event alarm services.

Rule distribution service is divided into server and client side, the main program of the server is a socket transfer module written with Java, which is manually activated in the Security Center. The client is embedded in peer-UTM Firewall and Rules control. When the server and client are normally running, Security Center can quickly transfer rules from a specific folder to peer-UTM.

Event alarm service is a web application based on Apache Tomcat, which opens port 80/443 for this web services. It listens the security events reported from peer-UTM, and dynamically displays in the admin webpage.

The format of a rule in communication between Security Center and peer-UTM is described in Appendix A in detail.

3.1.4 Virtualization Network based on SDN

Experimental platform based on openflow SDN is shown in Figure 8, the detailed configuration is shown in Appendix B. Network topology and host's IP allocated as follows, each virtual machine is bridged to eth0 192.168.0.0 segment for ssh control. Controller is also through ssh to communicate with openflow switch.

In openflow switch, there are three Ethernet ports, which can be connected according to the experimental needs. Each client has an Ethernet port that can be connected to any openflow switch according to the experimental needs.

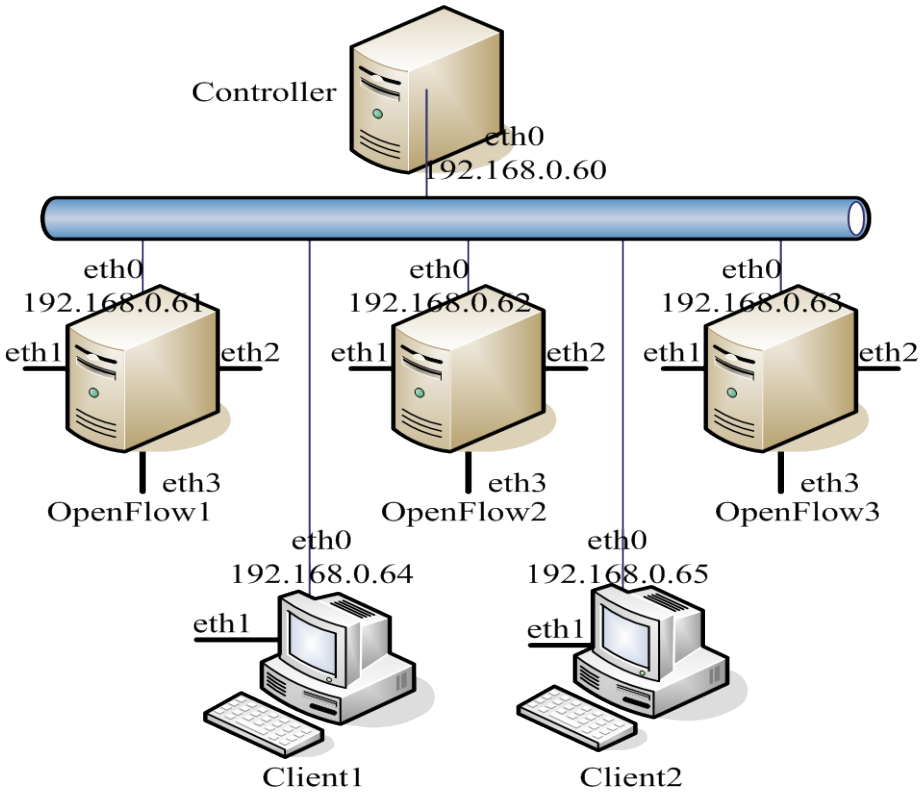

Figure 8. Virtualization Network based on SDN.

Here are some detailed deployments in setting-up.

1. In openflow 1.0, there is only a userspace datapath. The dpct which plays an important role in previous versions is replaced by ofdatapath and ofprotocol.
2. Ofdatapath is the implementation of datapath. It is responsible for forwarding packets based on flow table and communicating by ofprotocol and nox.
3. Ofprotocol is a middle program which controls ofdatapath and be controled by the nox.
4. dpctl can still be used to view the flow table in openflow switch, but can't be used to manage the datapath.
5. To change the behavior of openflow switch, the component (C or python) for nox is needed to run, and the component will be loaded with the nox.
   *Specific usage can be referred to the man page for ofdatapath, ofprotocol and dpctl.

3.2 Deep security check in vCNSMS

3.2.1 Function settings

Security Center, centrally managed security rules, collect the feedback information from the rule deployment, and store them into the security log.

1. Security rules is incrementally downloaded.

We set the current rule set and temporary rule set. In the Security Center, remove the duplicate rules in new rules. The new rule set will be added in a package and issued to the peer-UTM.

2. Firewall module.

Firewall rules will be downloaded from the Security Center and activated in the Firewall module, and the corresponding function will be achieved. Security events are collected and feedback to the Security Center.

3. UDP content filtering module.

Block the specified types of data packets matched with the specified patterns. The format of patterns are depicted in Appendix A.

3.2.2 Enhanced security functions

We modify the Protocol Control module in the prototype system to achieve the instant loading patterns and pattern matching for UDP protocol.

1. UTM routinely update security rules.

UTM regularly obtain and utilize configuration information of Firewall rule from the specified server. The configuration information includes the rules of content inspection of UDP and the blacklist based on “quintuple”.

2.Filtering based on the content of UDP packet.

According to the rules, the packets which contain specified features will be blocked, for example, the DNS request packets which contain the specified string "abc" will be blocked.

3. Blocking with blacklist

Block rule based on the "quintuple", e.g., disable the network connection which is like "TCP 1.2.3.4:444". Firewall module can quickly apply the update rule.

3.2.3 Implementation in peer-UTM

1. Update Firewall rules

Administrator puts a new Firewall rule set file in the specified folder named FirewallRule in the Security Center. The background program periodically checks this folder, if there is new rules in, Security Center will send announcement messages to all online peer-UTM to inform a new rule update. Peer-UTM receives messages from

Security Center, and compares with its own rules serial number (SN) in the rule database. If there is an update, peer-UTM will send the request to Security Center for downloading the appropriate rules. Then, the corresponding rules are downloaded to the local rule database, and activated in real time.

2.Firewall blocks the specified flow

Firewall module achieves rule based blocking function, and sends the corresponding block information to the Security Center by the announcement messages. After receiving the messages too, Security Center will feedback to the administrator from the page in real-time.

3. UDP Content Filtering

UDP Content Filtering uses the same mechanism with Firewall module, and set the appropriate tag in the announcement message. When UTM receives the announcement of the Security Center, it will sends a request message to Security Center for downloading the rules. When the rules are downloaded and activated, UTM will call the getPatterns function in LoadPatterns class to make the rules loaded in UDP filtering module. When UDP packet is parsed, the _findMatch() function in EventHandler class will be called for matching the content of UDP packet. The createRegExPattern function in PatternFactory class is called to generate RE pattern. And then, JAVA regular expression matching and matcher function in java.util.regex. Pattern class are used to match the rules and filter out the specified content. Finally, the UDP links which contain specify content will be blocked.

3.3 Intelligent flow processing in vCNSMS

Intelligent flow processing is an advanced method proposed in [26-28] for intrusion detection. Intelligent network flow processing of vCNSMS is based on smart packet verdict scheme. In this section, we propose security level based protection policy for intelligent flow processing in vCNSMS.

3.3.1 Security Level based Protection Policy

We describe the security level based protection policy as follows:

1. Security level setting
   Red, Yellow, Orange, and Green.
2. Intelligence network flow processing

According to the security level setting, different security rule set and packet verdict scheme are used with the consideration of different performance and load requirements.

3. Multi-function security gateway
   Peer-UTM can configure different security plugins on the demand of security level, and incurs different processing costs. Working mode of peer-UTM is also divided into Red, Yellow, Orange and Green.

3.3.2 Implementation-Security Level with security function plugins

We modify untangle shield module [11][12] and enhance it with smart packet verdict algorithm. The detailed functions are shown as follows.

1) Security plugin modules' rule set: S1(urgent)，S2(important)，S3(less important)，S4(trivial)
2) Security plugin module: plugin 1, plugin 2, plugin 3, ..., plugin *N*.
3) Security plugin module's processing: Tagging each packet with a Block mark, Pass mark, or Suspect mark.
4) Packet verdict on a packet is based on scoring system.
5) Packet processing principle:
   (1) Let the benign flows pass as soon as possible;
   (2) Recheck the suspected flow with larger rule set or security plugins;
   (3) Verdict packet with the context, i.e., current setting of security level, such as Red or Green.
6) Packet verdict processing takes also traffic load and performance penalty into consideration.

3.3.2.1 Smart packet verdict scheme with untangle Shield

Shield is a module of untangle, which function is to prevent DDoS. Shield reads a packet from nfqueue, and starts four threads, including a main thread, an event dispatch thread, an admission packet processing thread and a frontend microhttpd daemon. The main task of shield is to collect packet processing thread, and there is no parallel structure in the origin Shield implementation.

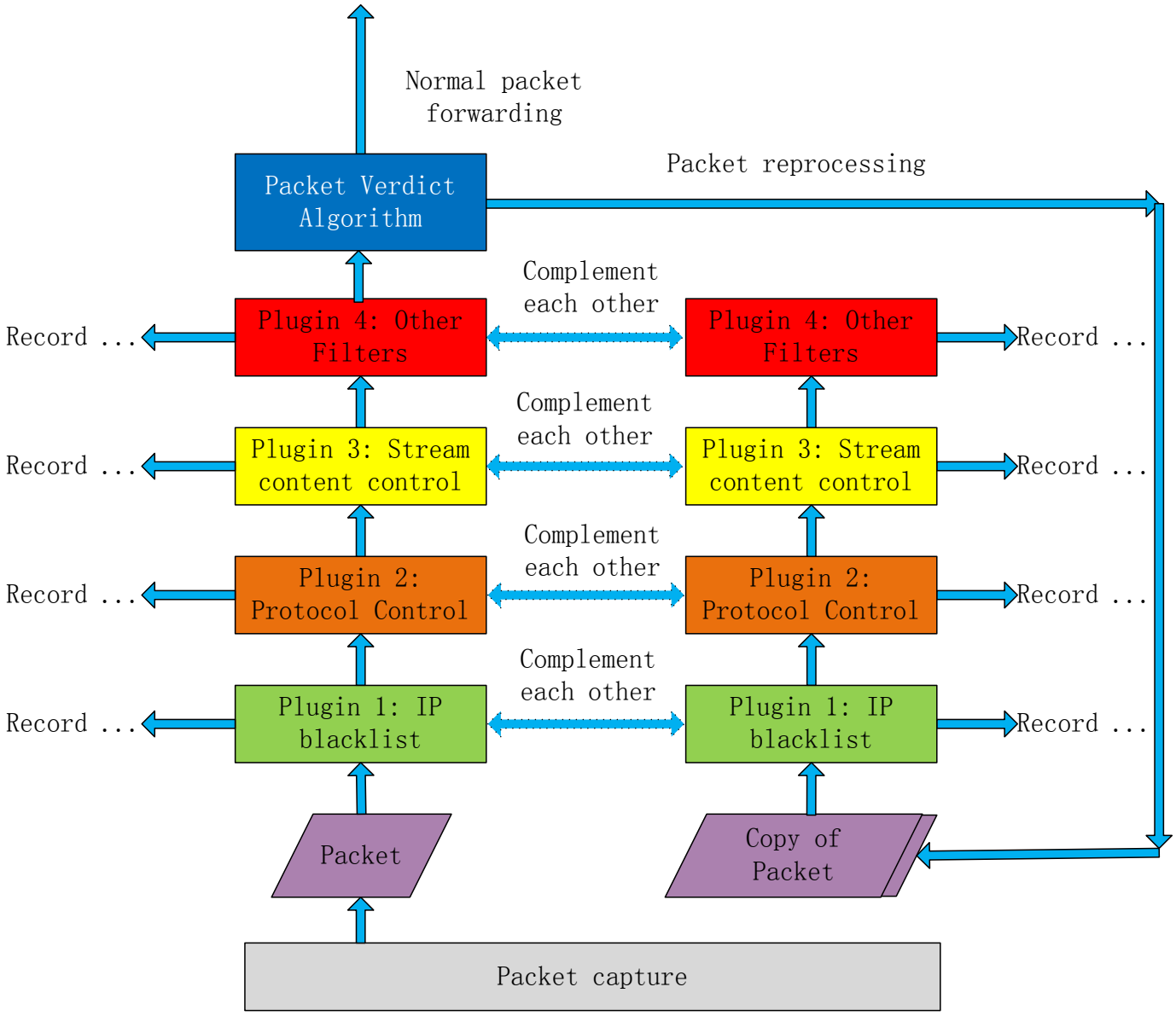

Figure 9. Smart packet verdict scheme used in vCNSMS.

3.3.2.2 Process level parallelization of Shield

We separate untangle Shield module, enhance it with smart packet verdict scheme. For performance improvement, we also modify Shield to integrate xtables[34] for parallelization.

Shield's test environment is shown in Figure 10. Parallelization scheme of shield has the following two ways:

1. Using multiple NFQUEUE and open multiple shield process, which is same as Snort_inline program [35].

Rewrite shield code to achieve a dispatcher and the corresponding queue in the program, achieving multi-threaded processing

The detailed parallelization scheme is given as follows:

1. Install xtables-addons-1.12-nslab, which is a NFQUEUEX module with the function of shunt.
2. iptables -A FORWARD -j NFQUEUEX --queue-num 4. The last parameter means to establish four NFQUEUE.
3. Start four or more Shield process, and their queue number and port number should be different.

```
./shield -p 3000 -q 0 &
./shield -p 3001 -q 1 &
./shield -p 3002 -q 2 &
./shield -p 3003 -q 3 &
```

Test environment topology is shown in figure xx. To achieve forward, not only with a good routing table of client1 and client2, but also through the following command to open the Shield machine kernel forwarding:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

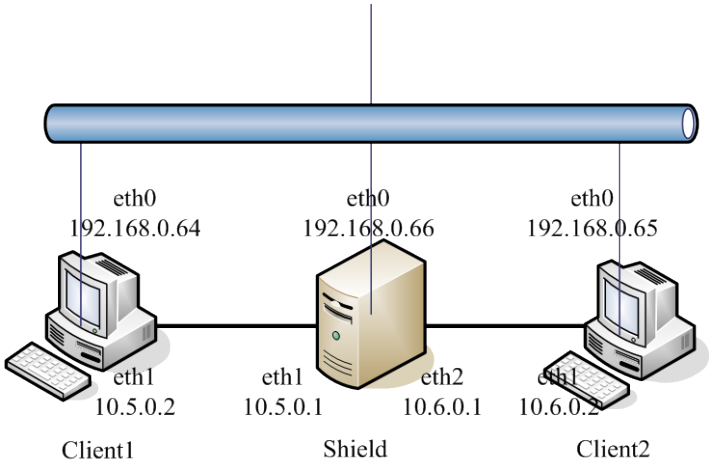

Figure 10. Performance evaluation in Shield parallelization implementation.

3.3.2.3 Performance testing of Shield's parallelization

Because of the amount of traffic with a single IP, UDP packets sent by traffic generator SmartBit will lead the origin Shield for blocking. In order to avoid the blocking, we modify the code of Shield. The performance measured in this way should reflect the true capability of Shield.

Test environment is described as follows:

1. Hardware:
   Intel(R) Core(TM) 2 Quad CPU Q9400 @ 2.66GHz. 4 cores. 2G memory. 8 Gigabit ports.
2. Software:
   Ubuntu 9.04, shield module of untangle.

3. SmartBit settings:

   4-way UDP streams with varied IP address and Ports. Packet loss ratio is collected in each experiment. Each experiment is conducted in a 1Gbps link with the varied input traffic load from 10% to 100% in percentage.

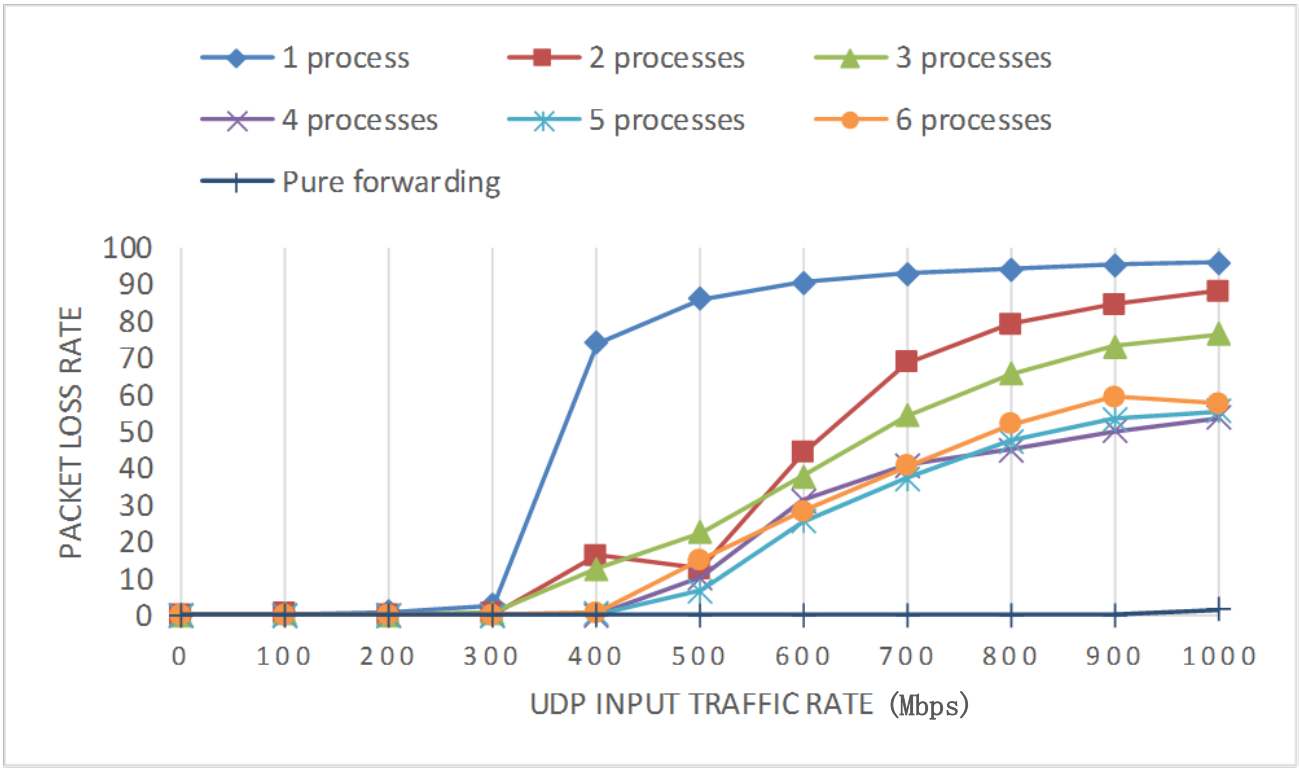

Figure 11. Throughput for smart packet verdict scheme in vCNSMS.

As shown in Figure 11, 5 parallel shield processes can handle 400 ~ 500Mbps traffic. CPU occupancy rate of Shield is about 40%. Memory consumption is quite minimal. As Shield only deal with the head of packet within this test, the performance is expectable, but it is slower than snort [14] in comparison.

3.3.2.4 Bottleneck performance Analysis of Shield

We use Intel VTune tool in the compile stage of Shield module. With performance evaluation experiments of Shield module, there is a lot of useful information in function call graph inside Shield module. The most time consuming operation in packet handling are concentrated in barfight_net_nfqueue_read () and _handle_packet () wherein smart packet verdict scheme is running. In addition, debug function is also very time consuming.

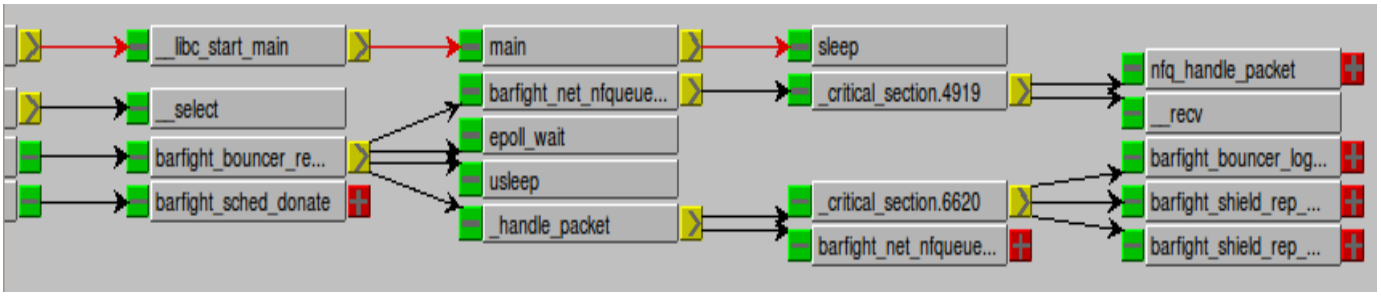

Figure 12. Performance analysis with VTune.

As we will run Shield module in a virtual machine, performance issue in virtualized environments is a critical and open problem, which has already been discussed in [32]. We can further improve performance with vPF_ring [33].

4. Conclusions

In this paper, we propose vCNSMS to address network security in multiple tenant's data center network and demonstrate vCNSMS with centralized collaborative scheme. vCNSMS can further integrate intelligence packet verdict algorithm for smart packet inspection to defense possible network attack inside data center network. SDN based virtualization network in data center can deploy vCNSMS for flexibility and scalability to protect multiple tenant with different scrutiny policy and security requirement. The smart packet verdict scheme can be used for deep defense in data center network.

5. Future work

As the practical deployment and operation experience of vCNSMS in data center network, vCNSMS's Security Center can collects more and more security rules and its events. It is possible to detect network policy violation and intrusion attack with AI (artificial intelligence) based on matching learning method w/o supervise. It is a promising area for further exploration in future.

Acknowledgements

This work was supported in part by Ministry of Science and Technology of China under National 973 Basic Research Program (No. 2013CB228206 and No.2012CB315801), National Natural Science Foundation of China (grant No. 61233016), and China NSFC A3 Program (No.61140320).

This work is also supported by Intel Research Council with the title of "Security

Vulnerability Analysis based on Cloud Platform with Intel IA Architecture".

References

[1] Fuye Han, Zhen Chen, Hongfeng Xu and Yong Liang, A Collaborative Botnets Suppression System Based on Overlay Network, International Journal of Security and Networks, Vo. 7, No. 4, 2012.

[2] Zhen Chen, FuYe Han, Junwei Cao, Xin Jiang, Shuo Chen, Cloud Computing-Based Forensic Analysis for Collaborative Network Security Management System, Tsinghua Science and Technology, 18 (1), pp.40-50, 2013.

[3] Xinming Chen, Kailin Ge, Zhen Chen and Jun Li, AC-Suffix-Tree: Buffer Free String Matching on Out-of-Sequence Packets, Proc. of the 7th ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS), 2011.

[4] Tianyang Li, Fuye Han, Shuai Ding, Zhen Chen, LARX: Large-scale Anti-phishing by Retrospective Data-Exploring Based on a Cloud Computing Platform, ICCCN GridPeer workshop, 2011.

[5] Beipeng Mu, Xinming Chen, Zhen Chen, A Collaborative Network Security Management System in Metropolitan Area Network, Proc. of the 3rd International Conference on Communications and Mobile Computing (CMC), 2011.

[6] Xinming Chen, Beipeng Mu, Zhen Chen, NetSecu: A Collaborative Network Security Platform for in-network Security, Proc. of the 3rd International Conference on Communications and Mobile Computing (CMC), 2011.

[7] Donghua Ruan and Zhen Chen et al., Handling High Speed Traffic Measurement Using Network Processors, ICCT 2006.

[8] Jia Ni, Zhen Chen et al., A Fast Multi-pattern Matching Algorithm for Deep Packet Inspection on a Network Processor, ICPP 2007.

[9] Zhen Chen et al., AntiWorm NPU-based Parallel Bloom filters in Giga-Ethernet LAN, IEEE ICC'2006.

[10] Zhen Chen et al., AntiWorm NPU-based Parallel Bloom filters for TCP-IP Content Processing in Giga-Ethernet LAN, IEEE LCN WoNS'2005.

[11] Untangle open source appliance, www.untangle.com, see https://gitorious.org/untangle.

[12] Dirk Morris, John Irwin, Robert Scott, Methods and systems for reputation based resource allocation for networking, US Patents application No.US20070043738.A1.

[13] L7 filter project, http://l7-filter.sourceforge.net/Pattern-HOWTO.

[14] Snort, http://www.snort.org

[15] Bro, http://www.bro-ids.org

[14] Qihoo 360 Internet Security Center, development trend of enterprise security in internet ages, http://www.gartner.com/technology/media-products/pdfindex.jsp?g=Qihoo_issue1.

[15] Shin, Seugwon, Phillip Porras, Vinod Yegneswaran, Martin Fong, Guofei Gu, and Mabry Tyson. "FRESCO: Modular composable security services for software-defined networks." In Proceedings of Network and Distributed Security Symposium. 2013.

[16] Anwer, Bilal, Theophilus Benson, Nick Feamster, Dave Levin, and Jennifer Rexford. "A Slick Control Plane for Network Middleboxes." HotSDN, 2013.

[17] Qazi, Zafar Ayyub, Cheng-Chun Tu, Luis Chiang, Rui Miao, Vyas Sekar, and Minlan Yu. "SIMPLE-fying Middlebox Policy Enforcement Using SDN." ACM Sigcomm, 2013.

[18] Sekar, Vyas, Michael K. Reiter, Walter Willinger, Hui Zhang, Ramana Rao Kompella, and David G. Andersen. "cSamp: A System for Network-Wide Flow Monitoring." In NSDI, pp. 233-246. 2008.

[19] Wang, Kai, Yaxuan Qi, Baohua Yang, Yibo Xue, and Jun Li. "LiveSec: Towards Effective Security Management in Large-scale Production Networks." In Distributed Computing Systems Workshops

(ICDCSW), 2012 32nd International Conference on, pp. 451-460. IEEE, 2012.
[20] Wang, Xiang, Zhi Liu, Yaxuan Qi, and Jun Li. "LiveCloud: A lucid orchestrator for cloud datacenters." In Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on, pp. 341-348. IEEE, 2012.
[21] Zhang, Ying, Fachao Deng, Zhen Chen, Yibo Xue, and Chuang Lin. "UTM-CM: A practical control mechanism solution for UTM system." In Communications and Mobile Computing (CMC), 2010 International Conference on, vol. 1, pp. 86-90. IEEE, 2010.
[22] VMware NSX network virtualization platform, http://www.vmware.com/products/nsx/
[23] VMware NSX network virtualization platform Security Services,
[24] VMWare Network security, http://www.vmware.com/products/nsx/resources.html
[25] Y. D. Lin, R. H. Hwang, F. Baker, "Computer Networks: An Open Source Approach," McGraw-Hill, February 2011.
[26] Lin, Ying-Dar, Huan-Yun Wei, and Shao-Tang Yu. "Building an integrated security gateway: Mechanisms, performance evaluations, implementations, and research issues." IEEE Communications Surveys & Tutorials, vol. 4, no. 1, pp.2-15, 2002.
[27] Lin, Ying-Dar, Chih-Wei Jan, Po-Ching Lin, and Yuan-Cheng Lai. "Designing an integrated architecture for network content security gateways." Computer 39, no. 11 (2006): 66-72.
[28] Lu, Chun-Nan, Chun-Ying Huang, Ying-Dar Lin, and Yuan-Cheng Lai. "Session level flow classification by packet size distribution and session grouping." Computer Networks 56, no. 1 (2012): 260-272.
[29] Jain, Sushant, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, Subbaiah Venkata et al. "B4: Experience with a globally-deployed software defined WAN." In Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM, pp. 3-14. ACM, 2013.
[30] Junda Liu, Ensuring Connectivity via Data Plane Mechanisms, 10th USENIX Symposium on Networked Systems Design and Implementation (2013).
[31] Junda Liu, Data-driven network connectivity, Proceedings of the 10th ACM Workshop on Hot Topics in Networks, 2011.
[32] Schultz, Michael J., and Patrick Crowley. "Performance Analysis of Packet Capture Methods in a 10 Gbps Virtualized Environment." In Computer Communications and Networks (ICCCN), 2012 21st International Conference on, pp. 1-8. IEEE, 2012.
[33] Cardigliano, Alfredo, Luca Deri, Joseph Gasparakis, and Francesco Fusco. "vPF_RING: towards wire-speed network monitoring using virtual machines." In Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference, pp. 533-548. ACM, 2011.
[34] Xtables-addons, http://xtables-addons.sourceforge.net.
[35] Snort-inline, http://snort-inline.sourceforge.net/.

Appendix A: Communication Message Structure between Security Center and peer-UTMs

A1. Announcement Message Format (XML or JSON)

Security Event Announcements in XML format

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE jxta:MSA>
```

```
<jxta:MSA xml:space="default" xmlns:jxta="http://jxta.org">
        <MSID>

urn:jxta:uuid-C13608789CFC42168A63914AC05092E4C6DFFF7E1DCD451B9D03C497CEEF01
3706
        </MSID>
        <Name>
                peerUTM advertisement
        </Name>
        <Desc>
                firewallevent
        </Desc>
        <Crtr/>
        <SURI>
                nullmac:00:10:f3:1c:ca:04  time:2011-54-11  06:54:57  TCP  from
/10.0.0.183:52264 to /166.111.137.20:80 was blocked$
        </SURI>
        <Vers/>
</jxta:MSA>
```

A2. Rule format for Firewall module in JSON

Format of prototype system Firewall rules.

[{"trafficblocker":"1","protocol":"any","srcaddress":"any","dstaddress":"any","srcport":"any","dstport":"any","srcintf":"any","dstintf":"any","description":"test"}]

A3. Rule format for Protocol Control module (refer to l7-filter HOWTO)

Rule format of Protocol Control rule such as MSN (# is comment).

```
# MySpace IM - MySpace chat client
# Protocol groups: chat proprietary
#myspaceim
# Written by community
^\\login2.*?\\final\\$
```

Rule Format of Protocol Control rule for DNS.

```
# Protocol – DNS request/reply
DNS
```

```
# Protocol groups:
UDP
#Port
53
# Written by community
Abc
```

Appendix B: Configuration in openflow SDN environment.

Table B1. Detailed configuration settings in Virtualization Network based on SDN.

HOST	IP	hostname	Hosts alias	system
Controller	192.168.0.60	saturn-controller	control	Ubuntu 10.10
OpenFlow1	192.168.0.61	saturn-openflow1	of1	Ubuntu 9.10
OpenFlow2	192.168.0.62	saturn-openflow2	of2	Ubuntu 9.10
OpenFlow3	192.168.0.63	saturn-openflow3	of3	Ubuntu 9.10
client1	192.168.0.64	client1	c1	Ubuntu 10.10
client2	192.168.0.65	client2	c2	Ubuntu 10.10