Challenges of Network Forensic Investigation in Virtual Networks

Daniel Spiekermann[1] and Tobias Eggendorfer[2]

[1]*FernUniversität Hagen, Germany*
[2]*Hochschule Ravensburg-Weingarten, Germany*
E-mail: daniel.spiekermann@fernuni-hagen.de;
tobias.eggendorfer@hs-weingarten.de

Received 10 November 2016; Accepted 21 December 2016;
Publication 3 January 2017

Abstract

The evolution of virtualization techniques is changing operating principles in today's datacenters. Virtualization of servers, networks and storage increases the flexibility and dynamic of the environment by reducing the administrative overhead. Based on a physical underlay network, different logical networks are implemented with new protocols like VXLAN, STT or GENEVE. New paradigms like Software-Defined-Networks or Network Function Virtualization offer new capabilities to redesign the whole network infrastructure. This trend creates new challenges for digital investigations analysing incidents by extracting and interpreting recorded data inside the environment. As a branch of digital investigation, network forensic investigation is used to examine network traffic by capturing the data of a suspicious target system and analysing this data. In this article, we analyse in detail new challenges in investigating virtual networks. We propose a classification in three categories, which might help to develop new methods and possible solutions to simplify further necessary investigations in virtual network environments. The defined challenges are classified according their potential to impede the investigation. Based on this classification we derive a list of basic conditions, describing different necessary requirements to implement a successful, valid and ongoing network forensic investigation in these virtual networks.

Journal of Cyber Security, Vol. 5_2, 15–46.
doi: 10.13052/jcsm2245-1439.522

Keywords: Virtual networks, network forensic, digital investigation.

1 Introduction

The evolution of cloud computing environments led to a new use of information and communication technology (ICT). Nowadays customers use virtual machines (VM), applications or storage pools maintained by a third party instead of managing the complete ICT infrastructure by themselves. Not only companies and professionals are taking advantage of the new possibilities but also the ordinary user.

The cloud service providers (CSP) offer different services [30], distinguishes between three implementations. An Infrastructure-as-a-Service (IaaS) environment offers the access to different VMs, each configured by the customer. Platform-as-a-Service (PaaS) as the second implementation provides a run-time environment for different application or programming languages. Users of a PaaS are able to implement their own applications using this virtual environment. The main advantage is the renunciation of administrating the run-time environment, the hardware and all affiliated components. Last Software-as-a-Service (SaaS) offers only one special application to be used by the customer.

The customers scope of influence of PaaS and SaaS environments is limited, but the usage of VMs provided by an IaaS environment raise new problems for CSPs. Customers claim their VMs interconnected with each other, demand a flexible, but secure environment and the capabilities to change their rented environment on their own. Additionally the interconnection of their VMs to the internet or inside a multi-tier architecture is administrated without further interaction of the CSP only by the customer itself. Providers endeavour an “easy to manage” environment with few additional manual interventions to reduce costs and error susceptibility. The evolution of software containerization platforms like Docker [31] or LXC [11] increase the flexibility inside the environment. By using orchestration services like Kubernetes or OpenShift the management overhead of containerized applications is further reduced.

The implementation of virtual networks involves new options to overcome the problems and to satisfy customer and CSP. Virtual networks abstract from the underlying physical network and provide a large number of so-called overlay networks. These virtual networks expand the flexibility inside the ICT and enable the customer to maintain the assigned network on their own. The CSP does not have to configure the requested network environment, the customer is able to configure the network infrastructure and desired connections without any further support. The implementation of virtual networks span different techniques like new network protocols or new paradigms like Software Defined networks (SDN) and Network Function Virtualization (NFV).

However, the need for digital investigation in virtual environments is nevertheless essential. Digital forensics encompass the recovery and analysis of data, stored or processed on digital devices to investigate cyber crimes. A digital investigation inside a cloud environment is already a complex, error-prone and tedious task, aggravated by the flexible environment and jurisdictional, organizational or technical problems [38].

By establishing connections between client and server, provider and customer or storage systems and application by transferring the requested data from sender to receiver, networks are an important part of modern IT systems. By intercepting and recording this traffic for a posterior analysis, NFI may extract relevant information to clarify the issue. Therefore Network Forensic Investigation (NFI) is mostly a reliable source of information gathering in current networks, which might help to clarify the investigated alleged facts.

The change of network infrastructure creates lots of new challenges for NFI. The use of virtual networks inside IaaS environments impedes the digital investigation in this environment. Proved techniques and methods fail because of the increased complexity and flexibility of the new logical networks. All stages of NFI are faced with different issues originating with the virtual networks. The high dynamic of the environment complicates the capture and storage process of the NFI, the use of new network protocols like VXLAN or STT impede the subsequent analysis of the stored data. Hence the entire NFI process needs to be refined, in order to ensure a successful NFI in virtual networks.

The remainder of this article is structured as follows. Section 2 provides an overview of the current research, related to network forensics, network virtualization and digital investigation in virtual environments. Section 3 defines the process of NFI, clarifies differences to other sections of digital investigation and explains in detail the three main phases of a NFI. Section 4 provides the background knowledge of network virtualization and the different parts of SDN, NFV and relevant protocols. The arising problems of NFI in virtual networks are derived by analysing the differences between physical and virtual networks and described in detail in Section 5. Section 6 subsequently identifies requirements, which ensure a valid and ongoing NFI in virtual networks. Section 7 concludes and gives an outlook on the further intended research.

2 Related Work

The evolution of virtual networks raise different problems for digital investigation in these environments. Problems of digital investigation in virtual environments like cloud computing or virtual datacenters are discussed in [12, 38, 47].

NFI as a branch of digital investigation is discussed in [36] with a focus on tools, techniques and process models [9]. defines the capture and analysis process of NFI and the different steps of parsing and extracting information. Tools and techniques of NFI are discussed in [16, 29]. [41] describes the integration of digital investigation in network environments. [25] presents an overview about the main tools and techniques to ensure forensic investigation in networks and defines a ruleset for evidences in NFI.

The multitude of different implementations of network virtualization is discussed in [2, 4, 5, 18].

SDN and its inherent principles and features are discussed in [19, 24]. NFV is discussed in [15, 44].

This paper discusses critical problems in virtual networks, which impede or abort a NFI. A discussion of network forensic investigation in cloud computing environments can be found in [39, 42]. A list of open problems can be found in [21], although not regarding the special characteristics of virtual networks.

3 Network Forensic Investigation

Digital investigations are used to solve crimes involving computers, networks or other IT components. Depending on the kind of cyber crime, different highly specialized investigation methods are necessary. If the communication between two systems contains relevant information, a detailed analysis of this data is helpful to identify the further circumstances. The combination of the retrieved results with other branches of digital investigation like computer forensics might improve the overall examination.

[33] defines NFI as

> *the use of scientifically proven techniques to collect, fuse, identify, examine, correlate, analyse, and document digital evidence from multiple, actively processing and transmitting digital sources for*

> *the purpose of uncovering facts related to the planned intent, or measured success of unauthorized activities meant to disrupt, corrupt, and or compromise system components as well as providing information to assist in response to or recovery from these activities.*

Thereby NFI is separated into different phases [36], which can be classified in *Capture, Record* and *Analysis.* We define the two aforementioned phases as *online,* the last part of analysing as the *offline* phase.

3.1 Capture

Contrary to the other branches of digital investigation, there is no automatic data storage without interventions. Personal computer (PC) or mobile phones store the user data even when the system is shut down. Network devices transfer the data from sender to receiver over a separate medium without storing any of this data. Techniques like buffering network data is only used to compensate latency or short-ranged interferences and yields no helpful information.

The purpose of the capture process is to get all transferred data of the target system without packet loss or a potentially manipulation of the packets. Because of this an approach of manipulating routing tables or firewall rules is not acceptable in NFI.

The capture process is necessary to gain access to the transferred data and to copy the data. Three main techniques exist to fulfil this requirement:

- **Bridge**
  A Bridge is a separate device interleaved in the connection between the target system and the next network device. If the bridge device loses the connection, the whole communication to the target system breaks down.
- **Tap**
  A Test-Access-Port defines a special network device, which is interleaved in the cable connection between target system and network device. The main advantage of a TAP against a bridge is the established connection, which remains online even when the TAP crashes.
- **Port mirror**
  A port mirror[1] can be created on professional network switches to mirror all traffic from or to a given physical port on the switch. The switch copies all network traffic passing this source port and delivers it to the

[1] Also called *Switch Port ANalyzer (SPAN).*

destination port. If the mirror crash[2], the connection between sender and receiver is still online.

Each of these techniques has to be adapted to the given physical medium network packets are transmitted through. The captured network data contains the information of all layers of the OSI model. Two other methods of copying transferred data are a proxy-environment or the use of dedicated VMs, but both implementations are incongruous in virtual networks.

3.2 Record

The mirrored data needs to be recorded by a separate storage system. The amount of captured data depends on the target system, therefore the storage system should provide enough free space to store the networks packets. Additionally the storage system should be capable to handle transmission peaks or a higher transfer rate.

The limiting factor of the storage system is the write rate of the hard disk. Networks with a transfer rate of 10 gigabit per second (GBit/s) of higher transmit approximately 1.1 gigabyte per second (GB/s), hence the storage system has to write this data with the same speed to the hard disk to avoid data loss. Higher transfer rates like 40 GE or 100 GE[3] transfer even more data per second, therefore the storage system has to be aligned to the network environment.

Capture filters provide a limitation of the recorded data by analysing each incoming network packet and determine the further processing by exact criteria. By matching a given value the packet is stored, otherwise it is discarded. The use of capture filter is contentious, in law enforcement investigation the use of capture filter is denied. The risk to lose relevant packets is too high.

3.3 Analysis

In this offline phase the captured data is examined and checked in detail. Different software tools help the investigator to reduce the amount of data, filter for relevant information and reassemble the transferred data. The result of the analysis depends on the aforementioned phase, a valid capture and recording process ensures the storage of the entire network traffic, which was

[2] E.g. in case of a system overload, port mirrors are mostly established with a lower priority.

[3] 40 GE describes 40 GBit/s or $\approx$ 5 GB/s, 100 GE describes 100 GBit/s or $\approx$ 12.5 GB/s transfer rate.

sent or received by the suspicious system. Even an absence of few packets hamper the further analysis and might cause wrong pointed results.

In addition to this other sources like logfiles, SNMP messages [43] or NetFlow [20] information are used to combine the results with the captured network traffic.

4 Network Virtualization

Like VMs act as virtual implementations of computer systems running on physical systems, virtual networks act as logical implementations of networks running on underlying network devices.

[5] defines virtual networks as

> *A networking environment supports network virtualization if it allows coexistence of multiple virtual networks on the same physical substrate. [...] Essentially, a virtual network is a subset of the underlying physical network resources.*

The beginning of network virtualization was made with techniques like Virtual Private Networks (VPN) or Virtual Local Area Networks (VLAN). But the need for a higher virtualization rate gets more important with the evolution of cloud computing environments. In this environment events like creation, deletion or movement of VMs are performed more frequently than in current ICT-environment. But the networks implemented inside the current ICT-environments limit the needed flexibility, reduce the dynamic in the network and thwart a comprehensive benefit of the virtualization. The CSP still has to administrate the network environment with vendor-specific tools or manually by changing configuration files, which interferes the quick change of network topologies and opposes the intended purposes.

Customers demand a secure environment, running their VMs in an isolated network, without foreign VMs inside their setting, however with a connection to external networks. Their environment should be maintained by themselves, without any administration tasks by provider.

With the virtualization of networks this process gets more comfortable. VMs are connected to a logical, separated network, which is provided inside the virtual environment. The user does not need to know how the underlying network is implemented or how its assigned network is created. All tasks of administration, operations and maintenance of the underlying network is hidden by providing the logical overlay network.

Other paradigms of virtual networks are SDN and NFV, which implement different ways to manage to network and change the kind of used equipment.

4.1 Novel Protocols

An implementation with VLANs fulfils the requirements of security and isolation, but the limitation of 4096[4] logical networks impede an implementation in bigger networks. These limitations led to an evolution of new network protocols, which intend to increase the number of possible participants or to hide internal address schemes of the logical networks. Protocols like QinQ [17], Stateless Transport Tunneling (STT) [1] or Virtual eXtensible LAN (VXLAN) [26] encapsulate the internal network protocols and transfer the new protocols to a given endpoint. Generic Network Virtualization Encapsulation (GENEVE) [14] encapsulates the payload by prefixing a GENEVE and a UDP-Header. VXLAN implements a new VXLAN-Header with a 24 bit tag called *VNI* (VXLAN network identifier) and a UDP-Header, too. Figure 1 shows the final packet format for transmitting the internal payload via VXLAN.

STT uses a STT and a TCP-like header to tunnel the origin network data inside the network. Table 1 lists the different types of encapsulation of the new protocols.

4.2 Software Defined Networks

SDN as a new paradigm of network infrastructure allows the network administrator to manage the whole network at an abstract level. The need of configuring single devices is replaced by a central configuration of the whole network as single.

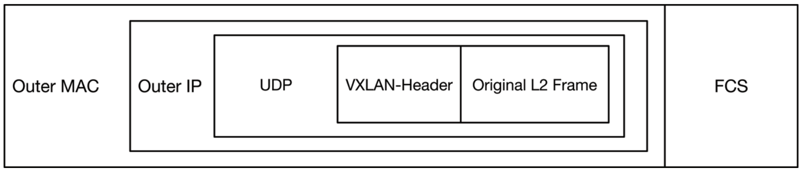


Figure 1 VXLAN encapsulation.

[4] If the IEEE 802. 1q standardization is used.

Table 1 Encapsulation protocols

Protocol	Outer Protocol	Additional Header
STT	TCP-like	STT
VXLAN	UDP	VXLAN
QinQ	VLAN IEEE 802.1q	VLAN IEEE 802.1q
GENEVE	UDP	GENEVE
NVGRE	GRE	–

To achieve this, SDN separates the control and the data plane of a switch and shifts the control plane to an external device, namely the controller. The data plane resides on the switch and still forwards the network packets to the correct interfaces. The decision of the forwarding process is made by the decoupled forwarding plane located on the controller. A special communication channel is established between controller and switch to transfer the relevant forwarding decision. The most notable protocol to transfer this information is OpenFlow [28].

Another fundamental feature of SDN are open interfaces, which enable an interchangeability of network devices of different vendors. In the absence of open interfaces, the advantages would be taken away, which would lead to a proprietary and inflexible environment.

Based on the open interfaces the third key feature of SDN gets real. With the shift to SDN the programmability of the network becomes significant. By decoupling all forwarding planes and shifting them into the controller, the network can be administrated by changing the configuration at only one place. This eliminates the necessity to configure different devices because of a network change. These changes might emerge by the installation of new servers, creation of new network scopes or just by moving services from one to another machine. The network administrator currently has to adapt logical assignment from ports to VLAN-IDs, rewrite firewall rules or add new routing entries. The abstraction of management enables the configuration of controller without touching every device. Neither the configuration of single devices nor the vendor-specific configuration remains.

4.3 Network Function Virtualization

NFV describes a new approach to decouple software implementation from the underlying hardware of network devices [15]. A group at the European Telecommunications Standards Institute (ETSI) [13] is working on standardization of these implementations.

By decoupling the software from the hardware it is possible to transfer the software in virtual appliances, which are capable to run on commodity hardware. Thus a network function like a firewall, a router or a switch can be provided on demand at nearly any position in the network infrastructure. Additional benefits are cost reduction by omitting hardware solutions and better service provisioning by scaling up the performance based on the network state. One of the most notable implementations of such a component is Open vSwitch (OVS) [35] as a virtual switch.

5 Problems

This section describes the new problems, which arise in NFI of virtual networks. Not all problems are integrally new in the wide field of digital investigation, but occur now for the first time in NFI. Aspects like multi-tenancy or the customization of the leased environment complicate current computer forensic investigation equally. The evolution of network virtualization takes this problems to the field of NFI. Current workflows as described in Section 3 are now error-prone, with lower success rate of intercepting relevant data and a more difficult analysis.

Other problems still exist in traditional NFI. Anonymized or encrypted communication hamper the investigation and might distract the analysis of the captured data. Tor [10] provides anonymity based on onion routing, I2P [46] uses a decentralized mix network that prevents the identification of the users [3]. analyses the traffic travelling through the Tor by using deep packet inspection techniques. The increasing amount of encrypted data prevents the analysis of the transmitted data, only meta information like ip-addresses, port numbers or timestamps are accessible. Different researchers, e.g. [23] discuss the analysing of encrypted network packets.

The processing of encrypted or anonymized network communication in virtual networks does not differ significantly from traditional networks, therefore current techniques of capturing, recording and analysing are applicable in virtual environments too.

We divide the arising issues in three sections, the classification guides on the phases of the network investigation. Problems that effect the online phase of the investigation are summarized in the section *online*, and problems that affect the analysing phase are summarized in the section *offline*. Problems that do not touch any of these phases depend on the encountered environment, which is typically designed by the CSP, hence these problems are classified as *organizational*.

The challenges are classified in three different categories describing the capability of aborting or impeding the NFI.

5.1 Online

The online phase covers the capture and the record process of NFI. In these steps the network data is identified, copied and stored on separate devices for the subsequent analysis. The afterwards defined problems have repercussions on capturing or recording of the network packets and are therefore classified as *online*.

5.1.1 Virtual NIC

DCs in the past without virtualization techniques provided the requested services on physical servers. In case of digital investigations the suspicious system has to be identified to demount the hard disks, perform memory forensics or capture the network traffic. While each server only hosts one or two services stationary without automatically changing this hosting, the whole environment is still rigid. Thereby the identification of a relevant server is easy and thus, the identification of the correct network card was easy to realize. The dogma of “one server – one port” was valid in this kind of implementation.

The development of high speed connections and the demand for high availability networks lead to a softer use of this dogma. E. g. link-aggregation provides the combination of two or more links acting as one logical interface to accelerate the connection or increase the availability. A NFI has to adapt only the capture process to get all transmitted packets whichever interface is used.

The growing of virtual networks breaks up with this dogma. Current hosting server provide more and more VMs, each of them at least connected to a virtual network. The connection to a virtual network is established by a virtual network interface card (vNIC) which is nearly congruent with a physical network interface card (pNIC) on hardware systems. Each vNIC is equipped with a mac-address and an ip-address which are valid inside the virtual network. The vNICs distinguish by name and internal id, which are assigned by the cloud controller (CC) or hypervisor.

All incoming or outgoing packets of all locally hosted VMs to external networks are consolidated with a few pNICs. Thereby the capture of the relevant data on a pNIC is not promising any more. An installed capture process connected with the related pNIC records lots of packets, which belong to different customers. The capture of this data may violate fundamental rights including privacy and data protection. The well-tried hardware solutions like a TAP or a bridge are not applicable any more with the vNIC of the suspicious VM. The lack of a usable pNIC, which only transmits traffic of the suspected VM impede a hardware solution.

5.1.2 Duplicate Mac addresses

Mac addresses are designed as a unique identifier for the NIC, and are used for the communication in ethernet based networks. The mac-address is divided in two parts, the first 24 bits represent the vendor, the last 24 bits are a contiguous value of the production process of the vendor. Thus, a mac address is a common filter criterion in NFI to reduce the amount of data and extract relevant information faster.

The use of hypervisors like *KVM* or *Xen* allows an administrator to select the mac address of a given VM. By separating the VMs into isolated networks, the mac address does not have to be unique any more in the virtual environment. The implementation of new network protocols as discussed prior amplify the ambiguous use of mac addresses. Protocols like VXLAN get rid of this restriction and demand the uniqueness of the mac address limited inside the VXLAN-segment.

This impedes the NFI in virtual networks. Filtering the captured data for mac addresses is not suitable any more. The captured data might contain, depending of the position of the capture implementation in the network infrastructure, the same mac address more than one time, each assigned to different hosts, which renders the identification of the suspected system impossible.

5.1.3 Overlapping IP addresses

In most cases NFI starts with a given IP address, discovered in logfiles or by analysing the communication of other suspects. This IP address is normally a public IP address, pointing to a provider, that has assigned the address to his customer. In virtual environments the CSP has to record this mapping of public IP addresses to the local ones. The result of the conversion leads to a customer and the assigned private subnet. A further capture process based on this local subnet is not suitable, because of the ambiguous and overlapping use of the internal IP addresses [37]. Overlapping network addresses are used to improve the flexibility in assigning the IP addresses dynamically [6].

If the provider implements overlapping ip addresses in the environment, he has to provide additional systems which translate the local ip addresses in unique addresses and reverse. Network Address Translation (NAT) is used for packets sent to external systems, the CSP uses a pool of so-called floating IP addresses to provide their customers the access to public IP addresses.

But the use of overlapping ip addresses constrains the internal addressing scheme, too. The CSP has to implement additional services to isolate these overlapping subnets. VMs of different customers, which inadvertently use the same IP addresses in their subnets, have to be separated and insulated in their local environment. Hence the CSP uses VLAN or network protocols like VXLAN, NVGRE or GENEVE which encapsulate the internal data and transfer it within a virtual tunnel to the communication partner. This protocols eradicate the limitations of VLAN and offer related techniques to tunnel network data and hide internal addresses schemes.

The identification of these suspicious systems is difficult without any further information. The knowledge about the used internal private IP addresses is not sufficient any more.

5.1.4 Internal traffic

The kind of captured data depends heavily on the correct capture position in the network. A capture process at the uplink of the network records all outgoing network traffic from the suspicious system, but the number of captured network packets is enormous, which led to an exhausting filtering and cumbersome analysis. The use of the pNIC of the hosting server records less traffic than the aforementioned technique, but there is still a high number of network packets which are irrelevant for the investigation. Additionally this technique will fail if the suspicious VM is moved to another hosting server as described later in Section 5.3.3.

None of the two positions of capturing ensure the entire communication data of the suspicious VM. In case of running two or more VMs of the same customer on the same Compute Node (CN), the network traffic between these VMs is hidden to the aforementioned positions. This traffic is transmitted inside the CN without using the pNIC or the uplink. Only traffic to VMs, that are located outside the CN or those, which belong to other customers is transmitted by the pNIC of the CN and is recorded by the capture process. Thus the analysis of network traffic being transmitted or received by these systems is essential, but will fail by using the wrong capture position.

5.2 Offline

The offline phase defines all necessary steps for analysing the captured data and extracting the relevant information. Problems in this section effect the analysis, either by complicating or by preventing the examination. These problems are software-related, which means, that they are independent from deployed hardware or the organizational structure of the environment. Nevertheless they are not independent from the real implementation in the cloud environment.

5.2.1 Software support

The amount of network packets transmitted in modern networks is enormous, hence the NFI has to handle this quantity of information. In virtual networks, a limitation of the captured traffic is difficult due to the dynamic and flexible environment. The definition of suitable filter is hardly possible without the risk of loosing forensic. So the analysis of this data is time-consuming and impossible without software tools and automated processes.

The market of network forensic software is exhausting, from short scripts to complex software solutions. Some of these tools are open-source, others are commercial. [16, 36] present an overview of different tools used in NFI.

We analysed different network forensic tools and their capability to handle new network protocols and huge input files. These aspects provoke the main issues of network traffic analysis of data captured in virtual networks. We performed predefined communication between two VMs, which ensures the transmitted data and enables the further valid comparison of transmitted and the extracted and analysed data. The testbed contains two physical hosts, running KVM as the used hypervisor. Each hypervisor provides the environment for the running VMs, including the connection to the local vswitch. The configuration of the vswitch was performed by an OpenFlow controller, which inserts the rules to the vswitch. The communication between the two VMs covers the transfer of two files with FTP and a client server communication via HTTP. The hash of each file was calculated for the further comparison with the extracted data. The communication was performed inside a VXLAN tunnel and over IPv6. Additionally we captured the data transmitted between the OpenFlow controller and the vswitch to determine the capability of analysing the OpenFlow protocol.

Depending of the location of the capture process, the amount of captured data grows extremely fast. Installed processes, that capture the data at the uplink port have to store more network packets than a process, that captures the data near the suspicious target system. The afterwards listed tools diversify by the speed of analysing the import files.

We assumed that an import of files with at least 1 GB taking more than 45 seconds is not suitable for NFI. We performed the import on an specialized investigation PC with an Intel Core i7 cpu, 32 GB RAM and a 250 GB SSD installed with Debian 8. The speed of analysing the imported data depends heavily on the hardware resources of the investigation system, with more available resources the processing of the data finishes quicker. However, the resources are typically finite, which merely delays the limit. Using cluster computing and distributed storage systems performance of data analysis might be further increased in the future. Currently, not all tools support parallel or distributed processing. Table 2 lists a short overview of the results[5].

The analysed tools were *Wireshark 2.2.0* [8], *Bro 2.4.1* [34], *ipsumdump 1.85*[6] or *Moloch 0.15.1* [45] as open-source or *Network Miner 2.1 beta*[7] as commercial tools.

Only *Wireshark* is capable to analyse the novel protocols. But it fails in analysing huge amounts of network traffic and big capture files in a timely manner. *Bro* or *Moloch* are able to handle big files, but they fail in analysing the newer network protocols. None of them is able to extract the transferred data of VXLAN and to figure out the transmitted internal payload. In addition to this, neither *Bro* nor *Moloch* are able to recognise the OpenFlow protocol. *Network Miner* in the current beta version is now capable to analyse the VXLAN and the OpenFlow protocol, but is not able to analyse huge capture files nor to use clustering.

Thus no software tool is capable to handle all requirements of a NFI in a virtual network. Of course it is possible to implement the needed features

Table 2 Capabilities of network forensic software

Software	IPv6	VXLAN	OpenFlow	Import of Huge Data	Clustering
Wireshark	*	*	*	–	–
Moloch	–	–	–	*	*
Network Miner	*	*	*	–	–
Bro	*	–	*	*	*
ipsumdump	–	–	–	*	–

[5] A ‘*’ marks success.

[6] http://www.read.seas.harvard.edu/kohler/ipsumdump/

[7] Details can be found at *http://www.netresec.com/?page=Networkminer*

for the tools, but this research demonstrates the missing flexible evolution of software tools. But the absence of eligible software, which supports the digital investigation in cloud environments impedes this examination.

5.2.2 Protocols

New protocols like VXLAN, GENEVE or STT offer new possibilities, expand the flexibility and are the basis for a virtual network. The main purpose of these protocols is to get rid of limitations of older protocols like IEEE 802.1q VLAN or Cisco ISL.

The usage of these protocols does not complicate the online phase of a network investigation. The capture and the recording of the network data is independent from the internally used protocols. However the position of the installed capture process influences the type and amount of recorded network packets. A capture process running nearly the suspicious target system reduces the amount of irrelevant network traffic, but is more difficult to implement, as described in Section 5.1.1. An installed capture process near an uplink is easier to implement, but increases the amount of irrelevant data. A trade-off between this two positions might reduce the amount of data, but provides more encapsulated network data, which transmits information between the different VMs.

The encapsulation complicates the further analysis of the captured data. As described prior the lack of suitable software tools to analyse the captured data is still an existing problem.

To extract the transferred information the software has to decapsulate each protocol information by itself, without losing any information transmitted on this layer. Depending on the rate of encapsulation, different network information are extracted. Without deeper knowledge of the used virtual infrastructure, the decision which IP- or mac-addresses are relevant is impossible. Depending on the capture position in the network infrastructure either the addresses of the outer or the inner header are relevant, the extraction and filtering of internal addresses requires a decapsulation of all outer header.

5.3 Organizational

We define organizational problems as independent of the used hardware or internal protocols. Problems in this section depend on the inherent behaviour of virtual networks used to interconnect a great amount of VMs provided to different customers.

5.3.1 Multitenancy

The virtual environment enables a highly dynamic and scalable provision of VMs to lots of customers at the same time. Current servers are able to host more than 100 running VMs concurrently, administrated by an other customer. These different VMs share the same hardware components like hard disks, memory or network cards. Sharing those relevant components still complicates the digital investigation, especially the extraction of the relevant user data is hardly feasible [47].

The transferred network data of the hosting server contains communication data of all running VMs, possible internal system data or backup and imaging information, depending on the configuration of the underlay network. A capture process may obtain lots of odd data without further arrangements.

Whereas the recording of the system and backup data delivers unneeded network packets to the capture file, the storage of non-involved user data is mostly incompatible with the local legislation.

In addition to this the storage of unnecessary data leads to an overcrowded capture file, which complicates the further analysis. As described prior in Section 3.2 the use of a capture filter to decrease the amount of data by prevention of storing these data might be critical.

5.3.2 Multitude of controllers

In a SDN the controller is the central device, supervising the complete traffic control and all decision regarding the network. A multitude of controllers exists on the market, differentiated by the platform, the vendor, the application programmable interface (API) or the language support [22]. Our analysis of vendors, projects and implementations reveals 28 open source controller and 64 commercial SDN controller and vendors. This high number with different implementations reveals the unworkable multitude of controllers.

The current state of these controllers is extremely diverging, each implementations might be developed with an other model or purpose [40]. Table 3 lists 13 various controller, each with an exemplary distinctions that clarify the diversity.

This diversity of programming languages and runtime environments complicate the implementation of an ubiquitous approach, which is valid for NFI. Either the developed solution has to be adapted to the current environment or a very abstract approach has to be implemented. The commercial controllers complicate an approach furthermore. Without the knowledge of communication protocols or an existing API every digital investigation is impeded and will fail sooner or later.

Table 3 Comparison of SDN controller

Controller	Implementation	Vendor	Description
Beacon	Java	open source	multi-threaded
Brocade VCS Fabric	Java	Commercial	based on OpenDayLight
Cisco XNC	closed source	Commercial	based on OpenDayLight
Floodlight	Java	open source	multi-threaded
HP VAN	closed source	Commercial	–
Maestro	Java	open source	multi-threaded
NOX	C++	open source	multi-threaded
Nuage VSC	closed source	Commercial	use of MP-BGP
OpenDayLight	Java	open source	Basis for different controller
OpenMuL	C	open source	multi-threaded
Pox	Python	open source	single-threaded
Ryu	Python	open source	using gevent-wrapper
VMWare NSX	closed source	Commercial	Includes Layer2, Layer3 and Layer4 services

5.3.3 Migration

VMs are hosted on physical hosts, which have a high capacity of performance. In DCs lots of physical server host the different VMs, which are under control of a cloud-management platform (CMP) like OpenStack, OpenNebula or CloudStack. All of these implementations use cloud controller (CC) a central instance to control, monitor and manage the VMs. CNs provide the runtime environment of the VMs with locally installed hypervisors.

The load in a virtual environment is highly dynamic, with VMs consuming less cpu or storage capacity and others with a higher consumption rate. To guarantee an average system load, the CC monitors in combination with the CN the available resource pool to recognize peaks or increasing system loads [32]. If the load of a CN reaches a critical point, the CC is able to move particular VMs to an other, less busy CN. Not only the CC, even the customer is able to activate such a migration.

The movement of systems exists in webhosting solutions since the early 2000s, when the era of commercial webhosting was beginning. At these days a webserver hosted a lot of different domains under only one IP address. If the customer ordered an other hosting solution, like more free-space or more email addresses, the provider moves the relevant data to another host and changes the current network information like IP address or firewall policy. An installed capture process geared towards this customer host could be changed at the same time. The downtime during the manual migration enables the simultaneous reconfiguration of the capture process. The possibility to restart the suspicious server after creating the capture process ensures a complete capture file and a simplified analysis.

Nowadays the migration of a VM is performed without any additional processes inside the virtual environment. There are two main techniques to migrate the running VM, which are classified as postcopy or precopy migration [7]. The kind of migration is preconfigured by the CSP and the used CC.

Independent of the implementation, a migration of the suspected VM results in a movement from one physical host to another and the use of new network connections, involved pNICs and vNICs simultaneously. A migration affects either a hosting server inside the same server rack, another server rack inside the same DC or in another DC. The election of the new CN is largely unpredictable, therefore each hosting system is able to provide VMs of varying customers. A prior implementation of capture processes running on all used physical hosts to ensure the ongoing record of the transmitted network packets is naturally hardly to install. Additionally the flexible deployment of the VMs does not facilitate the predicting of the affected server. Therefore the migration of VMs impedes the ongoing capture of network data and exacerbate the capture process as well as the analysis.

Each of the new target systems needs a reconfiguration of the capture process depending on the new parameters which are valid after restarting the suspicious VM. The time to reconfigure the new capture process is critical, the longer the configuration takes, the longer the capture process stays invalid and transmitted data will not be captured. This loss of possible packets make the further analysis complex, error-prone and may result in a wrong conclusion.

A possible host-based capture solution seems to be suitable to record the packets, which are transferred while the VM is running on this host. This approach can be implemented by using proven hardware solutions like TAPs or SPANs on the involved switches observing the limitation of internal used traffic as described in Section 5.1.4. In case of a migration of the suspicious VM it is necessary to manually move the installed hardware to the new CN, which will fail in highly dynamic environments. An installed capture process inside the hosting server cause a change of the running environment and is not convenient for forensic investigations.

The migration of the target VM between different cloud environments is another critical part. So called hybrid clouds mix the the use of private cloud and public clouds managed by a third-party. A migration between two different datacenters increases the complexity of following the VM.

5.3.4 Customizable environment

One of the main purposes of virtual networks is to decrease the amount of administrative work and increase the dynamic in the network. The virtual networks provided to the customer relieve the provider of maintaining the user network. The CSP has only to guarantee a working underlay network.

The virtual network of a particular user is maintained by himself, which includes all aspects like addressing, internal routing, firewall policies or an implementation of VPN.

A NFI of a suspicious VM is getting more and more complex depending on the customization of the network. A running capture process might fail, if the user implements a new routing policy, activate a separate VPN connection or just migrates a particular VM to another hosting server.

Since containers do not migrate from one host to another, but are started and stopped by the user, the capture process fails and needs to be reconfigured every time the container is started again.

5.4 Severity

The aforementioned challenges impede different phases of digital investigation. Though the challenges are not equally important and differentiate by the capability to abort the digital investigation. Some of the issues only complicate the necessary processing, but might not abort the specific investigation. Other challenges are more critical and will abort the NFI certainly without solving the assigned problem.

In this section we classify the challenges in three categories according to their potential of impeding the NFI.

- **high**
  Challenges marked as *high* influence the NFI eminently. These issues abort the capture, record or the analysis process, therefore a solution is needed.
- **medium**
  Issues marked as *medium* might abort the NFI in virtual networks. The failure of the digital investigation depends on the given circumstances and must not occur.

- **low**
  Challenges marked as *low* complicate the NFI, but might not abort the whole process of capturing, recording and analysing. Without a solution, the NFI is more time-consuming or has to be performed without additional support, but an evasion is possible.

Table 4 defines the severity of the aforementioned challenges grouped by the phases.

The most critical problems are *migration*, *customization*, *internal traffic* and *vNIC*, so a definite process is necessary to eradicate this challenges and ensure a successful and valid NFI. *Duplicate MAC addresses*, *overlapping IP addresses*, *software support* and the *multitude of controllers* might abort the NFI, but with manual rework a time-consuming NFI is possible, e.g. the high amount of different controller implementations require a specific development referring to the given installation. Whereas the development referring to an open source controller might be possible, the process of extracting information out of a commercial and closed source controller might fail. In that case the investigation might break down because of the loss of possible data. Challenges like *novel protocols* and the *multitenancy* might hamper the the investigation, but a subsequent filtering of the stored data might reduce the irrelevant data. The encapsulated network traffic transmitted inside the novel protocols might be extracted by a manual deletion of the unwanted protocol headers and the extraction of the relevant information out of the network packet.

6 Basic Conditions

The aforementioned discussed problems affect all three parts of NFI. To provide a valid, ongoing and accurate capture file, which ensures a purposeful analysis, new basic conditions gets relevant. These basic conditions

Table 4 Severity of challenges

Classification	Challenge	Severity
Online	vNIC	high
	MAC address	medium
	IP address	medium
	Internal traffic	high
Offline	Software support	medium
	Protocols	low
Organizational	Multitenancy	low
	Multitude of controller	medium
	Migration	high
	Customization	high

define important aspects, that have to be realized in order to eradicate the issues of NFI in virtual networks.

6.1 Data Reduction

The capture file should contain only network traffic related to the suspicious VM without any additional data of other VMs, system information, backup data or management network packets. The reduced amount of captured network traffic improves the subsequent analysis by providing an expeditiously parsing, extracting and processing process. The limitation of captured data simplifies the recording, which has to store now only relevant network packets without additional overhead.

6.2 Flexibility

The migration of the target VM might crash the running capture process. The capture process has to be dynamic in order to react referring to all critical changes in the network. These changes are migration and user customization, which impede the capture process. By providing a flexible and autonomous environment, which implements methods to change the installed capture process as fast as the migration process occurs, the most critical changes in the network are mitigated. The flexibility has not only limited to the migration of a VM, changes inside the logical user network have to be handled in an appropriate way.

6.3 Vendor Independence

The multitude of implementations and vendor specific solution impede the development of valid tools to perform NFI in a SDN. By implementing a vendor independent solution the multitude of controller gets irrelevant. This independence facilitates the use of proven methods, which simplifies the development of valid and reliable tools, that might work in different virtual networks without demanding changes or adapted installations of the workflow.

6.4 Local Jurisdiction

A NFI captures all network data of a given target. This network traffic contains password, user information, names, email addresses and files like pictures, videos or documents. The recording of this network traffic makes an extraction of this transmitted data possible. This extraction is wanted in digital investigation to examine cyber crimes, but the analysis has to be performed in full observation of the legislation. If the NFI is performed under full observance of law, only relevant network traffic is captured without additional overhead. Associated with the local jurisdiction the aforementioned data reduction pursue the same target.

6.5 Hardware Independence

The virtual environments provide less pNICs, which are usable with current hardware based techniques like TAPs or bridges. The inherent use of vNICs hamper the use of this techniques, hence the hardware based techniques get irrelevant and might be omitted. To realize a valid capture process, a software based approach is necessary, which implements hardware independent capture techniques. Therefore techniques like mirror ports gain in importance, they operate in combination with the used network devices. By using the virtual switches running in the network, the implementation of mirror ports provided by these devices are necessary to ensure the valid NFI in virtual networks.

6.6 Monitoring

The postulated flexibility is not sufficient to implement a valid and ongoing capture process in virtual networks. To realize an autonomous NFI in these networks the monitoring of arising changes gets necessary. Without an adequate monitoring, the surveillance of the suspicious target VM is an error-prone and arduous process, which might fail after some changes inside the virtual network. The monitoring of a target VM by using a distinct identifier enables an autonomous and self-acting process of tracking the VM inside the virtual network.

6.7 Inference

The arising problems of NFI in virtual networks prevent the use of proven methods and techniques. In the aforementioned section we listed different problems and derive a classification in three parts of these problems. In this section we defined six different basic conditions, each of them defined to eliminate the arising problems.

Table 5 summarizes the basic conditions and the assigned solved problems.

Table 5 Challenges affected by the basic conditions

	Challenge									
Conditions	vNIC	MAC Address	IP Address	Internal Traffic	Software support	Protocols	Multitenancy	Multitenancy of controller	Migration	Customization
Data reduction	–	–	–	–	*	–	*	–	–	–
Flexibility	–	–	–	–	–	–	–	–	*	*
Vendor independence	–	–	–	–	–	–	–	–	*	–
Local jurisdiction	–	–	–	–	–	–	–	*	–	–
Hardware independence	*	–	–	–	–	–	–	–	–	–
Capturing before tunneling	–	*	*	*	*	*	*	–	–	–
Monitoring	–	*	*	–	–	–	–	–	*	*

The defined basic conditions eradicate the new arising problems of NFI in virtual networks, a possible implementation, that considers these conditions is able to guarantee a valid NFI in virtual networks. However the basic conditions do not describe a real implementation, but provide a possible guideline for the new development of techniques and tools for NFI in virtual networks.

We assume the high flexibility and the ensuing migration and user customization as the most critical problems for a valid capture process. Therefore the tasks *Monitoring* and *Flexibility* are the most important conditions to implement. Without these two tasks the NFI will fail after a change in the network, a manual reconfiguration is too slow and will result in an incomplete packet capture file. To obtain a packet capture file which consists of all network traffic, the tasks *Capture before tunneling* and *Hardware independence* have to be implemented. Without these two tasks, the internal network traffic might not traverse the installed capture process and stay hidden to the investigator.

The other conditions improve the NFI in virtual networks, but are not crucial for a sufficient investigation. The *vendor independence* simplifies NFI in different networks by providing a common implementation, but an adapted solution for a given environment might satisfy the requirements in this special network infrastructure. The data reduction improves the subsequent analysis of the data, but a prior, but cumbersome and error-prone, manual data reduction with filters like the Berkeley Packet Filter

(BPF) [27] is also possible. Additionally this filtering might delete unwanted and irrelevant network packets out of the capture files to observe the local jurisdiction.

One possible implementation of a valid method following this basic conditions is the use of the vNIC of the target system and implement a monitoring and reconfiguration service, which is able to recognize critical changes in the network and perform the relevant reconfiguration of the capture process. If such a service is implemented in the virtual environment, the whole environment might be watched and the network traffic of the suspicious VM is permanently monitored.

7 Conclusion

With the increasing demand of flexible ICT environments, the use of virtual networks gains in importance. Concurrent to this development, there is an increasing emphasis on reliable digital examination methods, in particular the NFI.

In this paper, we have analysed the transition from physical to virtual networks from the perspective of forensics and derived the arising problems of NFI in virtual networks. We separated this problems in three parts namely *online*, *offline* and *organizational* and classify their potential of aborting the NFI. Based on this separation we derive six basic conditions, which are necessary to implement for a valid NFI in virtual networks. This chapter summarizes the challenges and their classification, subsequent we present our future research and the development of ForCon as an implementation for NFI in OpenFlow networks.

7.1 Summary

The evolution of network virtualization led to a highly dynamic and flexible infrastructure offering two views on the network. The underlying physical network is the basis for lots of different overlay networks. These overlay networks provide a logical implementation of independent virtual networks. The administration of these virtual networks is separated from the administration of the underlay network and may be performed by the customer. Each virtual network is isolated from others and might be customized without additional work of the provider.

A higher rate of abstraction is reached with SDN and NFV. SDN implements a central control of the network and provides the ability to program the network infrastructure. This led to a highly flexible environment, which may act mostly autonomous without any further interaction of the network administrator.

NFV transforms given network services into virtual appliances to improve the flexibility in the network by providing the needed services on demand.

The digital investigation in such virtual networks is complex and error-prone. Aspects like migration of suspicious VMs or vNICs complicate the capture process, encapsulating protocols like VXLAN impede the further analysis. Different CMPs compound the development of an available approach.

Forensic investigation in a SDN is a complex task, the control of the network by the CC led to an mostly unpredictable behaviour of the network and a suspicious VM.

The deployment of NFV enables a customizable environment with high user influence of the installed virtual network infrastructure. The CSP is no longer concerned for any internal changes in the virtual network. An implementation of a new VPN by activating a separate virtual appliance in the local network might change the whole communication mode in the virtual network and abort the capture process.

Capturing the data at universal positions like uplinks affect local jurisdiction, because of the storing of network traffic assigned to uninvolved VMs. Additionally this capture position is unable to record internal traffic, which is transmitted inside the physical host.

The defined challenges differentiate in the potential to abort the NFI in virtual networks. We derive a classification of severity and separate the challenges in a *high*, *medium* and *low* risk of aborting the NFI. We define the challenges *migration*, *customization*, the capture of *internal traffic* and *vNIC* as the most critical issues and mark them therefore as *high* risky. An evasion of this issues is not sufficient, to ensure a successful NFI in virtual networks a specific solution referred to this three challenges is necessary.

Based on this issues six requirements were defined ensuring NFI in virtual networks. They affect all aforementioned challenges and eradicate the different issues. The most critical issue for NFI in virtual networks are the high flexibility implemented by the migration and the user customization. Therefore the requirements of *Flexibility* and *Monitoring* gain in importance, aspects like *Hardware independence* and *Capture before tunneling* ensure the valid network packet capture of the complete traffic of a suspicious target VM.

7.2 Future Work

Digital investigation in virtual networks is still a not fully researched, rather a largely uncharted topic in digital forensic.

Network virtualization makes the network forensic process inside a cloud environment more complex, because none of the traditional techniques seems to be successful. VNICs impede the data recording, new network protocols like VXLAN complicate the analysis.

A promising approach is to implement a capture process on the target vNIC. But as discussed in this paper each operational solution might fail after a period of time. Shutting down and restarting the VM changes the internal identifier of the vNIC, which aborts the running capture process. Migrations or customized environments abort a running capture process, too. So an additional monitoring process is needed to recognize each change in the controlled network infrastructure.

Our focus in future work is to create a process of capturing network data in virtual environments, including all network data arising of a suspicious VM, even when this VM is stopped or migrated, or the virtual environment is changed by the customer. A first implementation of a tool named *ForCon* acts as a proof-of-concept in OpenFlow based networks. ForCon uses dislocated agents running on the physical hosts to identify the suspicious VM and to implement a capture process by using the vNICs inside the virtual network. The agents extract the information used from the switches to forward the network packets and transmit them to the ForCon server. This server investigates the transmitted flow information and identifies flows assigned to the suspicious VM. The involved switch is informed by the server to manipulate all relevant flows in order to add a additional output port to the flow. This additional destination implements the copy of all affected network packets. After manipulating the flows the switch activates a monitoring task to react on critical changes. Changes like the migration or the customization cause a change of the supervised vNIC and might require a reconfiguration of the running capture process. So the monitoring agent informs the ForCon server, which analyse the change and updates the stored information. This can lead to a reconfiguration of the capture process on the same switch or the command to update the extracted flows on all agents to identify the suspicious VM again.

The used agents in this proof-of-concept interact with OVS instances, but the implementation of a specific communication protocol named *ForCon Protocol* (FCP) makes the development of agents running on other installations possible. FCP defines different message types transmitted between server and agents to extract, manipulate and delete flows, exchange information, establish tunnel connections and gather data of installed switches. By using this defined message types the communication of different agents and the server expands the capabilities of ForCon.

References

[1] Davie, B. and Gross, J. (2016). *A Stateless Transport Tunneling Protocol for Network Virtualization*. Internet Draft, Informational: New York, NY.

[2] Faizul, M., Boutaba, B. R., Esteves, R., Granville, L. Z., Podlesny, M. D., Rabbani, G., et al. (2013). Data center network virtualization: A survey. *Commun. Surv. Tut. IEEE* 15, 909–928.

[3] Chaabane, A., Manils, P., and Ali Kaafar, M. (2010). Digging into anonymous traffic: A deep analysis of the tor anonymizing network. In *Proceedings of the Network and System Security (NSS), 2010 4th International Conference on* IEEE 167–174.

[4] Mosharaf Kabir Chowdhury, N. M., and Boutaba, R. (2009). Network virtualization: state of the art and research challenges. *IEEE Commun. Mag.* 47, 20–26.

[5] Mosharaf Kabir Chowdhury, N. M., and Boutaba, R. (2010). A survey of network virtualization. *Comput. Netw.* 54, 862–876.

[6] Cisco Systems Inc. (2016). *Ip Overlapping Address Pools*. Available at: http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipaddr_ipv4/configuration/xe-3s/asrl000/IPv4-xe-3s-asrl000-book/IP-overlap-addr-pools.pdf [accessed by November 03, 2016].

[7] Clark, C., Fraser, K., Hand, S., Hansen, J. G., Jul, E., Limpach, C., et al. (2005). Live migration of virtual machines. In *Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation*, Berkeley, CA, 2273–2286.

[8] Combs, G., et al. (2016). *Wireshark*. Available at: http://www.wireshark.org/

[9] Corey, V., Peterman, C., Shearin, S., Greenberg, M. S., and Van Bokkelen, J. (2002). Network forensics analysis. *IEEE Int. Comput.* 6, 60–66.

[10] Dingledine, R., Mathewson, N., and Syverson, P. (2004). *Tor: The Second-Generation Onion Router*. Technical Report, DTIC Document.

[11] Dua, A., Raja, A. R., and Kakadia, D. (2014). Virtualization vs containerization to support paas. in *Proceedings of the Cloud Engineering (IC2E), 2014 IEEE International Conference on* IEEE, 610–614.

[12] Dykstra, J., and Sherman, A. T. (2013). Design and implementation of frost: Digital forensic tools for the openstack cloud computing platform. *Digital Investig.* 10, S87–S95.

[13] ETSI (2012). *Network Functions Virtualisation – Introductory White Paper*. *SDN and OpenFlow World Congress*, 1.

[14] Gross, J., Sridhar, T., Garg, P., Wright, C., and Ganga, I. (2016). Geneve: Generic network virtualization encapsulation. Internet-Draft.

[15] Han, B., Gopalakrishnan, V., Ji, L., and Lee, S. (2015). Network function virtualization: challenges and opportunities for innovations. *IEEE Commun. Mag.* 53(2), 90–97.

[16] Hunt, R., and Zeadally, S. (2012). Network forensics: an analysis of techniques, tools, and trends. *IEEE Comput.* 45, 36–43.

[17] IEEE Computer Society (2005). *Virtual Bridged Local Area Networks Amendment 4: Provider Bridges*. Technical Report, IEEE Standard for Local and metropolitan area networks.

[18] Jain, R., and Paul, S. (2013). Network virtualization and software defined networking for cloud computing: a survey. *IEEE Commun. Mag.* 51, 24–31.

[19] Jarschel, M., Zinner, T., Hoßfeld, T., Tran-Gia, P., and Kellerer, W. (2014). Interfaces, attributes, and use cases: a compass for sdn. *IEEE Commun. Mag.* 52, 210–217.

[20] Delgadillo, K. (2015). Netflow services and applications. *Cisco Whitepaper*, 1996 (accessed November 5, 2015).

[21] Khan, S., Gani, A., Abdul Wahab, A. W., Shiraz, M., and Ahmad, I. (2015). Network forensics: review, taxonomy, and open challenges. *J. Netw. Comput. Appl.* 66, 214–235.

[22] Khondoker, R, Zaalouk, A., Marx, R., and Bayarou, K. (2014). "Feature-based comparison and selection of software defined networking (sdn) controllers," in *Computer Applications and Information Systems (WCCAIS), 2014 World Congress* (Rome: IEEE), 1–7.

[23] Koch, R. (2011). *Systemarchitektur zur Ein-und Ausbruchserkennung in verschluüsselten Umgebungen*. Ph.D. thesis, Universit at der Bundeswehr Mu nchen, Neubiberg.

[24] Kreutz, D., Ramos, F. M. V., Verissimo, P. E. Rothen-berg, C. E., Azodolmolky, S., and Uhlig, S. (2015). Software-defined networking: a comprehensive survey. *Proc. IEEE*, 103, 14–76.

[25] Lazzez, A. (2013). A survey about network forensics tools. *Int. J. Comput. Inf. Technol.* 2, 74–81.

[26] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T. et al. (2014). *Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks. RFC 7348 (Informational).*

[27] McCanne, S., and Jacobson, V. (1993). "The bsd packet filter: A new architecture for user-level packet capture," in *USENIX Winter* (Berkeley, CA: USENIX Association), 259–270.

[28] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L. (2008). Jennifer Rexford, Scott Shenker, and Jonathan Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM Comput. Commun. Rev.* 38, 69–74.

[29] Meghanathan, N., Allam, S. R., and Moore, L. A. (2009). Tools and techniques for network forensics. *Int. J. Netw. Secur. Appl.* 1.

[30] Mell, P., and Grance, T. (2011). The nist definition of cloud computing. *Natl. Inst. Stand. Technol.* 800–145.

[31] Merkel, D. (2014). Docker: lightweight linux containers for consistent development and deployment. *Linux J.* 2014:2.

[32] Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., and Zagorodnov, D. (2009). "The eucalyptus open-source cloud-computing system," in *Proceedings of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, 2009. CCGRID'09,* (Rome: IEEE), 124–131.

[33] Palmer, G., and Corporation, M. (2001). *A Road Map for Digital Forensic Research*. Technical Report 1. Digital Forensic Research Workshop, Utica, NY.

[34] Paxson, V. (1999). Bro: a system for detecting network intruders in real-time. *Comput. Netw*. 31, 2435–2463.

[35] Pfaff, B., Pettit, J., Koponen, T., Jackson, E., Zhou, A., Rajahalme, J., et al. (2015). "The design and implementation of open vswitch," in *Proceedings of the 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, 117–130.

[36] Pilli, E. S., Joshi, R. C., and Niyogi, R. (2010). Network forensic frameworks: survey and research challenges. *Digit. Invest*. 7, 14–27.

[37] Rekhter, Y., Moskowitz, B., Karrenberg, D. G., de Groot, J., and Lear, E. (1996). *Address Allocation for Private Internets*. Technial Report RFC 1918.

[38] Ruan, K., and Carthy, J. (2012). "Cloud forensic maturity model," in *ICDF2C of Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Eds, K. Ruan and J. Carthy (Berlin: Springer), 22–41.

[39] Ruan, K., Carthy, J., Kechadi, T., and Crosbie, M. (2011). *Cloud Forensics: An Overview*. Dublin: Centre for Cybercrime Investigation.

[40] Shalimov, A., Zuikov, D., Zimarina, D., Pashkov, V., and Smeliansky, R. (2013). "Advanced study of sdn/openflow controllers," in *Proceedings of the 9th Central & Eastern European Software Engineering Conference* Russia, (New York, NY: ACM).

[41] Shanmugasundaram, K., Brönnimann, H., and Memon, N. (2006). "Integrating digital forensics in network infrastructures," in *Advances in Digital Forensics*, volume 194 of *IFIP — The International Federation for Information Processing* (Boston, MA: Kluwer Academic Publishers), 127–140.

[42] Spiekermann, D., Eggendorfer, T., and Keller, J. (2015). "Using network data to improve digital investigation in cloud computing environments," in *High Performance Computing & Simulation (HPCS), 2015 International Conference* (Rome: IEEE), 98–105.

[43] Stallings, W. (1993). *SNMP, SNMPv2, and CMIP: The Practical Guide to Network-Management Standards*. (Reading, MA: Addison-Wesley Pub. Co.,).

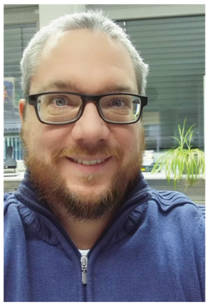
[44] Vilalta, R., Muñoz, R., Mayoral, A., Casellas, R., Martínez, R., López, V., and López, D. (2015). Transport network function virtualization. *J. Lightwave Technol.* 33, 1557–1564.

[45] Wick, A., and Miller, E. (2016). Moloch. Available at: http://molo.ch (accessed November 3, 2016).

[46] Zantout, B and Haraty, R. (2011). "I2p data communication system," in *Proceedings of ICN,* 401–409.

[47] Zawoad, S., and Hasan, R. (2013). Cloud forensics: a meta-study of challenges, approaches, and open problems. *CoRR*.

Biographies



D. Spiekermann received his B.Sc. degree in Computer Science in 2009 and his M.Sc. degree in Electronic and Computer Engineering from FernUniversität in Hagen, Germany in 2014. He is currently working towards his Ph.D. degree in Computer Science at FernUnivestiät Hagen, Germany. He works as a forensic investigator at North-Rhine Westphalia Police, Dortmund, Germany. His research interests are network forensics, virtual networks and it security.

T. Eggendorfer is a professor for IT security at Hochschule Ravensburg-Weingarten, before he was a professor for IT forensics in Hamburg. He received his Ph.D. on email security at FernUnivestiät Hagen in 2007. He holds lecturing positions at multiple universities. Besides his duties at the university, he is a freelance IT security and forensics specialist as well as a privacy advocate.