Hochschule Wismar

University of Applied Sciences: Technology, Business and Design

Faculty of Engineering, Department of Electrical Engineering and Computer Science

Master-Thesis

Development of Forensic Strategies and Methods in Software-Defined Networks

by Ing. Florian Weijers

[redacted]

to obtain the academic degree

Master of Engineering

Submitted: 27.05.2024

First Reviewer: Prof. Dr.-Ing. Meiko Jensen

Second Reviewer: Prof. Dr.-Ing. Antje Raab-Düsterhöft

Aufgabenstellung (deutsch)

Im Rahmen der vorzulegenden Arbeit soll der Aufbau und die Struktur eines modernen softwaredefinierten Netzwerkes (SDN) untersucht werden.

Dabei soll u.a. ein typischer Aufbau grafisch dargestellt werden.

Weiterhin ist auf die aktuellen technischen Feinheiten der gewählten Netzwerktechnologie einzugehen und mit aktuellen Methoden der technischen IT-Sicherheit und digitalen Forensik in Verbindung zu bringen.

Im Rahmen der Untersuchungen soll auf Problemstellungen in SDN eingegangen werden. Wie kann eine digitale Forensik in SDN methodisch und strukturiert ablaufen? Welche Grundsätze und Abläufe sind dabei zu beachten?

Schließlich sollen im Ergebnis die technischen Methoden und theoretischen Abläufe verallgemeinert werden. Das Ziel der Arbeit ist somit eine Art Handlungsempfehlung für die digitale Forensik in SDN zu entwickeln.

Task Definition (English)

This thesis should delve with the design and structure of a modern software-defined network (SDN) with the background of digital forensics.

Among other things, a typical graphical structure is to be presented.

Furthermore, the current technical subtleties of the chosen network technology are to be addressed and linked to current methods of technical IT security and digital forensics.

Problems in SDN should be addressed as part of the investigations. How can digital forensics in SDN be organised in a methodical and structured way? What principles and processes need to be observed?

Finally, the technical methods and theoretical processes are to be standardised. The aim of the work is thus to develop a kind of recommendation for action for digital forensics in SDN.

Abstract

This work shows the methodical and strategic approach of a forensic investigation in advanced SDN. First, the basics of SDN technology are explained in general terms. This is followed by a presentation of classic approaches to digital forensics in networks and a brief description of typical security mechanisms in SDN.

The basic technical and structural characteristics of SDN are then outlined. This concerns specific characteristics of SDN in contrast to other networks.

Typical forensic tools that are used for network investigations and that may also be suitable for investigations in SDN are also presented. The classic process of a forensic network investigation is also shown.

Furthermore, a network of the company ZeroTier Inc. is examined using an example of SDN and fundamentally analysed with selected tools. The main focus is on the applicability of typical investigation tools and the special features of the results in SDN. Basic network information is illustrated, as well as the functionalities and architecture of the ZeroTier network. Typical characteristics of the ZeroTier network are then worked out, and finally different tools are used to obtain information in a ZeroTier example network.

Strategies and methods for investigations in SDN are then derived from the practical investigations and presented graphically as a result.

The methods of network forensics in SDN and the strategies of the SDN investigation processes are illustrated and summarized in tabular form using developed graphics.

The summary also deals with the development of structured network forensics in SDN and the presentation of legal problems and conflicts, with the aim of establishing methodological and strategic standards in subsequent projects.

Content

Preface

“The only truly secure system is one that is powered off, cast in a block of concrete and sealed in a lead-lined room with armed guards - and even then I have my doubts.”

“Spaf” Eugene Howard Spafford - Computer Science Expert

This quote from the well-known Linux enthusiast and IT security expert Prof. Dr. Eugene Spafford from 1989 vividly illustrates the importance of the security of a computer system. There can be no safe computer system.

Therefore, there are certainly no secure computer networks, as these require at least two parties to exchange information with each other on switched-on computing devices.

This fact presents a technical and social dilemma when you consider that the majority of society uses computer networks and their security is suggested or assumed.

But how is security implemented in modern computer networks? How is insecurity determined? What are the traces that exist in current network structures, specifically software defined networks? How can forensic experts secure traces and document and present them in a clear way?

These and other questions arise for me as the author of this thesis and in my work as a forensic scientist and expert witness for customers, clients and courts.

I personally also lack technical and legal evidence for a standardised and targeted approach to professional forensics in software-defined networks.

This paper is therefore intended to discuss methodological and strategic possibilities for forensics in SDN.

Acknowledgement

I am not a good person - and not a bad one. With this in mind, I would like to thank my crazy digital forensics lab colleagues who have put up with me over the years. Wolfi, Sepp, Markus, Christian and the two Jürgens - it wouldn't have been funny without you!

I would also like to thank my wonderful wife Dajana and my lovely daughters, who have always looked after me and tolerated my crazy computer projects.

I love you all!

Finally, I would also like to thank the University of Wismar in the person of Prof. Dr. Raab-Düsterhöft, who has shaped my academic development with her support. Without the offers and options provided by the university, my further development in recent years would never have been possible.

1. Introduction

So what is this thesis all about?

In the course of the following elaboration, possible methodologies and strategies for obtaining information or for forensic work in software-defined networks will be developed from the already existing and practically tested methods and strategies of network forensics.

In the following, we will first look at the basic structure of software defined computer networks (SDN) as a relatively new and emerging technology. An illustrative and topical example was chosen for this, and the network structure of a popular SDN-solution from ZeroTier Inc. was analysed and described. This allows us to define and understand the basic structure and technical background of such a modern and advanced computer network. After describing the general SDN technology, we will have a closer look at the technical and structural conditions in an SDN and attempt to derive and model classic forensic investigation methods and strategies in SDN networks.

Since software-defined networks are also used by criminals, for example for botnets, criminal peer-to-peer sharing-networks or ransomware structures, forensic investigation tools will be presented under these aspects and individual examples will be illustrated.

Another point is the practical forensic analysis of a software-defined network. The general forensic approach is described, with a focus on the functionality and architecture of this network technology.

It is therefore essentially about the practical tools and features of forensic work in SDN environments to detect critical anomalies and to be able to analyse and consistently document the incidents.

Typical problems will also be discussed and at least shown for different use cases.

The derived results are then summarised and presented graphically in order to make the results of a forensic analysis of SDN understandable and usable in court, even for laypersons.

The final goal is to create a proposal and a first step for consistent strategic and methodological forensic work in SDN, in order to develop standard strategies and methodologies as we progress.

The technical and legal difficulties and conflicts are also addressed at the end of the work. The investigation in SDN raises both legal and ethical questions and concerns.

2. Basics

In order to develop forensic methods and strategies in SDN, it is firstly important to define some basic properties of the technical matter that should fundamentally be concerned. The basic properties of SDN technology are described and illustrated on the following pages.

Secondly, it is about the basics of forensic work in computer networks in order to show the applicable methods and strategies that are used by forensic experts in the last years.

In the following technical description, we do not want to go into every eventuality and subtlety of the different SDN implementations. Every software-controlled network has its individual technical characteristics, for example in data encapsulation and dynamic encryption. Different protocols are also used frequently, which cannot be presented conclusively, as the software changes at short notice and dynamic security mechanisms are adapted. Basically, it is not about technical forensics in detail, but rather about the methodology and strategies of network forensics in such dynamically developing network environments. These strategies and methods should be applicable to different network structures in the end and ultimately adapt to the needs of the forensic experts in their daily processes.

The basic background work on network forensics is the documents from ENISA (Introduction to Network Forensics, 2019), NIST (SP 800-86: Guide to Integrating Forensic Techniques into Incident Response, 2009) and BSI (Guideline IT Forensics, 2011) assumed to be state of the art. Another work on the state of the art in network forensics is Florian Schreiber's diploma thesis from October 2021 at the St. Pölten University of Applied Sciences. Mr. Schreiber's work in particular provides informative current references in the young field of network forensics.

2.1. SDN Technology

Software defined networks are widely used in today's networked applications. The focus is on software-dependent control of these networks and makes the technology almost hardware-independent.

SDNs do not have to be additionally physically configured in a complex manner, but are based on an existing hardware infrastructure.

The definitions and classifications of the German Chambers of Industry and Commerce shows that a clear definition of forensics in SDN has not yet been provided [De15]. The specialist areas of the information technology experts speak of hardware specialities in particular networks and telecommunications. This shows that there are currently no guidelines for expert witnessing in SDN in the practice of certified experts, especially in Germany.

The fact that SDN still plays a major role in the everyday work of network forensics will be explained in the following.

SDN are now used very frequently in practice:

- in data centres
- in cloud computing
- in remote work
- in WANs
- in private local networks
- in IoT
- in p2p Networks
- in many other network structures

But SDN networks are not only built and used in regular productive applications.

SDN network structures also play a major role in online games or in the wide area of computer crime.

In online games, for example, millions of players are regularly connected to one another via a gaming network (e.g. Palworld: over 2 million users at the same time in 2024, or on Steam generally over 30 million users in March 2024) [Sc24].

On the other side, as part of botnets, millions of infected computers (Mariposa botnet: 13 million computers) wait for feedback from a control server [Th10].

Based on these descriptions, you can see that SDNs are widespread and computers are connected even without any special knowledge of the users.

But how do such a network structures work technically?

This simple question requires a detailed answer and cannot be clearly defined across the board.

First, we look at the well-known OSI and TCP/IP stack of the network layers to locate the network structures.

OSI Model		TCP/IP Model		
7.	Application	4.	Application	
6.	Presentation			
5.	Session			SDN
4.	Transport	3.	Transport	
3.	Network	2.	Internet	
2.	Data	1.	Network Access	
1.	Physical			

Fig. 1: SDN located in the OSI Model

As can be seen in the figure above, the SDN network structures are not clearly defined in a specific OSI layer. SDN controllers can interact with the above layers and are sometimes located in layer 2 (data link layer).

SDN controllers can also be located in layer 3 in order to configure routing tables and firewall rules there. Still, SDN structures are usually in the area around Layer 4 and therefore have a direct influence on data transport.

In individual use cases, SDN via Layer 7 can also be used directly in applications (e.g. online games) to implement content delivery or web proxying.

Moving forward, it is therefore very important to understand that SDN can be active at different levels of the OSI model. In a technical sense, this abstraction of SDN enables a more flexible and scalable application of this network technology.

This means that security-critical SDN solutions can be implemented on Layer 3 with the current IPSec protocol. SDN on Layer 2 offers high compatibility with the PPTP, MPPE or L2TP protocols. High-performance SDNs often rely on Layer 4 and the SSL/TLS protocols. And individual applications use layer 7, for example the SSH protocol in Linux/Unix environments. In addition, protocols for authentication such as EAP and Radius play a role in setting up secure SDN structures such as the LLDP and DHCP protocols for configuring the SDN controllers. Furthermore, the OpenFlow, NetConf and BGP protocols are often used to control data traffic via IP or TCP/UDP. Other protocols can also be used and are often proprietary. These protocols then have a serious impact on the security, scalability and performance of the network.

As an interim conclusion, it can be said that SDN can be used independently and in parallel with other network applications and technologies. The different network structures are not accessible to other areas - or at least this should be the case - which can make forensics in these networks more difficult. We will come back to this later in the practical part.

Another important technical point is the control authority of the SDN. There are practically three options for network control. On the one hand, central control via a master node that manages all clients and with which the clients have to log in and authenticate. Secondly, there is a decentralised solution for SDN networks, where tables of members are stored on each client and clients authenticate each other by exchanging secrets. Or there is a hybrid solution in which there are different levels of authentication servers (roots or “moons”) responsible for network administration. All three structures have advantages and disadvantages, e.g. in reliability, data speed, scalability and user-friendliness [St22].

Depending on the SDN application, there are different methods of establishing a secure connection. The connection between the two communication partners is often set up as directly as possible via the underlying network, and only the encryption parameters are negotiated via the SDN connection. This allows individual powerful encryption, even for high security purposes.

We currently do not consider real-time decryption of this network traffic to be possible, although there are attack strategies and in the long term these encryptions can probably be broken.

Simply summarised, we find that SDN represents a dynamic, scalable and secure way to connect different clients to each other over a different underlying network.

For digital forensics, the aforementioned technical features mean new challenges and different approaches to traditional network forensics. The dynamics, scalability and security mechanisms in SDN alone must be taken into account in the forensic approach. The investigation processes must therefore be adapted and special strategies and methods must be developed.

2.2. Classic Methods and Strategies of Digital Forensics

Digital forensics is a special field of informational computer analysis and includes areas from different computer science directions.

Forensic scientists are trained in various areas, such as mobile forensics, computer forensics, incident response, car forensics, IoT or in the area of network forensics.

The following will show some basic principles[1] of digital forensics, which are mostly used in various special fields already mentioned [Bu11]:

- *Live forensics:* forensics on a living object, i.e. on a running computer system to secure volatile data
- *Post-mortem forensic:* forensic investigations on computer systems or hard drives that have already been switched off

This rough division already allows us to identify relevant problems in network forensics. A classic postmortem forensic examination in a computer network is almost impossible. Without network activity, connections, network participants and especially network traffic cannot be adequately examined. Only the protocol and log files can partially reconstruct network activities on a switched off network. At least the data collection must take place in a running network system and requires therefore a *hybrid forensic approach*.

Therefore, in the following, we will only briefly discuss post-mortem analyses in computer networks and assign this area to classic computer forensics and partially exclude it from the special investigation of SDN, without forgetting the importance of log analysis, deleted areas, etc.

Events that are currently occurring or have occurred in the recent past are of interest to a network forensic scientist. There must be direct access to the network and the relevant data must be analysed as quickly as possible.

[1] It is arguable that there are further fine gradations and other forensic principles as well. In the context of the practical investigation of SDN, the two processes of classic live forensics and post-mortem forensics play a central role.

The classic forensic workflows of computer forensics [Ca10] are now available as helpful and proven pillars of support in live incident response:

1. Preparation
2. Detection and Containment
3. Data Collection
4. Preservation
5. Examination
6. Analysis
7. Investigation
8. Presentation

Above all points and between them is the task of complete documentation with regard to the traceability of the investigations. Evidence can be very useful, conclusive and meaningful for the forensic scientist, but may not legally exist due to a lack of documentation.

It should therefore go without saying that all keyboard entries are recorded and, if necessary, a screencast is created on the examination computer and all logging mechanisms are used. In the best case, live examinations are carried out using the four-eye principle.

On the following pages we will relate the principles of forensic work mentioned above to network forensics.

The focus is on the methodology and strategies, without going too deeply into the individual technical details [Ma07] of individual use cases.

A *method* of forensic work is intended to represent a specific scientific approach that is used for network forensics.

A *strategy* is a method with a specific objective, i.e. the focus on the goal of a forensic investigation (e.g. solving the incident).

Various technical papers contain overviews of standard procedures for digital forensics in standard environments:

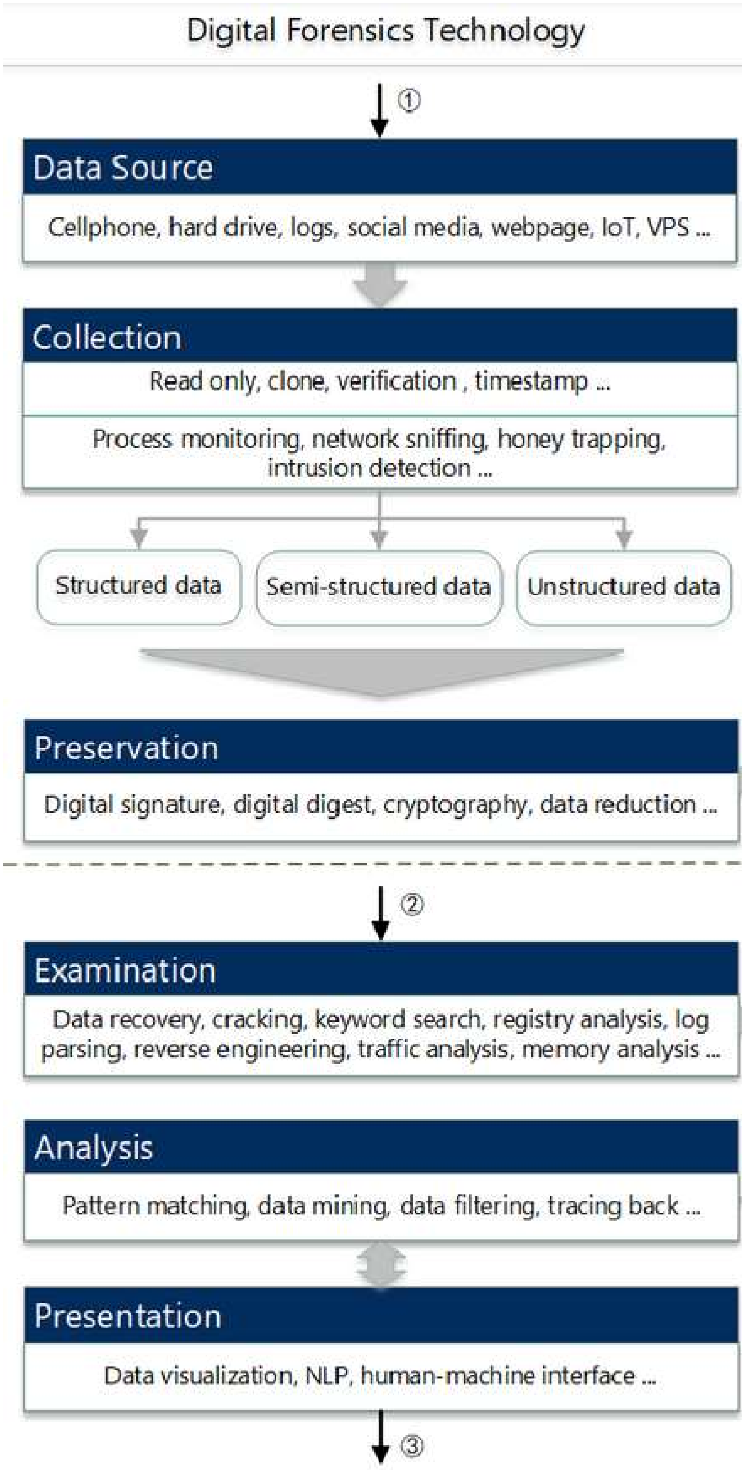

Digital Forensics Technology
①
Data Source
Cellphone, hard drive, logs, social media, webpage, IoT, VPS ...
Collection
Read only, clone, verification , timestamp ...
Process monitoring, network sniffing, honey trapping, intrusion detection ...
Structured data
Semi-structured data
Unstructured data
Preservation
Digital signature, digital digest, cryptography, data reduction ...
②
Examination
Data recovery, cracking, keyword search, registry analysis, log parsing, reverse engineering, traffic analysis, memory analysis ...
Analysis
Pattern matching, data mining, data filtering, tracing back ...
Presentation
Data visualization, NLP, human-machine interface ...
③

Fig. 2: Framework for Big Data Digital Forensic Investigation [So20]

The typical steps of the forensic process are explained in more detail on the following pages.

1. Preparation

The best possible preparation is an essential step for the network forensics investigation process. This serves not only to prepare for the current scenario, but also for constant adjustments to the conditions of evolving computer networks.

Methodology

- ➢ Development of a collection of tools for forensic purposes
- ➢ Clarification of possible legal authorities and permissions
- ➢ Provision of external resources (back office, cloud storage, etc.)

Strategy

- ➢ Adaptation of tools to the specific incident
- ➢ Prioritisation of technical tools for the purpose of the investigation
- ➢ Review of the legal environment and determination of possible scenarios
- ➢ The objectives of the investigations must be defined

As you can see here, different methods and strategies are possible, especially in the strategic preparation for the use of network forensics.

As a forensic investigation continues, these methods and strategies must be further adapted to the circumstances and modified if necessary. Preparation for different courses of a live examination therefore plays an important role. If, for example, legal authorities no longer apply, individual resources in the investigation process may no longer be accessed (e.g. sensitive personal data or data in protected foreign countries).

The preparation should cover these considerations and outline options for further action.

2. Detection and Containment

At the scene of the event, a focus must first be placed on what is essentially taking place. Which systems are actually relevant for the forensic investigation process? Which points are suitable for using forensic network tools? What kind of incident is this currently? These questions should be clarified at the beginning and of course must be documented. In this way, haphazard and excessive data collection can be avoided - which may prevent the focus on the essentials.

Methodology

- ➢ Qualified information gathering is the priority
- ➢ Quick interim network analysis results are often necessary
- ➢ Extensive network resources must be accessed

Strategy

- ➢ The methods of the investigation must be adapted to the individual goals of the investigation (e.g. minimising damage as quickly as possible or best possible prosecution)
- ➢ The network interventions must be weighted (assessment of network stability)

In order to obtain qualified information about the network, comprehensive access permissions to the network must be ensured. This can be done on a purely technical level, but also through simple surveys of administrators. To do this, an interface to the network must be created as quickly as possible in order to be able to implement and carry out further, targeted investigations. This shows that enormous hardware resources (e.g. for traffic mirroring) may become necessary during the investigation process. As part of preventing damage, it may also be necessary to separate individual systems in the network and carry out separate analyses there.

3. Data Collection

Data collection is truly the most important part of forensic work on computer networks. In practice, various problems such as inadequate network connection, missing export routes or missing authorizations regularly play a serious role. Primarily, as much data should be collected as possible. - But the network must not be placed under excessive strain or even collapse. This would destroy the further forensic process and could completely prevent the incident from being investigated.

Methodology

- ➢ The network traffic should be comprehensively secured for later analysis
- ➢ At the same time, an evaluation of the network traffic should take place
- ➢ Analysts should provide interim results live and derive targeted measures
- ➢ Use of existing log files from firewalls, IDS or IPS

Strategy

- ➢ After identifying important network processes, they must be prioritized in the analysis and recording of network traffic
- ➢ Sufficient resources must be available at all times for forensic activities and also for the functioning of the original network actions.
- ➢ Excessive disruption to the network due to data collection must be avoided, as there is a risk of a complete network collapse

As you can see here, data collection plays a central role in the forensic investigation process in computer networks. There are also serious risks, such as the network being compromised by forensic work and even its collapse. A sensitive and experienced approach is therefore advisable. This should always be done within the framework of proportionality, taking into account the network load and within the scope of the final objectives of the investigation.

4. Preservation

Preservation is the secure storage of information for later analysis.

And that is certainly not easy in the area of network forensics. The forensic process requires constant data consistency to ensure the immutability of the collected data.

You want to make sure that the network information is secured comprehensively and verifiably.

Methodology

- ➢ Network traffic is stored in a secure data format with additional protocol information
- ➢ The data can be copied, exported and displayed immutably

Strategy

- ➢ Important data must be separated from unimportant data in real time
- ➢ Only relevant data is saved
- ➢ Data must be saved in subsets in an unchangeable and reconstructable manner.

Some of the classic problems of traditional network forensics can already be seen in the points mentioned above, namely the problem of consistent data collection. We will talk about the technical details in more detail later. A fundamental problem, however, is the lack and swapping of data packets in the network. Another problem is recognizing important and unimportant information in real time with encrypted data connections. Last but not least, the secure storage of the data in a secure data format or data stream also plays a major role.

5. Examination

The examination is often described as the supreme discipline of forensics. But it can also be the most difficult discipline.

A wide variety of data packets must be examined in order to then be analysed and the contents to be logically evaluated.

Methodology

- Collected data or the data from the live examination must be checked for consistency
- What is important is distinguished from what is unimportant

Strategy

- Aspects of network traffic are already being worked out
- Patterns of communication should already be worked out
- A time estimate and an assessment of the amount of data are made

In the examination, the first structures of the relevant data are worked out in order to be able to make fundamental statements. An estimate of the expected amount of data is made. Statements can also be made about the data quality and the aim of the analysis.

It must also be checked whether the data collection needs to be readjusted.

The forensic scientist's work in this phase is to closely monitor the process in order to identify errors and, if necessary, provide feedback or more precise program settings.

6. Analysis

In the first data analysis - and this can be carried out promptly on site or at a later date in the laboratory - the forensic scientist tries to make initial goal-oriented statements about the data collected.

This can be done using special applications or by analysing the raw or log data.

Methodology

- ➢ Initial results are provided in the forensic process
- ➢ The data is correlated with each other
- ➢ Automated filtering and analysing algorithms highlight relevant information

Strategy

- ➢ Data is already analysed in a goal-oriented manner
- ➢ Initial interpretations of the available data are already being made
- ➢ The evaluation of the data flows into the further forensic process

In most cases, the analysis of the data is automated and filtered by the forensic scientist's chosen software.

It is important to first grasp the general information and then concentrate on the objectives of the investigation or identify problems. However, a hasty interpretation of the data should be avoided in the interests of objectivity and impartiality.

7. Investigation

The investigation describes the in-depth process of interpreting and evaluating the available data for possible passing on to other experts.

In most cases, a written technical report is produced that contains comprehensive information about the forensic process.

However, a legal assessment should not take place, but facts can be identified using circumstantial evidence.

Methodology

- transcription of the results of the analysis
- description of problems and errors
- compliance with the four-eyes principle

Strategy

- possible filtering of the confusing data situation
- Comparison of the incident with other incidents
- Elaboration of the core questions of the analysis assignment

In the area of investigation, the results of the previous analysis are implemented in a human-readable way. This includes describing the procedure for other forensic experts to review the results. Reference must be made to the specific data situation at all times.

Logical conflicts or errors must be presented and described.

Problems can be the large amount of data to be displayed, as well as a lack of distinction between unimportant data or the omission of later important data constellations.

8. Presentation

The final step of the forensic work is the presentation of the available data and investigation results.

A graphical representation is often used to present the data in a way that is understandable to laypeople.

Methodology

➢ Preparation of data for non-experts

➢ Creation of graphic references

➢ Plastic comparisons and results

Strategy

➢ Presentation of the objectives of the study

➢ Pointing out problems and errors

➢ Showing logical conclusions to the investigative task

Presentation is a very important part of forensic work. Third parties should also come to a conclusive result. In the presentation, the incident should be presented in detail with the essential content in order to enable an external assessment. Important data is included in the presentation and must be described factually and logically.

The presentation is only an excerpt of the complete analysis and of course the underlying data collection. However, the referenced raw data must be used in the event of demand.

2.3. Typical Security Strategies in Computer Networks

Nowadays, important IT security mechanisms are implemented and practically applied in SDN as well as in classic computer networks. When these IT security mechanisms are specifically used in SDN, these procedures are adapted and modified for use in SDN. The core of this is about hardening the computer systems in the network in order to prevent (malicious) manipulation or data leakage. In the following, we will first move on to the classic IT security strategies[2] and their implementation in SDN.

- Access Control, Two-Factor Authentication, e.g.
- Network Segmentation & Security
- Public Key Infrastructure
- System Hardening, Zero Trust Architecture
- Monitoring of Systems and Networks
- Logging and Log Analysis
- Vulnerability Analysis (Pentesting, etc.)
- Response to Security Incidents
- Adapted Forensics
- Data Recovery and Backups (Data Loss Prevention)
- Documentation, Transparency and Consistency
- Data Encryption
- Compliance with Individual Legal Requirements for Data Security
- Patch Management and Antivirus
- Intrusion Detection & Prevention Systems (IDS, IPS)
- Endpoint Security

[2] Of course, there are many different approaches to IT security, and only the most common methods are listed here.

In the course of this work, we cannot explicitly discuss every single point of IT security in detail for SDN and show the different adaptations. But it is important to understand that these (and other) security mechanisms are currently implemented in SDN and can be implemented dynamically and adapted very easily. So let's take some of the above points and try to discuss what this means in SDN technology.

- Access Control

Access control is mostly about rights management and user administration. Not every user has the same rights and views in the network. This is particularly important to take into account with regard to forensics in SDN, as it may not be possible to gain insight into important network areas. There are often approaches for obtaining information in SDN via the central control instance. Extremely important information about the network can be obtained there.

- Automated Response to Security Incidents

In the context of forensics in the SDN, it should also be noted that forensic activity can automatically be assessed as a dangerous incident. This can lead to the computer system being severely impaired or even automatically isolated. Therefore, one should always start by passively collecting information about the network. Otherwise, security settings such as Intrusion Detection & Prevention Systems (IDS, IPS) could trigger reactions that falsify the results or, in the worst case, partially paralyse the network.

- Data Encryption

Sensitive data is usually encrypted when it is stored and transmitted. And this often involves data that is of forensic importance. An attempt could now be made to decrypt this data after the data capture process, which is very complex or impossible if the encryption is implemented correctly. As a rule, it is better to access the data in a decrypted state with the respective communication partner or with appropriate authorisations. This makes live forensics particularly important, as it is not enough to record all network traffic at any point; instead, direct access to the relevant data is required in SDN.

If you now look at the architecture and security mechanisms of an advanced SDN such as the Zero Tier One network, you will see that advanced cryptographic methods such as Curve 25519 or Ed 25519 have been implemented. This corresponds to 256-bit Salsa 20 encryption with Poly1305 MAC algorithms as part of authentication and secure data transmission. Since these involve proprietary algorithms and processes, the encryption operations can currently be viewed as secure. To control the network, ZeroTier uses network controllers that can be secured using double authentication, short-term passwords or biometric procedures. A controller can theoretically serve 2^{24} networks and millions of devices [Ze24].

So if you look at the security mechanisms in SDN, you will see that various combinations of IT security are used in different applications. In most cases, very secure end-to-end encryption is implemented in advanced SDN, which in turn makes intercepting encrypted network traffic not effective.

Even in applications from the same manufacturers, different encryption or authentication mechanisms are used depending on the update or program version or depending on the security needs of the user.

Without knowing and understanding these security mechanisms on site in individual cases, forensic investigations in SDN will often fail. In real cases, additional investigation methods must also be considered in order to possibly be able to access the SDN control centre. From here, forensic network investigations can be organised in a structured manner and important information can be obtained.

Further encryption methods will not be discussed in this work; this would concern the encryption of the data on hard drives, which is a task of digital forensics to secure this data.

3. Technical and Structural Study of SDN

3.1. Aspects of SDN

Let's have a look at the technical structure of a typical SDN computer network. Simply put, a software-defined network is a subcategory or a delimited area of a "normal" computer network.

The special network control software of the SDN accesses existing network resources and essentially creates its own (virtual) software-designed network with its own resources and functions [Sp16].

The fundamental speciality of SDNs lies in the separation of the control plane from the data plane. In traditional networks, both planes are closely linked, which makes the network configuration complex and static. SDNs, on the other hand, enable centralised management of the control plane via one or more SDN controllers, while the data plane is still implemented on the physical network devices such as switches and routers.

One difficulty in network forensics concerns is the recognition of these structures, whereby the view of things depends, as always, on one's own location. Seen from the outside, the structure of an SDN may not be visible at all. The traffic of an SDN can adapt to normal, visible other network traffic and hide in capsule mechanisms and is not visible at first glance, such as in VPN network tunnels.

If the (remote) viewer is within an SDN, such as a virtual machine, certain network resources may remain hidden from view. This leads to the potential for misunderstanding the true nature of the situation. Some network resources may not appear at all because they are temporarily disabled or do not respond to regular requests due to their special configuration.

These technical features of SDN make it difficult to examine network traffic and network resources in many cases. The unproblematic scaling of SDN-controlled networks can create very confusing networks, even for experts, in which a final analysis becomes almost impossible.

Nevertheless, in-depth knowledge of network forensics is extremely important in almost all cyber incidents currently taking place. Most attacks on computers take place via networks, and perpetrators rely on the fact that they can hide in special network structures.

In the following figure, you can see an arbitrary network structure and simplifies the presence of an SDN/VPN:

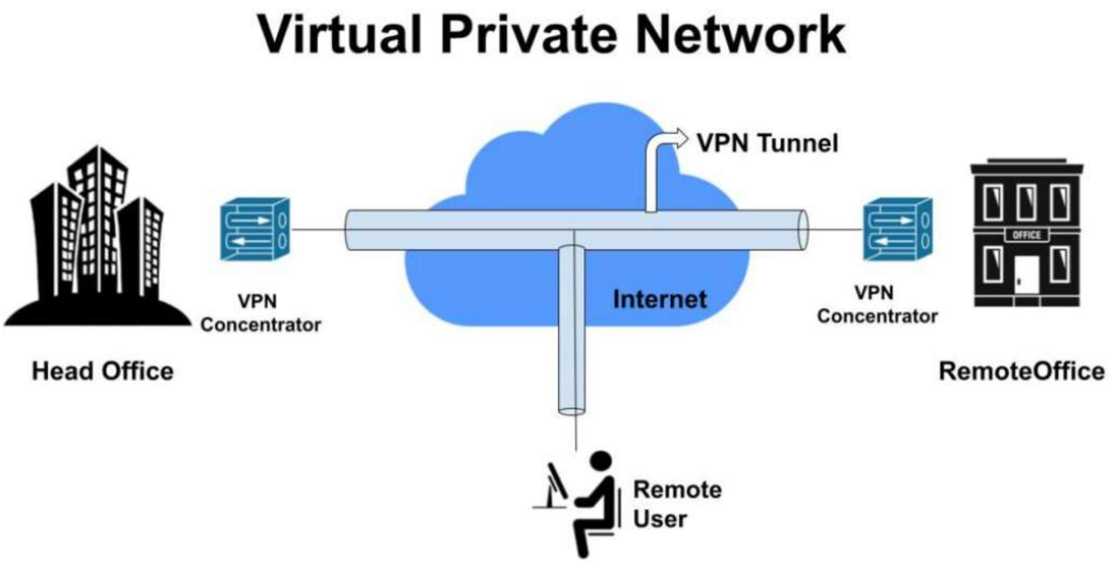


Fig. 3: Overview of VPN-tunnelling [Ja21]

This illustration clearly shows the typical function of a VPN that securely connects different clients over the Internet. Essentially, SDNs work very similarly to VPN when used for these purposes. However, SDN can also work without internet areas and often offer a wider range of functions than VPN.

Dynamic adaptability is one of the outstanding features of SDNs. Thanks to centralised control, networks can react to traffic changes and network failures in real time. Load distribution, traffic engineering and quality of service can be dynamically adjusted to optimise network performance. This is particularly beneficial in environments with highly fluctuating traffic patterns, such as data centres and cloud environments.

SDNs also offer enhanced security functions through centralised control and the possibility of programmatic network configuration. Security policies can be centrally defined and enforced, simplifying the management of firewalls, intrusion detection systems and other security mechanisms. In addition, virtual networks and network segmentation can be efficiently implemented to ensure the isolation of network segments and reduce the attack surface.

Another important aspect of SDNs is the promotion of interoperability through open standards. Protocols such as OpenFlow enable communication between SDN controllers and network devices from a wide range of manufacturers. This promotes the integration of heterogeneous network environments and avoids vendor dependencies. In addition, open standards drive innovation and enable the development of new technologies and protocols.

In summary, an SDN is a new and very dynamic network structure. Network control and management are separated from the data layer. This improves adaptability and enables individual programmability of the network forms. The SDN is based on almost any hardware network components, although these hardware structures are not natively accessible or even visible to the user. In addition, different hardware components can be virtualized and reprogrammed.

Last but not least, the security of complex forensic data must be guaranteed at all times so that the data is protected against manipulation and the integrity of the evidence is maintained.

Should forensic analysis data be captured by hackers, this could expose critical security vulnerabilities and enable an intrusion into the network.

3.2. Forensic Tools

As part of network forensics, there are different programs for many individual use cases. For digital forensics experts in Europe, the catalogue of forensic tools at www.dftoolscatalogue.eu (European Informatics Data Exchange Framework For Courts And Evidence) with over 1000 applications of digital forensics is a classic reference point for practical programs that enjoy a certain degree of recognition and have proven themselves legally among digital forensics experts in various practical applications including legal proceedings .

In the following, we want to concentrate on typical free network forensics tools [Al11] that are used in practice by digital forensic experts on behalf of the government or by special security companies. These are mainly tools that are accessed via the command line and analyse activities in networks or initiate actions in the network. These tools can be run on special forensic computers or on computers in the system in live operation. Special software, such as professional network analysis tools from Cellebride, EnCase, Sumuri, Guidance, MAGNET Forensics, or many others will therefore not be described below. These commercial solutions are often closed and protected applications, so that the forensic scientist himself has little in-depth insight into the processes of these programs. This ultimately has the consequence that the user can no longer make well-founded statements as an expert, but can only report on the use of the program.

In contrast to the programs described below, it is of course possible to implement and operate your own solutions. However, in accordance with forensic principles, this is explicitly discouraged. Your own programs not only have to be error-free and completely documented, but also be a tried and tested standard in the field of forensic scientists. However, in-house developments usually do not correspond to best practices and tried-and-tested practical implementation.

A practical tip is of course to have comprehensive knowledge of the program that is to be used. So, familiarise yourself with all the use cases and also the problems with the chosen application. This is part of the preparation for a forensic analysis of a network and is an important strategy in network forensics.

It must also be mentioned that most of the applications presented here cannot provide a look into the past of network activities. An analysis of activities that have already been completed, such as downloads, storage processes, replacement of network adapters, installation of network software, (re)configuration of network adapters, firewalls or intrusion detection systems must be considered separately and are not part of live network forensics of live network traffic. However, these forensic methods such as an analysis of log files are an essential part of the forensic process and in particular for obtaining information about network structures and special network processes.

On the following pages, some important and regularly used applications of network forensics will be described without providing complete documentation. The applications are not sorted according to importance, power or the quality of the results. For more detailed and in-depth descriptions, the manufacturer's website must be accessed, and the documentation must be observed.

The tasks and goals of the selected tools are initially to clarify the possibly completely unknown network to be examined. The user wants to examine the network structure and have little influence on the operation of the network. The programs shown often have passive and active to aggressive examination methods. You can also often set your own protocol actions. When it comes to the protocols, it is always important to record the maximum number of events for later in-depth analysis without negatively affecting performance. Try to record all your steps completely. This includes not only the user input, but also all program steps that are executed. An investigation should always be carried out in a comprehensible and ultimately transparent manner.

3.2.1. Nmap

Nmap (Network Mapper) is a powerful tool for network discovery and audits. It is an open source tool and is one of the most used tools in IT security, allowing administrators and security experts to collect information about networks and hosts, detect vulnerabilities and simulate attacks.

Nmap can be used, among other things, to find all active devices on a network. It uses various techniques, such as ping sweeps, TCP connect scans and SYN scans, to identify the IP addresses of active hosts.

Nmap can also attempt to detect a host's operating system using various fingerprinting techniques.

Another function is the detection of running services on a host. Nmap scans ports and uses various techniques to identify the service and the version of that service listening on a particular port.

Finally, Nmap supports the execution of individual scripts and thus the execution of automated tasks. However, typical network functions can also be put under strain and intensive network scans can have a critical impact on the network.

There is an interesting GUI solution for Windows called Zenmap. This solution graphically displays, among other things, typical network topologies and is therefore suitable for information visualisation for non-experts.

Web:	https://nmap.org
Download:	https://nmap.org/download.html
Author:	“Fyodor” Gordon Lyon
Version:	Nmap 7.94 (03/2024)
OS:	Windows, Linux, MacOS

3.2.2. WinPmem

WinPmem is an important open source memory acquisition tool in the field of digital forensics. It allows you to extract a complete image of a Windows computer's memory (RAM) for forensic investigations.

For this purpose, system-related functions are used to access the target system's RAM. The data is read from the RAM and an image is created in the selected file format.

WinPmem works on various operating systems including Windows, Linux and FreeBSD. Depending on the configuration, WinPmem can also be used for memory collection of remote systems and provides various options for configuring memory collection, such as file format selection and compression.

WinPmem is an important tool in digital forensics for securing and analysing volatile data from RAM. This data can, for example, contain information about running processes, deleted files or passwords.

During security incidents, WinPmem can be used to capture the state of memory at the time of the incident and preserve evidence.

It also enables the analysis of malware that is active in the RAM.

There is a similar tool for macOS called OSXPMem by Johannes Stuettgen. On Linux, RAM images are often created using dd, linpmem or LiME.

Web:	https://winpmem.velocidex.com/docs/
Download:	https://github.com/Velocidex/WinPmem
Author:	Viviane Zwanger, Mike Cohen, Emre Tinaztepe, Mehmet Göksu
Version:	WinPmem 4.0 RC2 (10/2020)
OS:	Windows

3.2.3. Wireshark

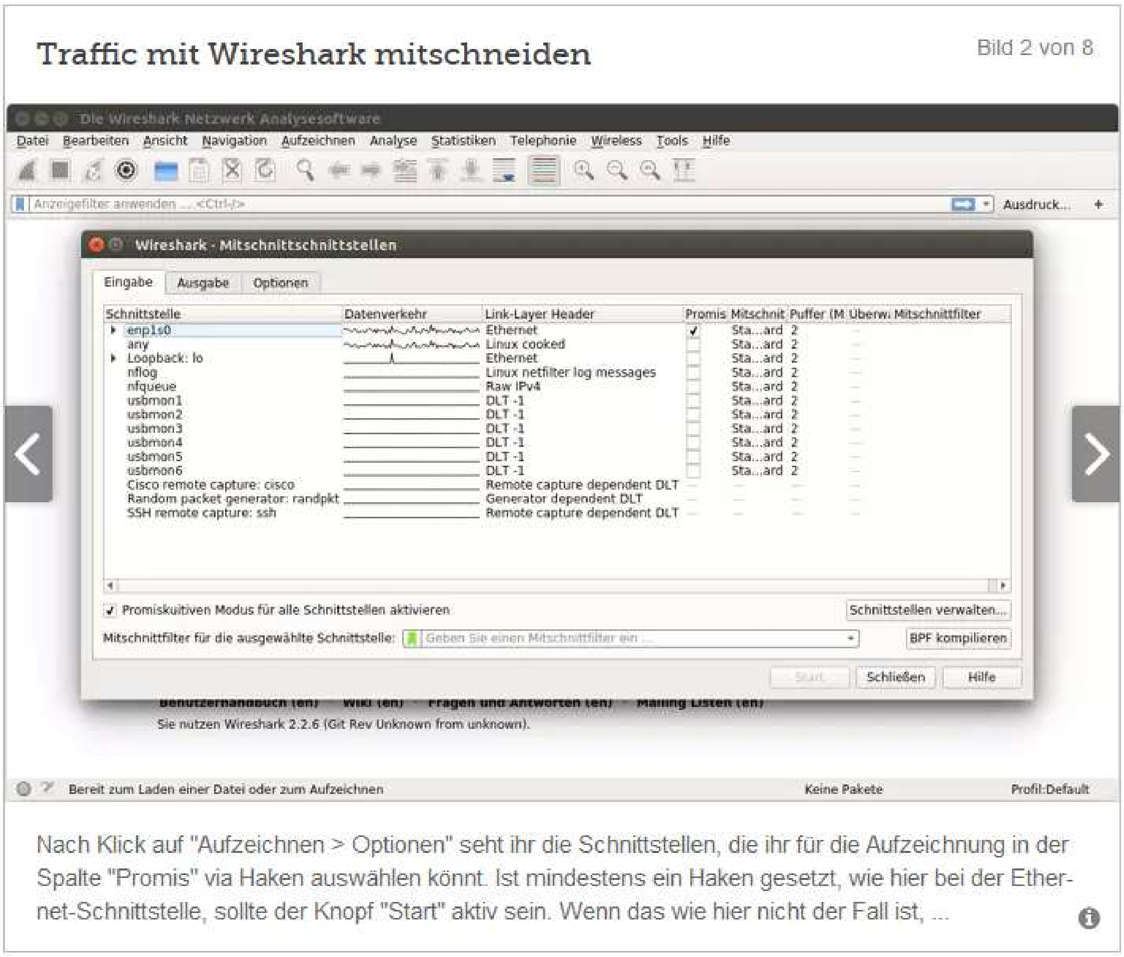
Wireshark is a powerful open source network protocol analysis tool used by network administrators, security experts, developers and forensic analysts. It has the ability to capture, analyse and visualise network traffic at various levels. The application offers a comprehensive solution for investigating and diagnosing network problems, monitoring network security and forensic analysis of network attacks.

Wireshark's functionality is primarily based on packet capture, where it can capture network traffic across various network adapters and interfaces. This capture can be done either in real time to analyse live traffic or by loading already recorded capture files. Wireshark decodes individual data packets in network traffic and automatically recognizes the protocols used such as TCP/IP, HTTP, DNS and others. This information is presented in a human-readable form, allowing users to easily understand and interpret network traffic.

Wireshark is a valuable tool for network protocol analysis and developing new network protocol implementations. With its ability to support and analyse a wide range of network protocols, Wireshark enables developers to understand and optimise communication between different devices on the network.

In digital forensics, Wireshark can be used to collect and analyse evidence in network attacks. By collecting and analysing network traffic, forensic analysts can reconstruct the activities of attackers and gain important information about the activities or attacks being carried out.

Wireshark offers a variety of advantages, including its open-source nature making it available for various operating systems, its extensive network traffic analysis and visualisation capabilities, and its support for a variety of network protocols. Despite some limitations, such as complexity for beginners and dependence on the type and volume of network traffic captured, Wireshark remains an indispensable tool for network professionals worldwide due to its power and versatility.



Traffic mit Wireshark mitschneiden

Bild 2 von 8

Die Wireshark Netzwerk Analysesoftware
Datei Bearbeiten Ansicht Navigation Aufzeichnen Analyse Statistiken Telephonie Wireless Tools Hilfe
Anzeigefilter anwenden ... <Ctrl-/>
Ausdruck... +
Wireshark - Mitschnittschnittstellen
Eingabe
Ausgabe
Optionen

Schnittstelle	Datenverkehr	Link-Layer Header	Promis	Mitschnit	Puffer (M	Überw.	Mitschnittfilter
enp1s0		Ethernet	☑	Sta...ard	2	—	
any		Linux cooked	□	Sta...ard	2	—	
Loopback: lo		Ethernet	□	Sta...ard	2	—	
nflog		Linux netfilter log messages	□	Sta...ard	2	—	
nfqueue		Raw IPv4	□	Sta...ard	2	—	
usbmon1		DLT -1	□	Sta...ard	2	—	
usbmon2		DLT -1	□	Sta...ard	2	—	
usbmon3		DLT -1	□	Sta...ard	2	—	
usbmon4		DLT -1	□	Sta...ard	2	—	
usbmon5		DLT -1	□	Sta...ard	2	—	
usbmon6		DLT -1	□	Sta...ard	2	—	
Cisco remote capture: cisco		Remote capture dependent DLT	—	—	—	—	
Random packet generator: randpkt		Generator dependent DLT	—	—	—	—	
SSH remote capture: ssh		Remote capture dependent DLT	—	—	—	—	

☑ Promiskuitiven Modus für alle Schnittstellen aktivieren

Mitschnittfilter für die ausgewählte Schnittstelle: Geben Sie einen Mitschnittfilter ein ...

Schnittstellen verwalten...
BPF kompilieren
Start
Schließen
Hilfe
Benutzerhandbuch (en) · Wiki (en) · Fragen und Antworten (en) · Mailing Listen (en)

Sie nutzen Wireshark 2.2.6 (Git Rev Unknown from unknown).

Bereit zum Laden einer Datei oder zum Aufzeichnen

Keine Pakete

Profil:Default

Nach Klick auf "Aufzeichnen > Optionen" seht ihr die Schnittstellen, die ihr für die Aufzeichnung in der Spalte "Promis" via Haken auswählen könnt. Ist mindestens ein Haken gesetzt, wie hier bei der Ethernet-Schnittstelle, sollte der Knopf "Start" aktiv sein. Wenn das wie hier nicht der Fall ist, ...

Fig. 4: Screenshot from a Wireshark tutorial from heise.de [Gr17]

There are now recommended professional online tutorials for training in the use of Wireshark.

Web:	https://www.wireshark.org
Download:	https://www.wireshark.org/download.html
Author:	Wireshark-Foundation
Version:	Wireshark 4.2.3 (04/2024)
OS:	Windows, macOS, Linux/Unix

3.2.4. Responder

Responder is a free classic command line tool in Kali Linux that focuses on intercepting specific network protocols to collect information from users, computers or the network.

It can passively listen for requests from client computers in network traffic that are aimed at finding network resources such as file servers or other services. Depending on the configuration, Responder can also pose as a target resource computer and imitate response packets.

Impersonated or fake login requests can be sent so that the client is deceived. These fake credentials can be used to intercept credentials by responders. The main protocols used by Responder are LLMNR, NBT-NS, WINS and HTTP. Responder can be used for legal penetration testing purposes ("Analysis Mode"). Use for illegal purposes ("Attack Mode") is punishable.

As part of real time analyses of SDN, Responder can be used to obtain quick results during ongoing operations, e.g. which services are active in the network, how much traffic is taking place and what user information is available. Fingerprinting and analysis of the target computers in the live network can also be carried out. Responder runs on Python 2 and is currently being further developed by the author Laurent Gaffie.

Web:	https://www.kali.org/tools/responder/#tool-documentation
Download:	https://github.com/lgandx/Responder
Author:	Laurent Gaffie
Version:	Responder 3.1.4.0 (Python 2)
OS:	independent (runs with Python 2)

3.2.5. Volatility

The Volatility Framework is a powerful open source tool developed by the Volatility Foundation and has wide application in digital forensics. Its speciality is analysing memory dump files from various operating systems. It offers forensic experts the ability to extract and evaluate information from the RAM of computers and mobile devices.

This versatile framework supports a variety of operating systems such as Windows, Linux, and macOS, as well as various memory dump formats, including hibernation files, crash dumps, and virtual machine snapshots. It has an impressive collection of plugins specialised in identifying and analysing specific artefacts in memory. This includes processes, file systems, network connections, registry entries and much more.

The Volatility Framework allows forensic investigators to determine what processes were active on a system, what files were opened, what network connections existed, and what user interactions occurred. These features prove extremely useful for tracking attacks, detecting suspicious activity, and preserving evidence for forensic investigations.

Volatility can process memory images from Windows 32 and 64-bit systems up to Windows 10, as well as Linux machines up to kernel 5.5. It can also create memory dumps of virtual machines and analyse running network processes.

Web:	https://github.com/volatilityfoundation/volatility
Download:	https://github.com/volatilityfoundation/volatility
Author:	Volatility Foundation
Version:	Volatility 2.6.1 (04/2024)
OS:	independent (runs with Python 3)

3.2.6. Netcat

Netcat, or nc, is a versatile command-line tool available on Linux, offering a wide range of networking capabilities. It is used for simple file transfers, transferring data to sockets or ports, data tunnelling, and various other functions, supporting both TCP and UDP protocols, making it invaluable for network diagnostics and troubleshooting.

Key features include facilitating file transfers between systems, performing port scans to identify open ports, and starting reverse shells for remote command-line access, essential for penetration testing. Netcat also helps explore network configurations and troubleshoot connectivity issues.

Netcat is frequently used to check security settings, such as firewall configurations, by sending and receiving data to test network defences. It can create data tunnels, helping to bypass network restrictions or encrypt data when combined with other tools. Derivatives like Ncat, which includes SSL support, Socat, a more advanced version supporting various protocols, and Cryptcat, an encrypted version, extend Netcat's functionality.

Included as a standard tool in most Linux distributions, Netcat is readily available for administrators and security professionals. However, its powerful features can also be used maliciously, making it crucial to monitor its presence on systems and restrict usage to authorized personnel.

In summary, Netcat is an indispensable tool for network administration, troubleshooting, and security testing in the Linux environment, capable of handling tasks from simple file transfers to complex network diagnostics.

Web:	https://docs.oracle.com/cd/E86824_01/html/E54763/netcat-1.html
Download:	standard in linux distributions
Author:	“Hobbit” (unknown)
Version:	1.10-48
OS:	Linux, MacOS

3.2.7. Metasploit

Metasploit is a well-known open source penetration testing framework that helps security experts and pen testers assess the security of computer systems and networks. It includes numerous tools, modules, scripts and exploits to find vulnerability information, test and exploit vulnerabilities and attack systems.

The complexity of the modules is quite high and requires some expertise from the user, as documentation is often sparse. Furthermore, some of the modules are classified as criminal by the security authorities and a legal assessment is therefore required when executing or maintaining these programs.

However, for students or those willing to learn, there are also special vulnerable systems for virtual machines such as Metasploitable and Metasploitable 2, so that attacks can be safely played through and tried out for practice purposes.

By April 2024 there are over 5500 Modules available in the Metasploit Framework.

Web:	https://www.metasploit.com
Download:	https://www.metasploit.com/download
Author:	various
Version:	6.4.0
OS:	Linux

3.2.8. Network Miner

Network Miner is a tool for collecting network information for Windows. Due to the emulation in Wine, it can also be run under Linux.

No installation is required, so the application can easily run on third-party Windows computer systems.

It is an open source tool for live forensic network analysis with an inbuilt passive network sniffer and packet capturing module. The modules attempt to detect operating systems, sessions, hostnames, open ports, messages, images, keywords, files, etc. without burdening the network.

Network Miner can also open .pcap and .ngpcap files and analyse recorded data streams for offline analysis.

There is a free, slightly limited edition, which was used in this work, and a paid professional version of the program, which has a wider range of features.

Web:	https://www.netresec.com
Download:	https://www.netresec.com?page=NetworkMiner
Author:	Erik Hjelmvik
Version:	NetworkMiner 2.8.1 (.exe-File)
OS:	Windows (Linux)

3.2.9. Velociraptor

Velociraptor is a client-server platform designed for collecting network information and supporting security investigations on computer networks. Written in the Go language, it is deployable on both Linux and Windows systems. The client application offers real-time monitoring of network traffic, allowing multiple clients to deliver data for analysis concurrently.

Evaluation processes can be automated using a separate scripting interface, and some results can be partially visualized. Velociraptor aims to move beyond the classic Digital Forensics and Incident Response (DFIR) workflow—comprising data acquisition, post-processing, analysis, and presentation—by performing targeted analyses directly on the clients.

However, a challenge with Velociraptor is that its libraries and program files may become outdated and incompatible with current processes or operating systems. Despite this, Velociraptor remains an innovative tool for modern network forensics, providing a dynamic approach to real-time data collection and analysis.

Web:	https://docs.velociraptor.app
Download:	https://www.github.com/Velocidex/velociraptor
Author:	Michael Cohen
Version:	velociraptor v0.72 (04/2024) (.exe-File)
OS:	Linux, Windows

3.2.10. Tcpdump

Tcpdump is a powerful and versatile standard Linux network traffic collection tool. It is actually the de facto standard for data collection in network forensics.

The data is either output directly to the console or .pcap files are created that can be used to save package data for later analysis.

There are many configurations and filtering parameters with which tcpdump can be used, so it is strongly recommended to read the manual carefully.

Also, you can conveniently use tcpdump in Bash or Python scripts for automation, such as to quickly hash your collected data or give you important quick information or issue automated alerts.

Some examples for using tcpdump on linux command line:

```
$ sudo tcpdump -h
$ sudo tcpdump -i eth0 -vvv
$ sudo tcpdump ’icmp’ -i eth0 -vvv
$ sudo tcpdump -i eth0 -vvv | grep ’google.com’
```

Fig. 5: Examples for using tcpdump on CMD

A lot of useful additional information can be found on the developer page of tcpdump.

Web:	https://www.tcpdump.org
Download:	https://www.tcpdump.org/index.html#latest-releases
Author:	Van Jacobson, Craig Leres, Steven McCanne
Version:	tcpdump 4.99.3 (04/2024), libcap 1.10.3
OS:	Linux/Unix

3.3. Classical Approaches to Forensics in SDN

Basically, it can be said that the programs and therefore possibly the methods of network forensics in SDN are not particularly different from the investigation methods in traditional networks because SDN are basically supposed to be compatible with existing network applications. Nevertheless, there are actually some important subtleties to keep in mind when examining SDN. On the one hand, it is worth noting that you often do not have access to the underlying infrastructure when being in an SDN. This is due to the technical property of an SDN that it can adapt dynamically and individually to the given network infrastructure and network levels are strictly separated from each other. Secondly, an SDN can be often seen as a limited part of a larger network structure. So there is only limited insight into global network activities when being within the SDN. Or several SDNs can exist in parallel on one computer system without being able to access the other from one or even know its existence.

It is therefore essential for a network forensic expert to deal with the properties of an SDN and to know the technical possibilities and peculiarities. Software-defined or software-controlled virtual networks have long since found their way into many applications, be it in legal and productive environments as well as in criminally organised network environments.

Nevertheless, the goals of forensic information gathering are usually similar: What happened? Where did it happen? When did which action take place? Which systems are affected? And last but not least: How can I protect the network?

The criteria for obtaining reliable evidence are extremely important for criminal prosecution in the evidence process. It must therefore be comprehensively documented, verified and validated so that the integrity and authenticity of the data is always maintained [La17]. The most comprehensive data collection is of no use to us in this process if the data is not protected against manipulation at all times.

Can the network data even be comprehensively secured? From a technical perspective, in most cases not. The cause is often the extreme volatility of the data, the high network complexity, lack of bandwidth, low network resilience, lack of legal basis or other criteria. If a network structure is securely hidden - similar to a VPN or a special SDN, such as the TOR network - then I cannot acquire this network traffic - or not in a meaningful way - for analysis. This also raises the fundamental question of whether encrypted network traffic needs to be stored at all. In practice, of course, this depends on the use case and should be considered as part of the data collection. In any case, such decisions must be consolidated and documented.

“Who am I? And if so: where and how many?”

This metaphor, based on the german author Richard David Precht, could potentially be one of the central questions in network forensics in SDN. If this question (about your own identity in the network) cannot be clarified in the further forensic process - then it looks very bad for usable results to be produced. Even the question of the number of networkers raises complications in detail:

Does each process count individually? Does each computer count individually? Does each socket count individually? Furthermore, it should somehow be shown where network actions take place. On the server? On the client? On the way in between (Man in the Middle)?

Some define network forensics like this:

“Network forensics is the capture, storage, and analysis of network events.” [Mo23]

Unfortunately, this definition does not comprehensively meet the current requirements for clarification and information gathering in special network structures like SDN and is actually formulated too simply and insufficiently deeply.

We also need to look more closely at the criteria of digital forensics in order to be able to clarify an incident comprehensively and completely. Therefore, it is not only network traffic that plays a role in the analysis of networks. Here, simply saving or exporting the entire traffic as a copy would be enough to satisfy the forensic idea. A comprehensive and reconstructable analysis also means sending special requests to networks or modelling the network hardware of the different participants in the network.

If we were to look at a botnet as an example of an SDN, we could sometimes record network traffic for a very long time until something interesting happens. To investigate a criminal botnet, information from the control server would be required, which can possibly be found out through a simple message or a special ping. As an individual participant in the network, you would probably find out little or nothing about the other infected computers. To further expand the idea of the botnet, a controlled DDoS attack could then be carried out by the network on a target. This means you can quickly find out a lot about the participating computers, such as their IP addresses, but sometimes rapid storage or analysis cannot handle so much data (which is the nature of a DDoS attack).

An important point in network forensics is therefore to have the highest possible performance network access with correspondingly fast hardware in order to be able to comprehensively store network traffic and evaluate it in a timely manner.

Typical classic forensic investigation strategies are the examination of local data, such as log data, secondly, the collection of network traffic data that is sniffed via attack points, thirdly, the analysis of network devices such as routers, switches, servers, etc., fourthly, the analysis of network traffic data using techniques of live monitoring or network protocol representations, fifthly, the creation of a timeline to display significant events in the network. Other classic investigation strategies are also used and can be found in various literature sources [Eu19][Ke06].

4. Practical Forensic Analysis of the ZeroTier Network

For the most realistic possible investigation of an SDN, a widely used tool from ZeroTier Inc. US was selected as a representative for SDN. This SDN model can be used to show typical investigation methods and strategies. Of course, these may differ from other SDN implementations. The security implementations in SDN in particular are often individual. There will be security gaps or misconfigurations in many applications that can be exploited by forensic experts to obtain information. But these individual security techniques also make SDN vulnerable to external attackers in individual cases. And this is exactly where forensic scientists have to start in order to obtain information about these attackers. In the following, some network information sources with common forensic tools will be shown as examples. First, the SDN of the ZeroTier company will be basically explained, which is documented in detail on the homepage www.zerotier.com.

4.1. Basics

ZeroTier, Inc. is a private US company founded in 2011 that provides advanced software-defined networking solutions. A company description can be found on the company website www.zerotier.com and on Wikipedia.

The company offers commercial products and services worldwide for creating and managing software-defined networks. There are also free models that hardly differ in functionality from the paid models. As part of this work, the ZeroTier network was chosen as a practical example to show typical forensic scenarios and demonstrate strategies and methods for obtaining information.

What should be emphasised is the universality of the application for a wide variety of operating systems and the intuitive setup of the SDN controllers.

Another point is the extensive scalability (in the paid version) and the customizable dynamic encryption of data transfers up to high-security scenarios.

These properties define a modern SDN solution and include concepts that are currently used in companies and correspond to the state of the art.

There are therefore different use cases for such a network, such as support for Windows Remote Desktop, as a gaming network, as a VLAN or VPN, as SSH support, and much more.

In the following figure, we can see an overview of the ZeroTier architecture with only two devices connected together through the internet:

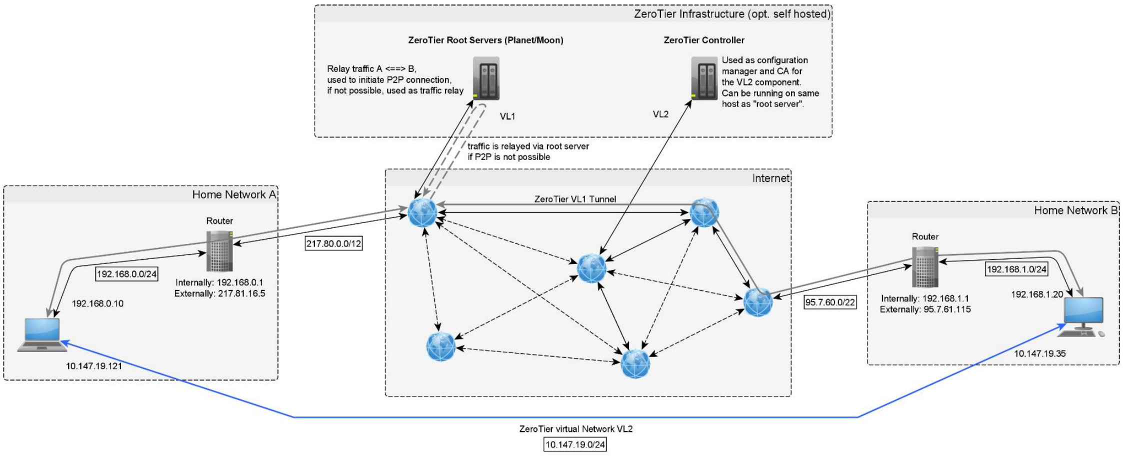


Fig. 6: Overview of the basic ZeroTier architecture [Ri20]

The modified figure above comes from the documentation of the ZeroTier network and shows how a secure connection between the individual computers is established via different control servers [Ze24].

As part of this work, a ZeroTier network was created for the investigations, documentation and representations.

Different operating systems were chosen as clients:

- Ubuntu Linux
- MX Linux (Debian)
- Raspbian (ARM)
- Garuda Linux (other)
- Windows 7 Ultimate 64
- Windows 10
- Windows 11
- Kali Linux
- Android 9 (BlueStacks, SM-S908E)

An graphical installer is available for Windows, Android, MacOS and iOS environments.

For the Linux operating systems or Docker, it is installed via the command line:

```
$ curl -s https://install.zerotier.com | sudo bash

…

**** Enabling and starting ZeroTier service…

Synchronizing state of zerotier-one.service with SysV service script
with /lib/systemd/systemd-sysv-install.

Executing /lib/systemd/systemd-sysv-install enable zerotier-one

*** Waiting for identity generation…

*** Success! You are ZeroTier address [**********]
```

```
$ zerotier-cli -v
1.12.2
```

```
$ zerotier-cli status
200 info ********** 1.12.2 ONLINE
```

```
$ sudo nmap -sS 10.147.20.26
Starting Nmap 7.93 ( https://nmap.org ) at 2024-04-14 17:22 CEST
Nmap scan report for 10.147.20.26
Host is up (0.17s latency).
Not shown: 997 closed tcp ports (reset)
PORT           STATE SERVICE
1234/tcp       open  hotline
5555/tcp       open  freeciv
12345/tcp      open  netbus
MAC Address: 12:05:03:03:79:E7 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 2.69 seconds
```

Fig. 7: Installing the ZeroTier Client under Linux

Joining a network is done as follows:

```
$ sudo zerotier-cli join ************


200 join OK
```

Fig. 8: Joining a network on the command line

Here, an example of setting up the ZeroTier client under Linux was shown. Setting it up on Windows, macOS or Android is just as easy and uncomplicated. Graphical interfaces are used to simplify setup.

In order to be accepted in the network, the respective request must be approved in the network control centre (my.zerotier.com) by an administrator.

Fundamentally, there are some additional pitfalls when examining software-defined networking that need to be addressed:

1. The problem of detecting the presence of SDN on a computer system.

2. The problem of determining whether one is in a SDN network.

3. The problem of understanding the complexity and dynamics of running foreign SDN.

In summary, the basic structure of an SDN can be described in such a way that there is a central administration interface and all members of the network are logged in there. However, this does not mean that computer systems outside the administration interface cannot also have access to the network. Connections that cannot be seen via the administration platform can also be created via routing or specially configured network bridges.

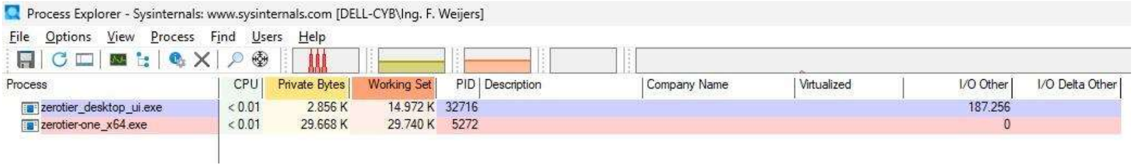
In the experiment for this work, it was possible to access a remote Orange Pi in a ZeroTier SDN with a Kali Linux under WSL in standard configuration. This of course contradicts applicable data security standards and clearly shows that the security mechanisms are incomplete, but it also shows that attackers can gain easy access to SDN systems undetected. Forensics in such SDNs must take these possibilities into account, and therefore must not assume strict data security implementations.

It should also be briefly mentioned that these network connections can be easily automated using scripts. The networks can be hidden from other users or only activated on a time-controlled basis. Network hopping can also be carried out so that actions in different SDNs are possible. When membership in an SDN is terminated, there is little trace that can subsequently represent these activities.

4.2. Functionality

In order to determine network activity, it is often necessary to examine the running processes with approaches of live forensics. The running processes can be displayed using Windows on-board tools, for example. There are also different functions under Linux and other operating systems for analysing running processes. It may be important to create a memory image beforehand. This way the processes can be identified afterwards. As mentioned, a system-level tool is required for live forensics that can display network processes in real time.

The Process Explorer for Windows[3] was used in this example to analyse SDN tasks. The program shows the running processes and a lot of valuable information about the running programs without installation.



Process Explorer - Sysinternals: www.sysinternals.com [DELL-CYB\Ing. F. Weijers]

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name	Virtualized	I/O Other	I/O Delta Other
zerotier_desktop_ui.exe	< 0.01	2.856 K	14.972 K	32716				187.256	
zerotier-one_x64.exe	< 0.01	29.668 K	29.740 K	5272				0	

Fig. 9: Screenshot of running ZeroTier processes in Process Explorer

In the detailed information tabs of the Process Explorer, you can now view running network processes of the selected process. In the image on the next page, you can see regular network queries from the zero-tier adapter, which wants to ensure a stable connection. However, intensive data transfers are often not illustrated in the process because the data flow is established via other sockets.

[3] Process Explorer v17.05 - Sysinternals: https://learn.microsoft.com/en-us/sysinternals/downloads/process-explorer

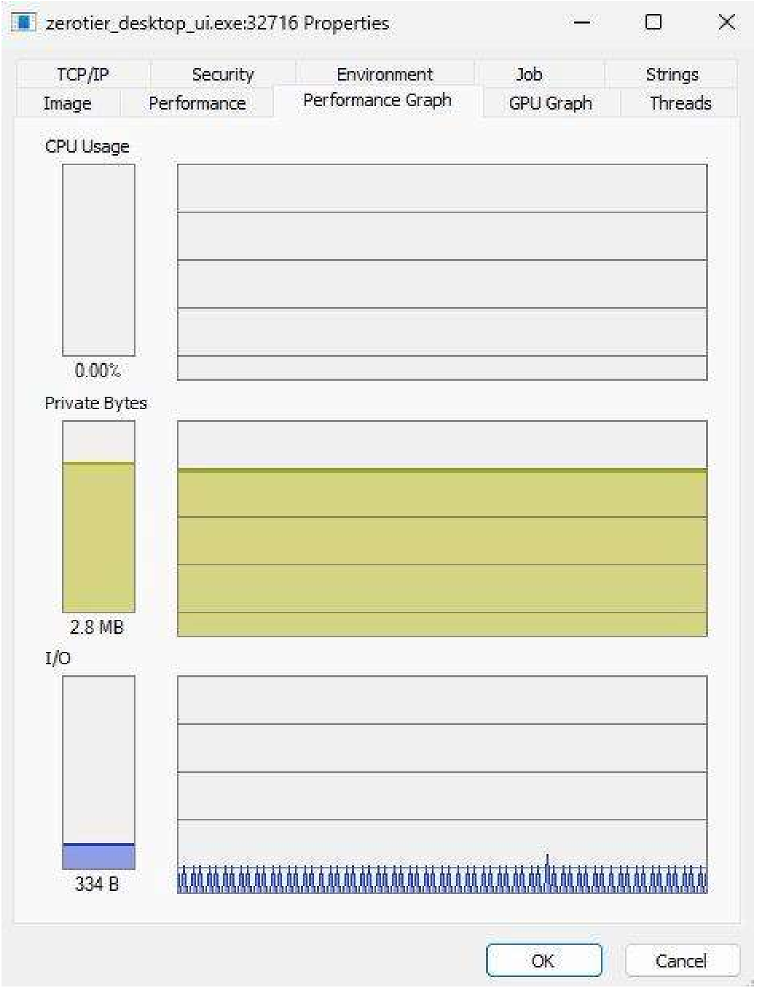
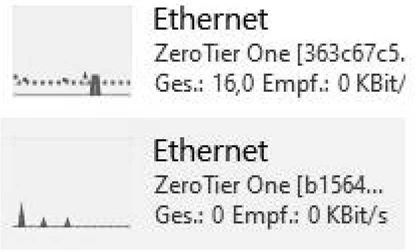


Fig. 10: Detailed view of zerotier desktop UI in Process Explorer

When investigating network activities under Windows, live forensics usually have a first look at the task manager and the network monitor. Here you can see two connections of the ZeroTier One adapter on which low traffic can be seen.

This is similar in other operating systems, although SDN adapters are not always present in the overviews if, for example, they are running with different user rights or are disguised as other tasks. Temporary deactivation of virtual network adapters can also be set up very easily under Windows or other operating systems, so that the adapter may only be active for a few minutes a day.

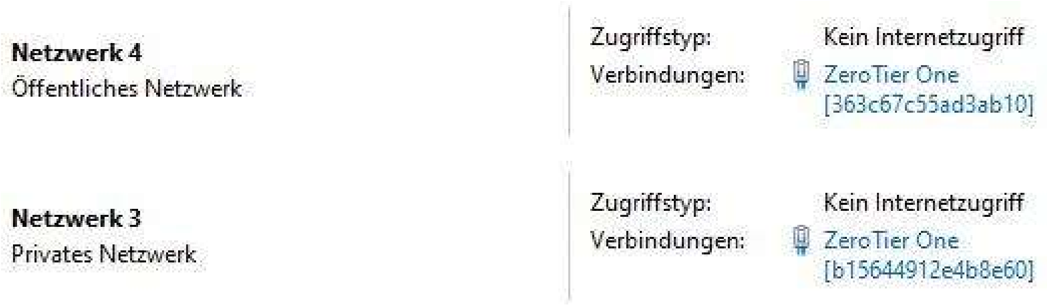


Ethernet
ZeroTier One [363c67c5.
Ges.: 16,0 Empf.: 0 KBit/

Ethernet
ZeroTier One [b1564...
Ges.: 0 Empf.: 0 KBit/s

Fig. 11: Screenshot of two available ZeroTier One Network devices in parallel

For experimental purposes, the ZeroTier network adapter was installed and put into operation on a Windows 11 operating system. A connection to the SDN was possible without any problems. A parallel connection to a second SDN could also be established. This shows the comprehensive configuration options of an SDN without any specialist knowledge. SDN can be configured, switched off and on as desired. Routing to different network areas or bridging connections is also easily possible.



Netzwerk 4
Öffentliches Netzwerk

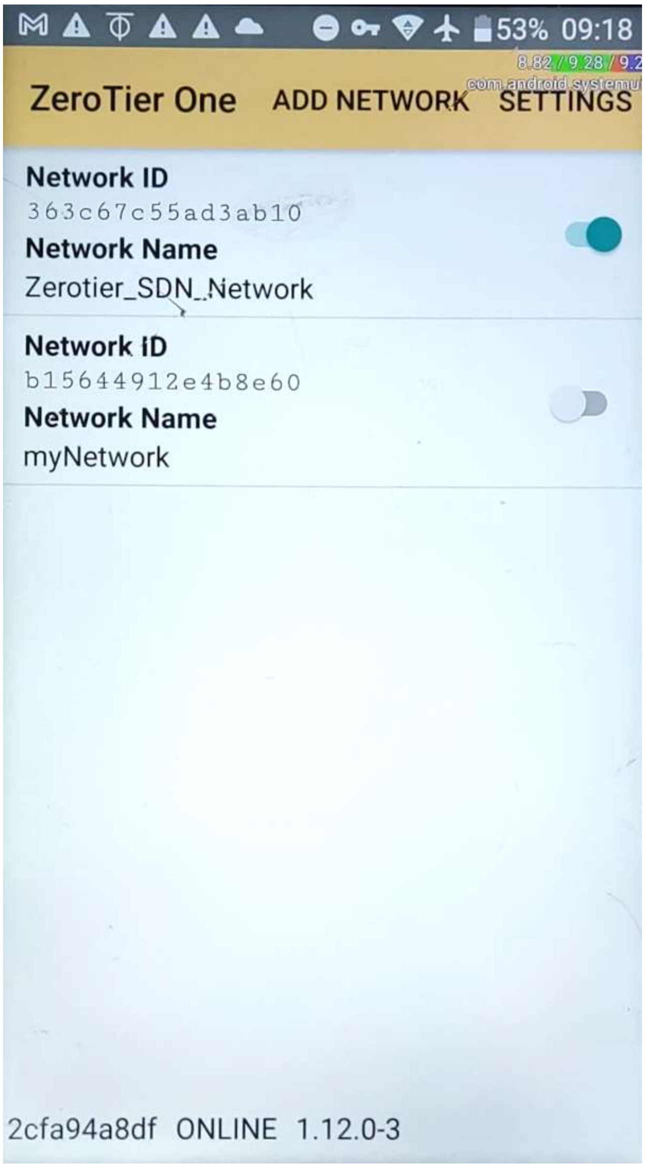
Zugriffstyp: Kein Internetzugriff
Verbindungen: ZeroTier One [363c67c55ad3ab10]

Netzwerk 3
Privates Netzwerk

Zugriffstyp: Kein Internetzugriff
Verbindungen: ZeroTier One [b15644912e4b8e60]

Fig. 12: Screenshot of two ZeroTier adapter in network control center in MS Windows

There are also apps for Apple App Store and Android's Play Store as client applications. For investigation purposes, the ZeroTier One app was installed on a rooted Android device and connected to the test network. A parallel setup with a second SDN was also possible without any problems. All other devices in the SDN could then be addressed via the device. After adjusting the options, it was also possible to route to the Internet.



Network ID
363c67c55ad3ab10
Network Name
Zerotier_SDN_Network

Network ID
b15644912e4b8e60
Network Name
myNetwork

2cfa94a8df ONLINE 1.12.0-3

Fig. 13: Screenshot of an Android device (Kali Nethunter) with two ZeroTier SDN connections

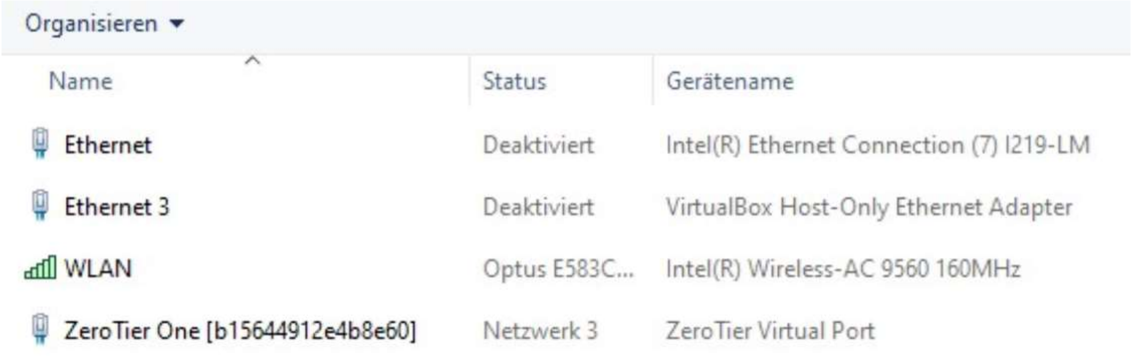
As you can see from the explanations, the software for connecting to the ZeroTier network can be easily installed on different devices. Older devices (from Android 6) or operating systems from Windows 7 can also be connected in this way. And each device runs its own software version, which can be more or less recognized by an outsider (forensic expert). Connections can also be quickly established or disabled using the user interfaces or the command line under Linux. This can also be done automatically using scripts.

The range of functions is very large, so that it is difficult to carry out professional forensics in these structures without special knowledge of these SDN programs.

4.3. Architecture

The ZeroTier network has a typical architecture found in modern SDN. This includes a simple client application and a server model. There is also an administration console that can also be accessed via an Internet browser. On Unix-like operating systems, installation is typically done via the command line, as already shown. An additional network adapter is installed under Windows, which can be addressed and configured via the control panel. In addition, several network adapters can even be installed next to each other for access to different SDNs. This wealth of options and configuration properties makes forensics in SDN quite exciting and complex.

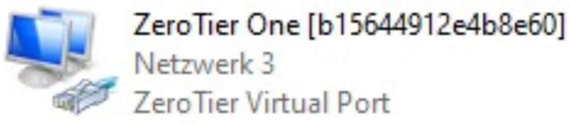
Under Windows, the ZeroTier Controller appears like a normal Ethernet controller in the device manager:



Name	Status	Gerätename
Ethernet	Deaktiviert	Intel(R) Ethernet Connection (7) I219-LM
Ethernet 3	Deaktiviert	VirtualBox Host-Only Ethernet Adapter
WLAN	Optus E583C...	Intel(R) Wireless-AC 9560 160MHz
ZeroTier One [b15644912e4b8e60]	Netzwerk 3	ZeroTier Virtual Port

Fig. 14: Screenshot of typical network adapters in Control Panel under Windows with virtual ZeroTier One Virtual Port

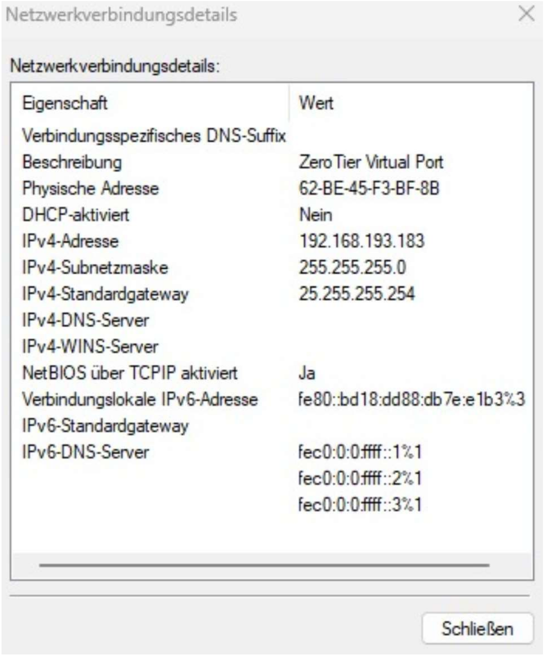
The network adapter is displayed in the control panel with the usual configuration options.



ZeroTier One [b15644912e4b8e60]
Netzwerk 3
ZeroTier Virtual Port

Fig. 15: Screenshot of typical ZeroTier One Network adapters in Windows Control Panel

A look at the details shows additional properties of the virtual, software-controlled controller:



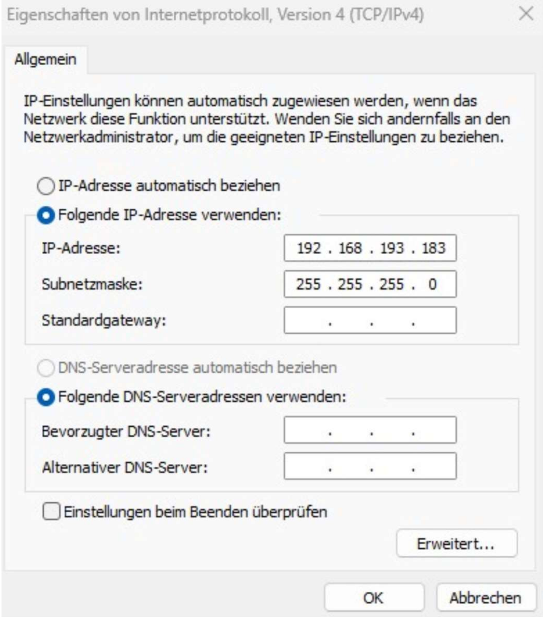
Netzwerkverbindungsdetails

Netzwerkverbindungsdetails:

Eigenschaft	Wert
Verbindungsspezifisches DNS-Suffix	
Beschreibung	ZeroTier Virtual Port
Physische Adresse	62-BE-45-F3-BF-8B
DHCP-aktiviert	Nein
IPv4-Adresse	192.168.193.183
IPv4-Subnetzmaske	255.255.255.0
IPv4-Standardgateway	25.255.255.254
IPv4-DNS-Server	
IPv4-WINS-Server	
NetBIOS über TCPIP aktiviert	Ja
Verbindungslokale IPv6-Adresse	fe80::bd18:dd88:db7e:e1b3%3
IPv6-Standardgateway	
IPv6-DNS-Server	fec0:0:0:ffff::1%1 fec0:0:0:ffff::2%1 fec0:0:0:ffff::3%1

Fig. 16: Screenshot of typical ZeroTier Virtual Port adapter details

Manually assigned IP addresses have already been entered into the IPv4 settings:



Eigenschaften von Internetprotokoll, Version 4 (TCP/IPv4)

Allgemein

IP-Einstellungen können automatisch zugewiesen werden, wenn das Netzwerk diese Funktion unterstützt. Wenden Sie sich andernfalls an den Netzwerkadministrator, um die geeigneten IP-Einstellungen zu beziehen.

○ IP-Adresse automatisch beziehen

● Folgende IP-Adresse verwenden:

IP-Adresse: 192 . 168 . 193 . 183

Subnetzmaske: 255 . 255 . 255 . 0

Standardgateway: . . .

○ DNS-Serveradresse automatisch beziehen

● Folgende DNS-Serveradressen verwenden:

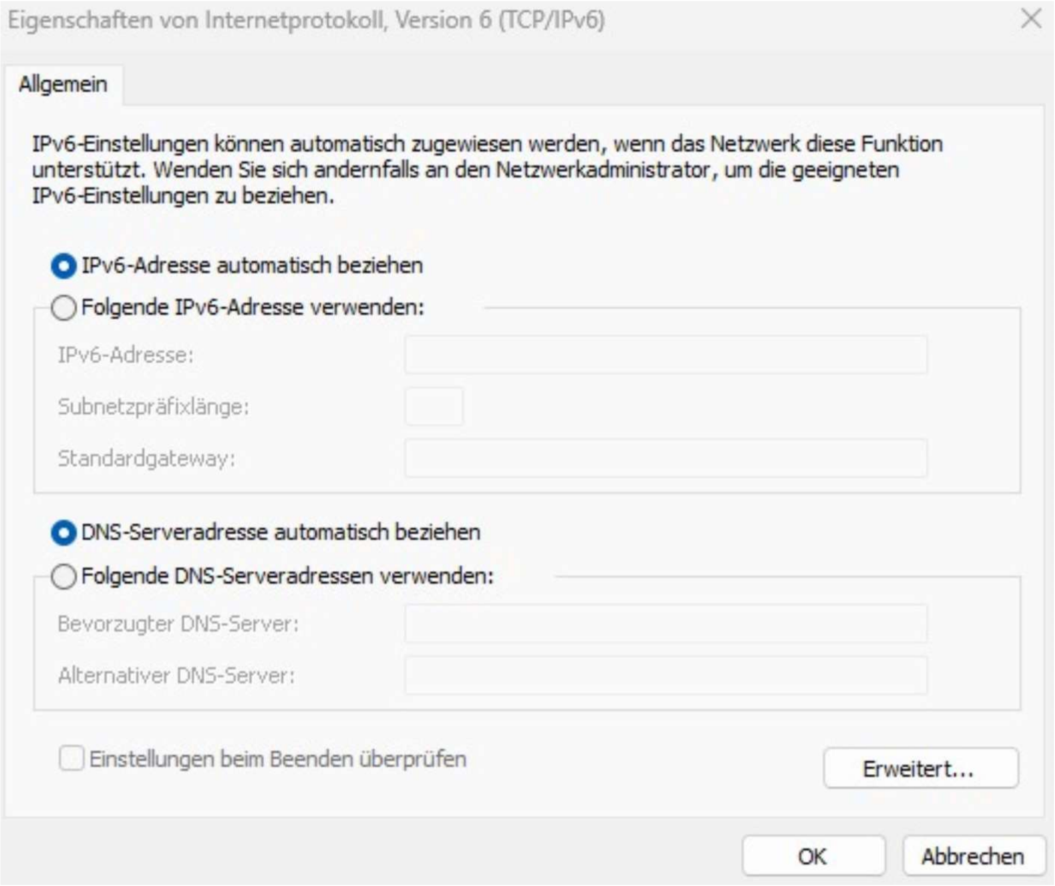
Bevorzugter DNS-Server: . . .

Alternativer DNS-Server: . . .

□ Einstellungen beim Beenden überprüfen

Fig. 17: Screenshot of typical ZeroTier automated IPv4 Address configuration

In this configuration, the IPv6 protocol was automatically configured. On the other hand, it can also be deactivated centrally via the ZeroTier Control Center.



Eigenschaften von Internetprotokoll, Version 6 (TCP/IPv6)

Allgemein

IPv6-Einstellungen können automatisch zugewiesen werden, wenn das Netzwerk diese Funktion unterstützt. Wenden Sie sich andernfalls an den Netzwerkadministrator, um die geeigneten IPv6-Einstellungen zu beziehen.

● IPv6-Adresse automatisch beziehen

○ Folgende IPv6-Adresse verwenden:

IPv6-Adresse:

Subnetzpräfixlänge:

Standardgateway:

● DNS-Serveradresse automatisch beziehen

○ Folgende DNS-Serveradressen verwenden:

Bevorzugter DNS-Server:

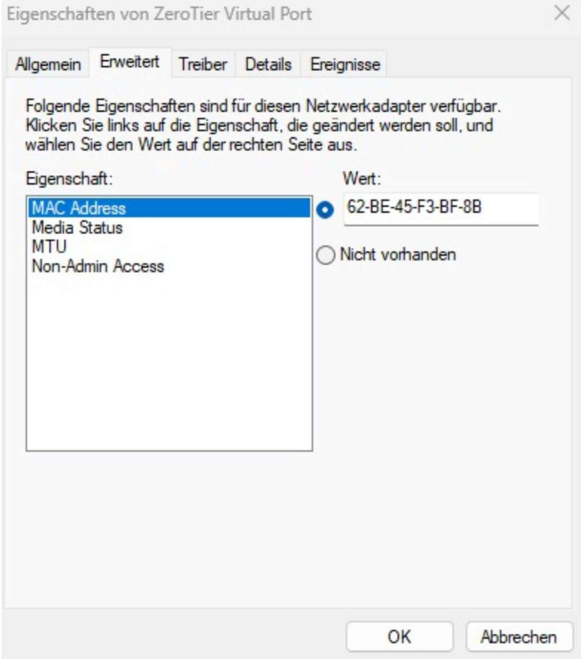
Alternativer DNS-Server:

□ Einstellungen beim Beenden überprüfen

Fig. 18: Screenshot of typical ZeroTier automated IPv6 Address configuration

In the detailed views of the ZeroTier network adapter, you can see typical network settings such as IP addresses, MAC addresses, installation date, version number, etc., which are important in the forensic process, especially when collecting data. Here you can already read out special configurations or set port mirroring functions yourself or redirect the traffic.

You can also easily determine the MAC address that the controller has created under Windows, which is very important for forensic purposes, e.g. to allocate computers in the network. However, it should not be forgotten that MAC addresses can also be generated dynamically, especially in SDN.



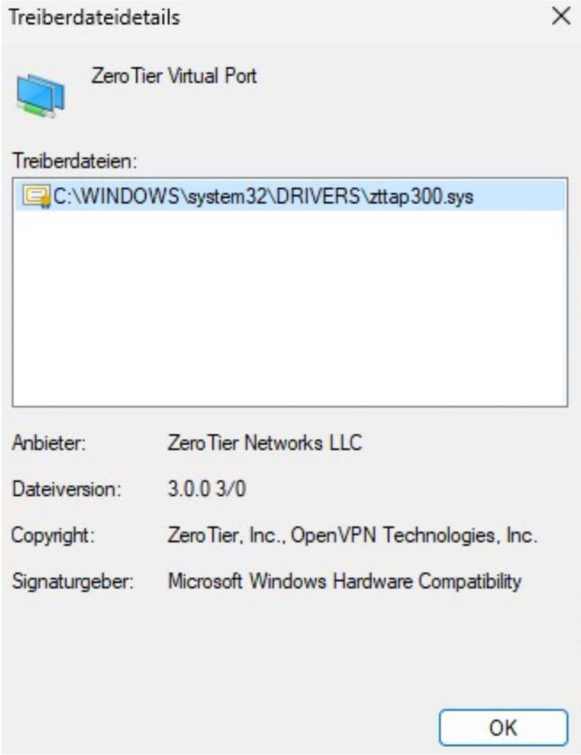
Folgende Eigenschaften sind für diesen Netzwerkadapter verfügbar. Klicken Sie links auf die Eigenschaft, die geändert werden soll, und wählen Sie den Wert auf der rechten Seite aus.

Eigenschaft:
- MAC Address
- Media Status
- MTU
- Non-Admin Access

Wert:
● 62-BE-45-F3-BF-8B
○ Nicht vorhanden

Fig. 19: Screenshot of typical ZeroTier automated MAC Address configuration

The driver information is also often important and can provide information about which version is used and when the network was put into operation.



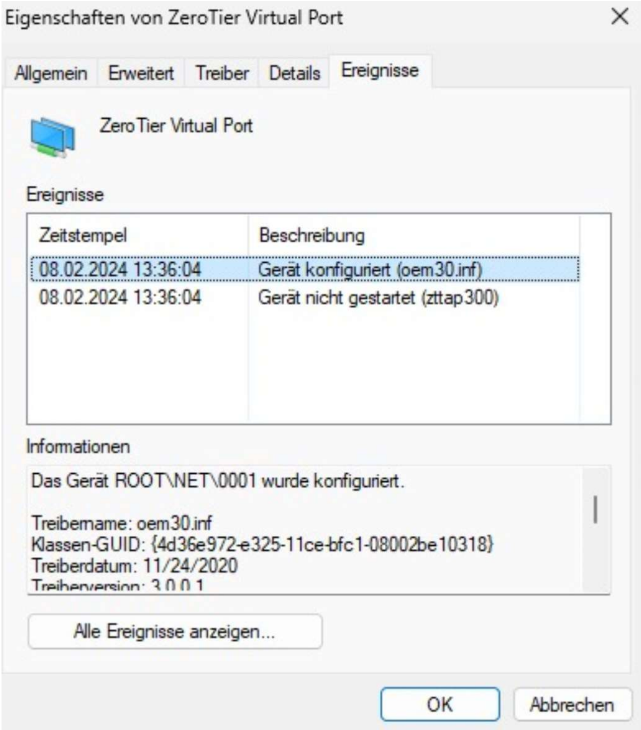
ZeroTier Virtual Port

Treiberdateien:
C:\WINDOWS\system32\DRIVERS\zttap300.sys

Anbieter: ZeroTier Networks LLC
Dateiversion: 3.0.0 3/0
Copyright: ZeroTier, Inc., OpenVPN Technologies, Inc.
Signaturgeber: Microsoft Windows Hardware Compatibility

Fig. 20: Screenshot of ZeroTier Virtual Port driver Information

Based on the driver information, you can see that the ZeroTier SDN network driver was installed on February 8, 2024 at around 1:36 p.m.



Zeitstempel	Beschreibung
08.02.2024 13:36:04	Gerät konfiguriert (oem30.inf)
08.02.2024 13:36:04	Gerät nicht gestartet (zttap300)

Das Gerät ROOT\NET\0001 wurde konfiguriert.

Treibername: oem30.inf
Klassen-GUID: {4d36e972-e325-11ce-bfc1-08002be10318}
Treiberdatum: 11/24/2020
Treiberversion: 3.0.0.1

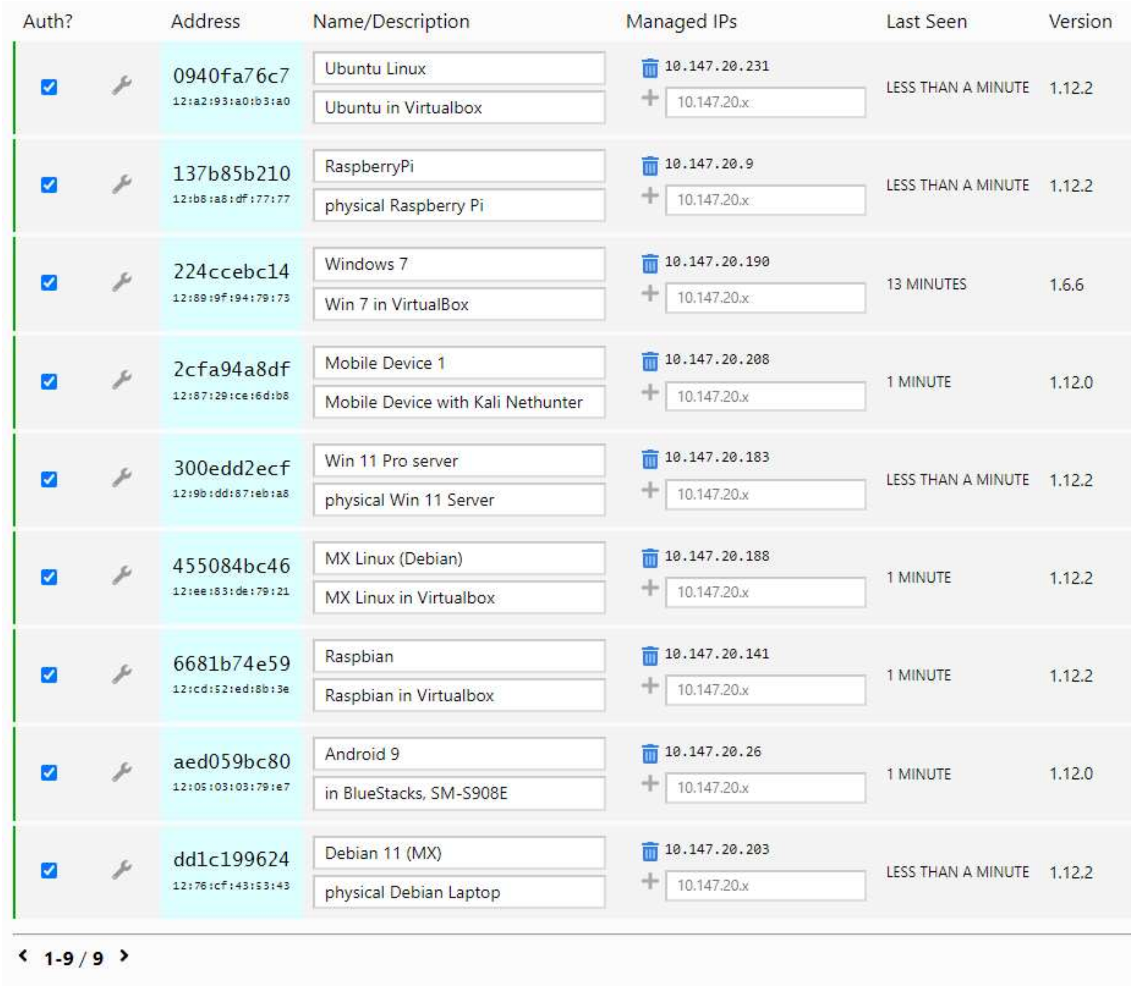
Fig. 21: Screenshot of ZeroTier Virtual Port driver configuration information

As shown above, important information about the more detailed properties of the network adapters can be obtained under Windows in particular.

Important information is also contained in the web interface of the ZeroTier network's configuration centre.

It should be noted that it can be extremely important to be able to access the central control unit, especially when investigating SDNs. The network rules and settings can be viewed in the control centre, new participants can be added or excluded, or security rules can be defined. This investigation option should first be prioritized before any network queries are carried out and, if possible, excluded. Last but not least, security mechanisms can be switched off in the control centre or bandwidths can be increased in order to carry out further investigations.

Next, an overview of the structure of the test ZeroTier network with 9 network participants is shown in the ZeroTier web interface:



Auth?		Address	Name/Description	Managed IPs	Last Seen	Version
☑		0940fa76c7 12:a2:93:a0:b3:a0	Ubuntu Linux Ubuntu in Virtualbox	10.147.20.231 10.147.20.x	LESS THAN A MINUTE	1.12.2
☑		137b85b210 12:b8:a8:df:77:77	RaspberryPi physical Raspberry Pi	10.147.20.9 10.147.20.x	LESS THAN A MINUTE	1.12.2
☑		224ccebc14 12:89:9f:94:79:73	Windows 7 Win 7 in VirtualBox	10.147.20.190 10.147.20.x	13 MINUTES	1.6.6
☑		2cfa94a8df 12:87:29:ce:6d:b8	Mobile Device 1 Mobile Device with Kali Nethunter	10.147.20.208 10.147.20.x	1 MINUTE	1.12.0
☑		300edd2ecf 12:9b:dd:87:eb:a8	Win 11 Pro server physical Win 11 Server	10.147.20.183 10.147.20.x	LESS THAN A MINUTE	1.12.2
☑		455084bc46 12:ee:83:de:79:21	MX Linux (Debian) MX Linux in Virtualbox	10.147.20.188 10.147.20.x	1 MINUTE	1.12.2
☑		6681b74e59 12:cd:52:ed:8b:3e	Raspbian Raspbian in Virtualbox	10.147.20.141 10.147.20.x	1 MINUTE	1.12.2
☑		aed059bc80 12:05:03:03:79:e7	Android 9 in BlueStacks, SM-S908E	10.147.20.26 10.147.20.x	1 MINUTE	1.12.0
☑		dd1c199624 12:76:cf:43:53:43	Debian 11 (MX) physical Debian Laptop	10.147.20.203 10.147.20.x	LESS THAN A MINUTE	1.12.2

Fig. 22: Screenshot of the ZeroTier administration web interface

In the ZeroTier web interface, you can clearly see the clients involved with address information, descriptions, uptime, driver version, etc. Network throughputs can also be displayed live here, and security settings such as whitelisting or blacklisting can be made.

Despite this, the network structures can also be clearly displayed using forensic network analysis with Zenmap, even if you do not have access to the web interface:

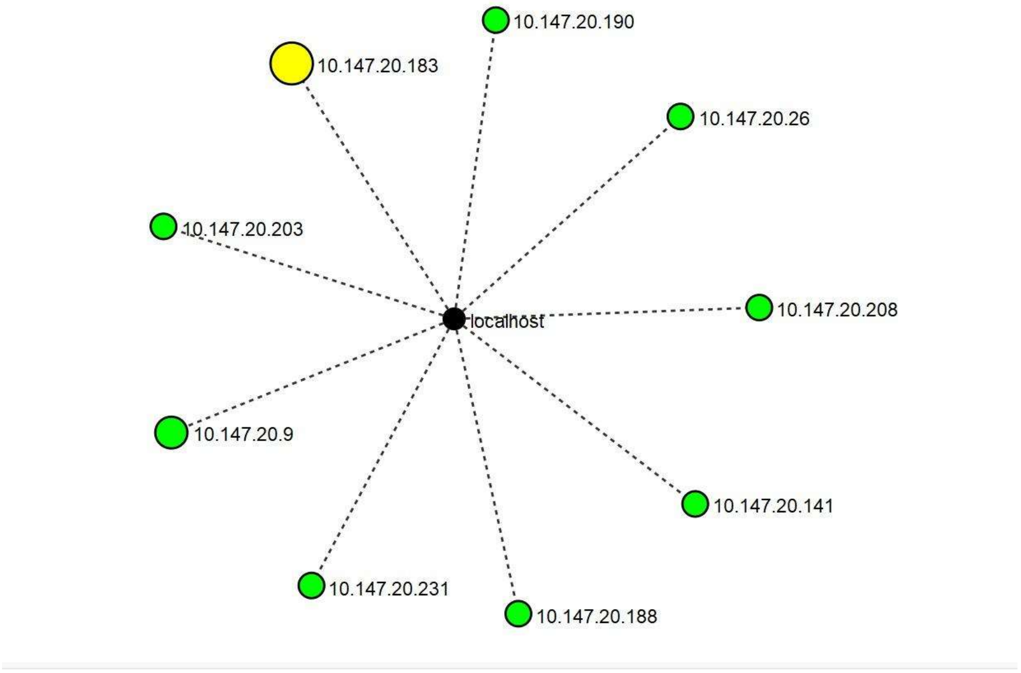
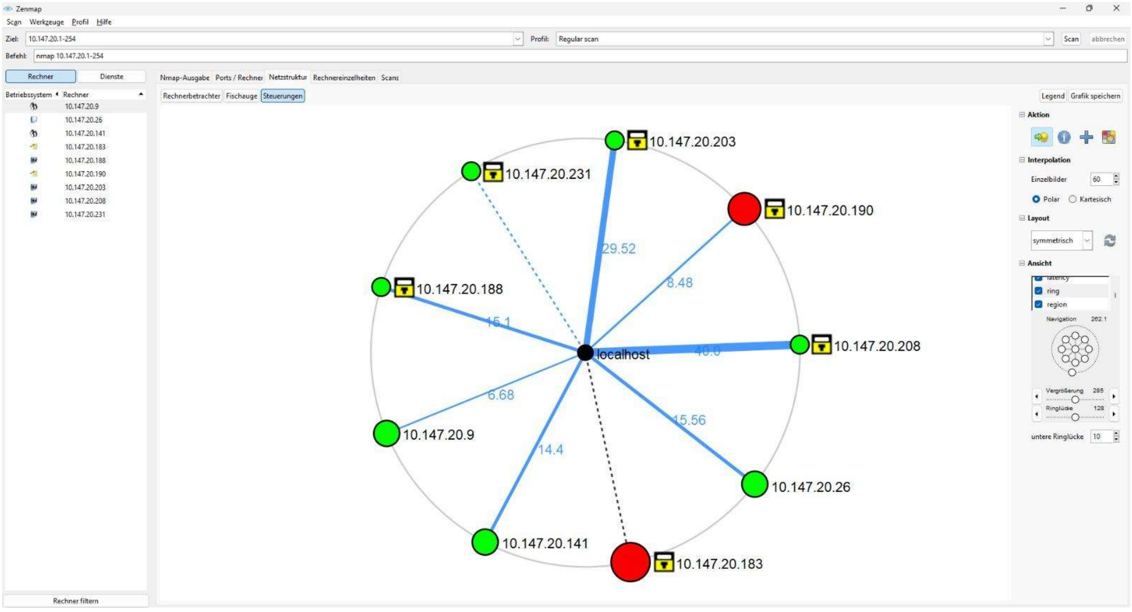


Fig. 23: Screenshot of Zenmap graphical output of the ZeroTier SDN

Using Zenmap - a modification of nmap under Windows - overview graphics of the network structure can be created in a manageable amount of time, which is very helpful for obtaining initial information or if further information about the existing network is missing.


Ziel: 10.147.20.1-254

Profil: Regular scan

Befehl: nmap 10.147.20.1-254

Betriebssystem	Rechner
	10.147.20.9
	10.147.20.26
	10.147.20.141
	10.147.20.183
	10.147.20.188
	10.147.20.190
	10.147.20.203
	10.147.20.208
	10.147.20.231

Einzelbilder 60

● Polar ○ Kartesisch

Layout: symmetrisch

☑ latency
☑ ring
☑ region

Navigation 262.1

Vergrößerung 285

Ringlücke 128

untere Ringlücke 10

Fig. 24: Screenshot of the graphical scan result of Zenmap

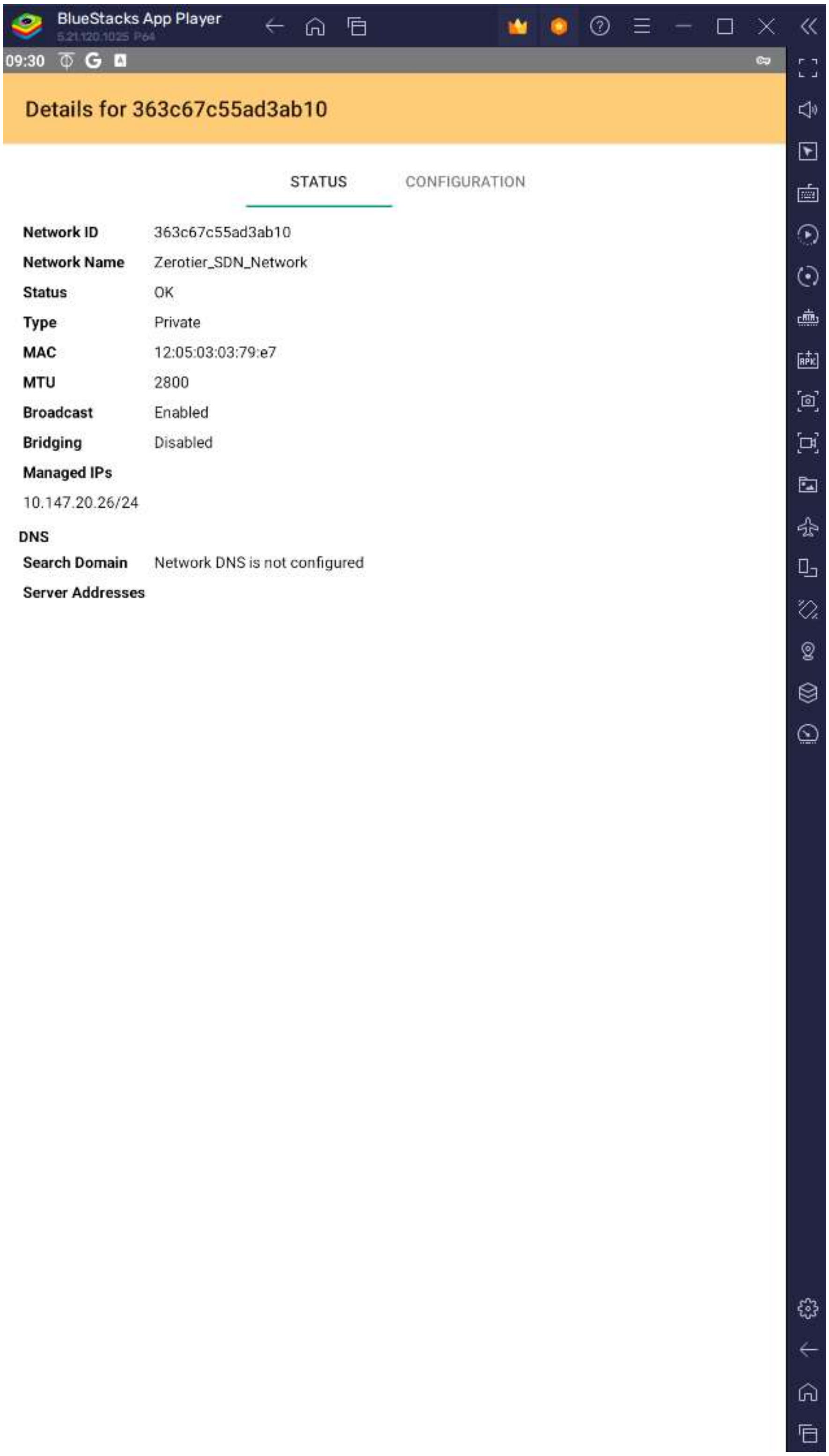
Additional network scans can be carried out in parallel using Zenmap to obtain operating system information or port information about running services.

In this phase, care must be taken not to trigger excessive network stress. Security mechanisms can be triggered here that separate the query computer from the network - and therefore no longer useful information can be received. In addition, these scans sometimes take a long time (a few hours in the test setup) and only then provide important results with which new forensic tasks have to be planned.

Exotic setup configurations can also be carried out using the ZeroTier adapters. In the test setup, an Android device was emulated under BlueStacks[4], on which a ZeroTier client was then installed. The installation went smoothly, and the device was immediately accessible.

Almost all Android apps can be installed on the virtual mobile device under BlueStacks. Typical network applications, such as an http server, ftp server, network scanner, etc. run in conjunction with the installed ZeroTier network adapter in the defined network. This means that outsiders can no longer see that it is a virtual device. For network forensics in SDN, this implies that hardware-based devices may be suspected when in fact they are virtual devices. There are few limits to deception and camouflage in SDN. On the contrary, attackers may find it easier to hide and mask virtual network devices in SDN than in traditional network structures.

[4] BlueStacks Android Emulator App Player www.bluestacks.com



Details for 363c67c55ad3ab10

Network ID	363c67c55ad3ab10
Network Name	Zerotier_SDN_Network
Status	OK
Type	Private
MAC	12:05:03:03:79:e7
MTU	2800
Broadcast	Enabled
Bridging	Disabled
Managed IPs	
10.147.20.26/24	
DNS	
Search Domain	Network DNS is not configured
Server Addresses	

Fig. 25: Screenshot of Virtual Android Device in BlueStacks with ZeroTier SDN

An HTTP server could also be installed on the virtual device, as well as an FTP server. The experiment showed that different services could also be set up on virtual devices that provide files or websites exclusively in the SDN.

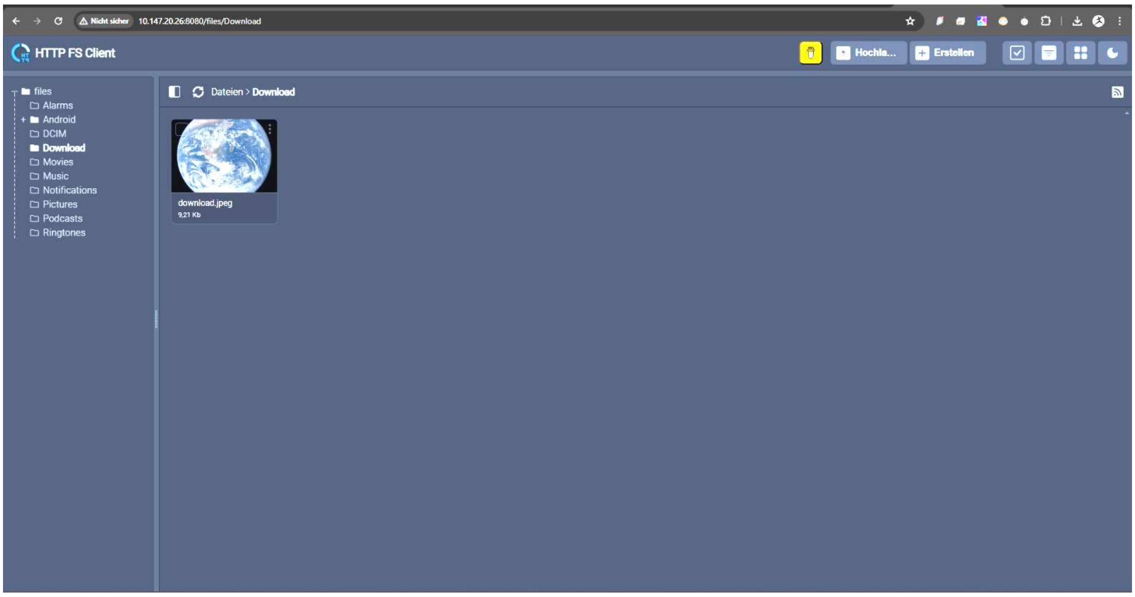
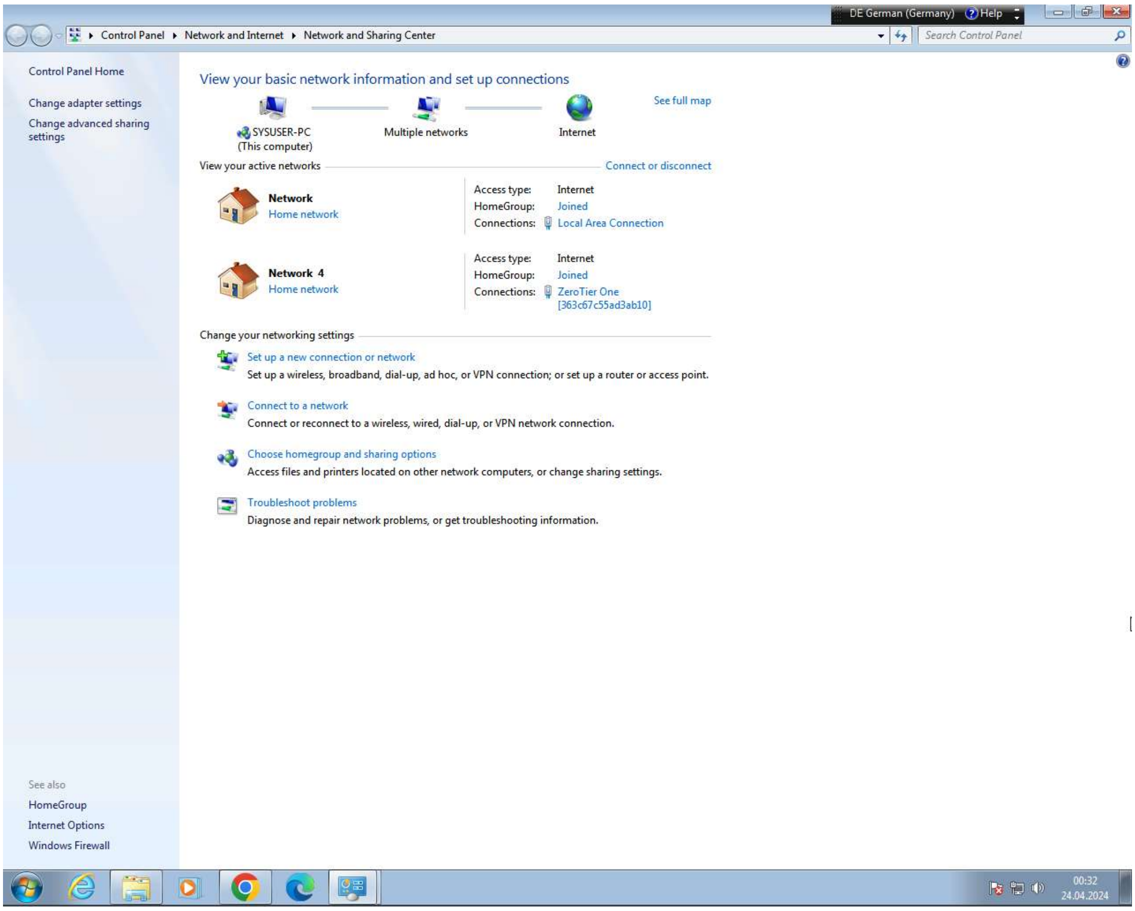


Fig. 26: Screenshot of an http server for picture sharing running on a virtual android device in the ZeroTier SDN

For testing purposes, an HTTP server was set up to offer images for download in SDN. The installed http server shows that any functions can be provided in the network. As an observer, you would not suspect at first glance that the server is running on a virtual Android device.

But older devices can also be upgraded using ZeroTier and ported to an SDN. This also opens up additional possibilities and further functions such as adaptive encryption of network traffic, VPN tunnelling, virtualization, etc. for old systems.

SDN functions can also be provided on older 32-bit operating systems. A virtual Windows 7 system was equipped with SDN functionality for test purposes and put into operation. Further SDN functions such as encrypted file sharing, RDP, etc. could also be implemented with this system.



View your basic network information and set up connections

View your active networks

Network
Home network

Access type: Internet
HomeGroup: Joined
Connections: Local Area Connection

Network 4
Home network

Access type: Internet
HomeGroup: Joined
Connections: ZeroTier One [363c67c55ad3ab10]

Change your networking settings

Set up a new connection or network
Set up a wireless, broadband, dial-up, ad hoc, or VPN connection; or set up a router or access point.

Connect to a network
Connect or reconnect to a wireless, wired, dial-up, or VPN network connection.

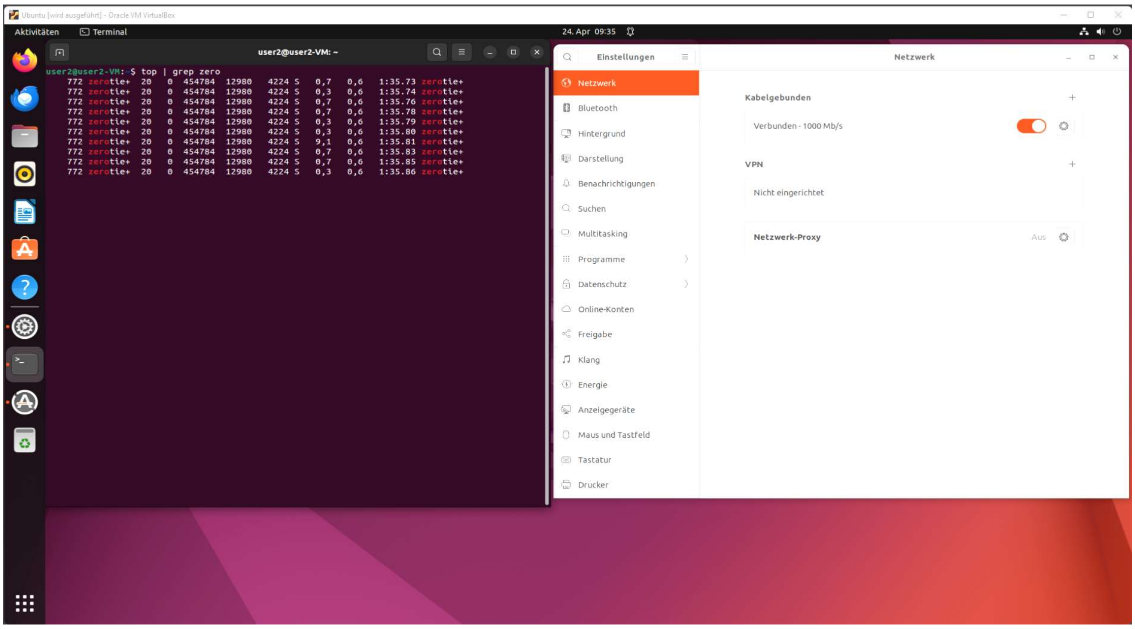
Choose homegroup and sharing options
Access files and printers located on other network computers, or change sharing settings.

Troubleshoot problems
Diagnose and repair network problems, or get troubleshooting information.

Fig. 27: Screenshot of Windows 7 Ultimate with ZeroTierOne Virtual Network

This means that advanced SDN cannot only be used on current operating systems. Older devices can also be made available in these network structures - which is probably very dubious for security reasons. Besides, it shows that the forensic scientist also has to reckon with older computer systems and may receive additional access here.

A classic example is setting up the ZeroTier SDN on an Ubuntu Linux. However, the network processes are not visible at first glance. To do this, you have to use the process manager on the command line, which lists the running network processes here.



```
user2@user2-VM:~$ top | grep zero
    772 zerotie+  20   0  454784  12980   4224 S   0,7   0,6   1:35.73 zerotie+
    772 zerotie+  20   0  454784  12980   4224 S   0,3   0,6   1:35.74 zerotie+
    772 zerotie+  20   0  454784  12980   4224 S   0,7   0,6   1:35.76 zerotie+
    772 zerotie+  20   0  454784  12980   4224 S   0,7   0,6   1:35.78 zerotie+
    772 zerotie+  20   0  454784  12980   4224 S   0,3   0,6   1:35.79 zerotie+
    772 zerotie+  20   0  454784  12980   4224 S   0,3   0,6   1:35.80 zerotie+
    772 zerotie+  20   0  454784  12980   4224 S   9,1   0,6   1:35.81 zerotie+
    772 zerotie+  20   0  454784  12980   4224 S   0,7   0,6   1:35.83 zerotie+
    772 zerotie+  20   0  454784  12980   4224 S   0,7   0,6   1:35.85 zerotie+
    772 zerotie+  20   0  454784  12980   4224 S   0,3   0,6   1:35.86 zerotie+
```

Netzwerk

Kabelgebunden

Verbunden - 1000 Mb/s

VPN

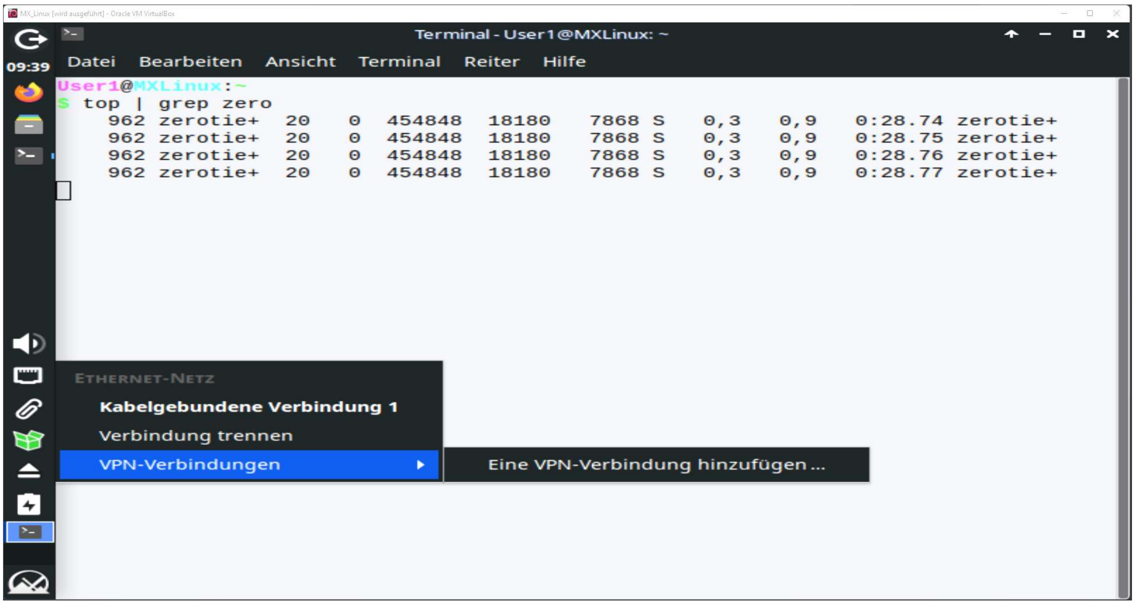
Nicht eingerichtet

Netzwerk-Proxy Aus

Fig. 28: Screenshot of Ubuntu with running zerotier cli process

The SDN processes can therefore be easily hidden in the system - especially if you don't immediately know what you are looking for.

A classic Debian Linux is also suitable for experienced users and high-performance network applications.

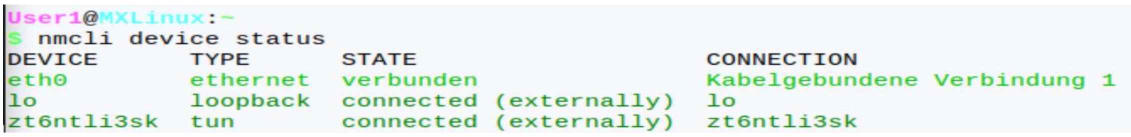

MX_Linux [wird ausgeführt] - Oracle VM VirtualBox
Terminal - User1@MXLinux: ~
09:39
Datei
Bearbeiten
Ansicht
Terminal
Reiter
Hilfe
ETHERNET-NETZ
Kabelgebundene Verbindung 1
Verbindung trennen
VPN-Verbindungen
Eine VPN-Verbindung hinzufügen ...

```
User1@MXLinux:~
$ top | grep zero
    962 zerotie+  20   0  454848  18180   7868 S   0,3   0,9   0:28.74 zerotie+
    962 zerotie+  20   0  454848  18180   7868 S   0,3   0,9   0:28.75 zerotie+
    962 zerotie+  20   0  454848  18180   7868 S   0,3   0,9   0:28.76 zerotie+
    962 zerotie+  20   0  454848  18180   7868 S   0,3   0,9   0:28.77 zerotie+
```

Fig. 29: Screenshot of Debian (MX) Linux with zerotier cli process

Even under Debian you have to use the command line to display the processes. The image above shows four running instances of the ZeroTier Adapter, but there is only one network connection to the ZeroTier SDN.

The network connections can then be displayed more clearly on the command line using the Network Manager (nmcli).



```
User1@MXLinux:~
$ nmcli device status
DEVICE       TYPE      STATE                   CONNECTION
eth0         ethernet  verbunden               Kabelgebundene Verbindung 1
lo           loopback  connected (externally)  lo
zt6ntli3sk   tun       connected (externally)  zt6ntli3sk
```

Fig. 30: Screenshot of nmcli command on Linux with connected ZeroTier SDN

The previous figures are intended to show that it is not always easy to detect SDN network adapters in a running system. These can then be masked and hidden so that they are hardly visible even to experts. If time-controlled connections are implemented, network connections can be easily overlooked, even by experts.

To summarize the structure and architecture of SDN, it can once again be said that these are highly adaptable and dynamic network structures.

Operating systems, hardware and all possible software options can be combined. This of course has an impact on network security as well as on possible forensics in such SDNs. Old or unpatched operating systems are always a point of attack for criminal hackers or computer security experts.

A deep understanding of the structure, architecture and functionality of SDN is always a good prerequisite for successful investigations in such advanced network structures.

4.4. Characteristics and Restrictions

In the ZeroTier SDN, resembling modern SDNs, a simple client application and server model coexist with an accessible administration console, accessible through web browsers. Installation on Unix-like systems generally involves command-line operations, whereas on Windows, it requires the addition of a network adapter, configurable via the control panel. The ability to install multiple network adapters concurrently for accessing various SDNs introduces complexity to SDN forensics [Sp17].

With SDN, you can often observe centralised control of the network using a single control instance, such as a control server with a web interface. Forensic analysis can be greatly facilitated by accessing this interface. Without access to the web administration interface, no basic settings in the network can be changed. In contrast, third parties could change network settings there that are no longer traceable. As part of forensic work in SDN, access to this administration interface is therefore urgently needed.

In addition, many network resources are virtualized in SDN, although their structure in the operating system is very similar to standard network components. Precisely because virtualization processes are already taking place here, such as access to a hypervisor or similar, this property can make subsequent virtualization in the forensic process very difficult.

Programmable and automatable network adapters are also a typical feature of an SDN. An SDN network adapter can be timed so that it is temporarily accessible via one or the other network address. Virtual hardware addresses can also be easily programmed in SDN. MAC addresses often become relative, and it is precisely these properties that must be taken into account in the forensic process.

Security protocols or logging mechanisms are often adapted via this dynamic programmability of SDN, which can make the evaluation of this data extremely complex.

Let's have a quick look at some of the peculiarities of SDN. This involves classic forensic methods such as mirroring and NIC promiscuous mode [Ke06].

Mirroring in SDN:

In traditional networking environments where switches and routers operate independently, port mirroring is a common practice in network forensics for data collection. In the ZeroTier Network, the architecture is fundamentally different. The SDN separates the control plane from the data plane, centralising network control and programmatically managing network behaviour through software.

While port mirroring as traditionally implemented (particularly through hardware interconnects) may not be directly applicable in SDN, similar functionalities can be achieved through different means. In SDN, traffic forwarding and monitoring can be controlled centrally through software-defined controllers. Rather than configuring port mirroring on individual switches, administrators can programmatically direct network traffic to specific monitoring or forensic devices using SDN controllers.

Therefore, while the concept of port mirroring may not be directly applicable in SDN, similar monitoring and traffic redirection functionalities can be achieved through SDN-based approaches, providing greater flexibility and control over network traffic.

Promiscuous Mode:

A promiscuous NIC (network interface card) or NIC TAP (test access point) can theoretically function in an SDN, but the implementation can be challenging. In the ZeroTier SDN, network functions are decoupled from the physical infrastructure and centrally managed, making it difficult to place traditional hardware TAPs. However, virtual TAPs or virtual NICs could be deployed in virtualized SDN environments to monitor and capture network traffic. These solutions require careful configuration and integration into the SDN management framework. No direct way to incorporate a NIC TAP or similar was found in the ZeroTier network. Obviously, there is no provision for centrally controlling monitoring or recording network traffic.

In a closed and secured software network environment, it will hardly be possible for us to mirror in an SDN. Putting a network adapter into promiscuous mode is also likely to be impossible in most SDN cases.

In many forensic guides, such as NIST or BSI or ENISA, hardware-based network taps are presented as a standard means of network forensics [Eu19].

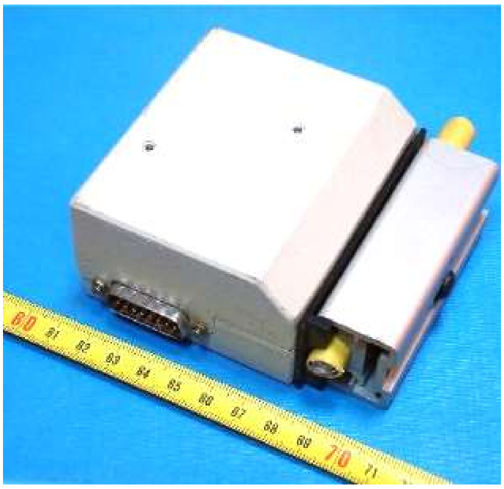


Fig. 31: A vampire tap presented by ENISA 2019 [Eu19]

These guidelines basically talk about data collection on cables or via WLAN. Nonetheless, this approach is completely obsolete in the ZeroTier Network.

To summarise, let's say that hardware-based network taps and forensic path recorders are essentially useless in SDN. Currently, no unified methods for data collection in SDN have been proposed by higher-level institutions. Unfortunately, due to the unique characteristics of SDN, we have to say goodbye to many traditional methods of network forensics [Bu06].

Nevertheless, current forensic techniques obviously work in the ZeroTier network, which are predominantly based on hacking methods and rely on various small tools. We will have a closer look at these tools and practices from a practical perspective in the next chapter.

4.5. Investigations with Practical Tools

In the practical investigation of SDN, individual tools play a greater role than in traditional network forensics, especially because live forensics is significantly more important in the investigation of SDN.

These small help programs from a forensic scientist play an important role in the preparation process for forensic investigations. There are actually only a handful of tools that have become established internationally among forensic experts and have the necessary technical testing and legal standing [Al11].

As already mentioned, a compilation of established forensic programs can be found at https://www.dftoolscatalogue.eu/, with various authorities from Europol to the Swiss police relying on this reliable catalogue [Eu23].

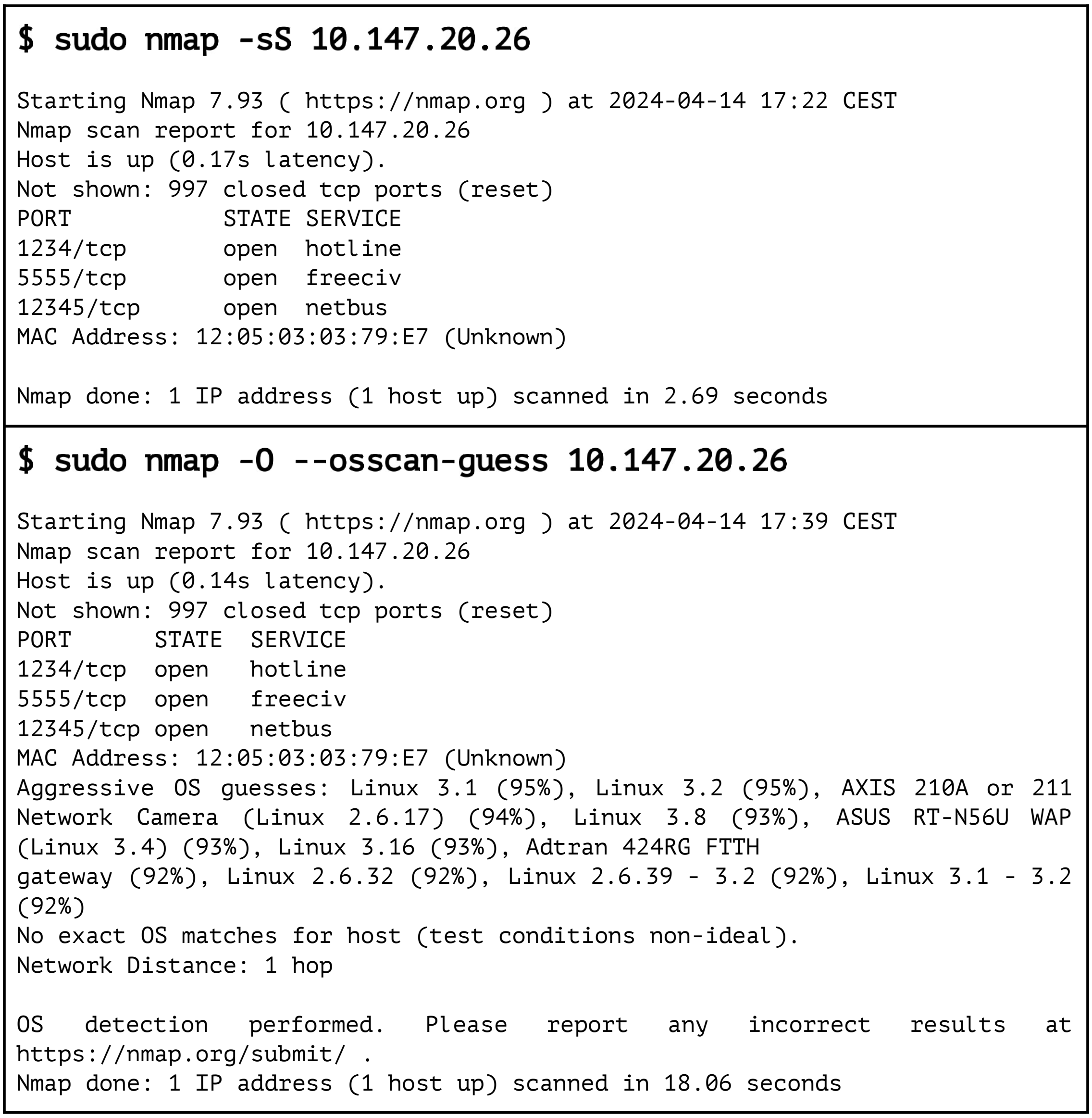
In order to examine parts of the ZeroTier network, individual meaningful tools were selected, and typical examination routines were carried out in the network.

The order of the applications was chosen purely at random or based on experience. Here it also depends on the practical individual case and the importance of the investigation objectives as to which programs or methods are carried out first.

4.5.1. NMAP

Nmap offers a variety of configuration options and delivers fast and often detailed results in all network environments, which are important for the forensic process.

In this case, a targeted nmap scan was performed on a network address to obtain information about the specific host. The result of the specific NMAP Scan in the ZeroTier network (target: Android 9 Device) shows as following:



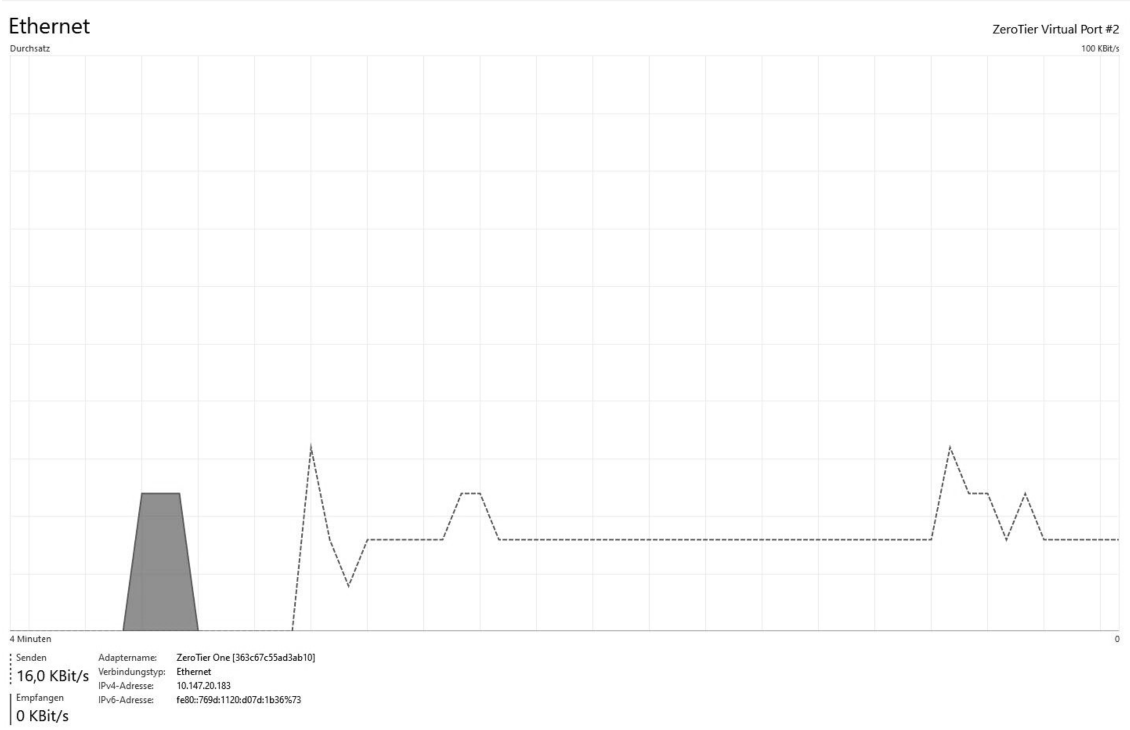
```
$ sudo nmap -sS 10.147.20.26

Starting Nmap 7.93 ( https://nmap.org ) at 2024-04-14 17:22 CEST
Nmap scan report for 10.147.20.26
Host is up (0.17s latency).
Not shown: 997 closed tcp ports (reset)
PORT           STATE SERVICE
1234/tcp       open  hotline
5555/tcp       open  freeciv
12345/tcp      open  netbus
MAC Address: 12:05:03:03:79:E7 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 2.69 seconds
```

```
$ sudo nmap -O --osscan-guess 10.147.20.26

Starting Nmap 7.93 ( https://nmap.org ) at 2024-04-14 17:39 CEST
Nmap scan report for 10.147.20.26
Host is up (0.14s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE  SERVICE
1234/tcp  open   hotline
5555/tcp  open   freeciv
12345/tcp open   netbus
MAC Address: 12:05:03:03:79:E7 (Unknown)
Aggressive OS guesses: Linux 3.1 (95%), Linux 3.2 (95%), AXIS 210A or 211
Network Camera (Linux 2.6.17) (94%), Linux 3.8 (93%), ASUS RT-N56U WAP
(Linux 3.4) (93%), Linux 3.16 (93%), Adtran 424RG FTTH
gateway (92%), Linux 2.6.32 (92%), Linux 2.6.39 - 3.2 (92%), Linux 3.1 - 3.2
(92%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.06 seconds
```

Fig. 32: Output of a specific nmap-scan with os-detection in an SDN

Based on the output, you can find out more information about open ports or even the running operating system. Of course, this information needs to be cross-checked. Linux 3.1 was suspected here, although the device is an Android 9 device.

We can clearly see that typical network scans with nmap in SDN technically work in the same way as in other network structures.



Senden
16,0 KBit/s

Empfangen
0 KBit/s

Adaptername: ZeroTier One [363c67c55ad3ab10]
Verbindungstyp: Ethernet
IPv4-Adresse: 10.147.20.183
IPv6-Adresse: fe80::769d:1120:d07d:1b36%73

Fig. 33: Screenshot of network stats during the nmap scans (~ 16 kbit/s)

In the network load view above, you can see that a typical scan with nmap at around 16 kbit/s does not produce excessive network traffic. However, such scans can overload networks and cause them to react critically, especially if complex security mechanisms are implemented.

4.5.2. RAM-Dump with winpmem

The need to save a RAM dump while analysing an SDN will only be briefly discussed.

In individual cases, the selection of the appropriate tool is often the responsibility of the on-site forensic expert.

But only by quickly backing up the RAM contents passwords, running processes, login information, etc. can be saved.

This backup method is therefore always recommended as part of network forensics, even if impairment of the running system during the RAM backup cannot be ruled out.

Investigative authorities worldwide use the FreeTool program interface (https://thefreetoolproject.eu/), under which a version of winpmem runs, depending on the configuration.

The entire current memory content is saved alongside other typical information for later, in-depth forensic analysis.

```
C:\>winpmem_mini_x64_rc2.exe memdump.dmp

WinPmem64
Extracting driver
Loaded Driver
The system time is: 08:26:44
Will generate a RAW image
 - buffer_size_: 0x1000
CR3: 0x00001AE002
 7 memory ranges:
Start 0x00001000 - Length 0x00057000
Start 0x00059000 - Length 0x00046000
Start 0x00100000 - Length 0xC7034000
Start 0xC713B000 - Length 0x00458000
Start 0xC79C3000 - Length 0x15B32000
Start 0xDEFFF000 - Length 0x00001000
Start 0x100000000 - Length 0x71F000000
max_physical_memory_ 0x81f000000
Acquitision mode PTE Remapping
Padding from 0x00000000 to 0x00001000
pad
 - length: 0x1000

00% 0x00000000 .
copy_memory
 - start: 0x1000
 - end: 0x58000
```

```
00% 0x00001000 .
Padding from 0x00058000 to 0x00059000
pad
 - length: 0x1000

00% 0x00058000 .
copy_memory
 - start: 0x59000
 - end: 0x9f000

00% 0x00059000 .
Padding from 0x0009F000 to 0x00100000
pad
 - length: 0x61000

00% 0x0009F000 .
copy_memory
 - start: 0x100000
 - end: 0xc7134000

00% 0x00100000 x..x..............................................
.
.
.

The system time is: 08:28:56
Driver Unloaded.
```

Fig. 34: Output of memory acquisition with winpmem

This is an important point when examining SDN network structures. The interfaces and the surrounding hardware must be included in the investigation. And this happens in the first step by backing up the RAM contents. This step is often missing in classic forensic network investigations because it is assigned to normal computer forensics. Hardware configurations can often be traced afterwards, but SDN configurations or processes can be completely lost when the system is switched off.

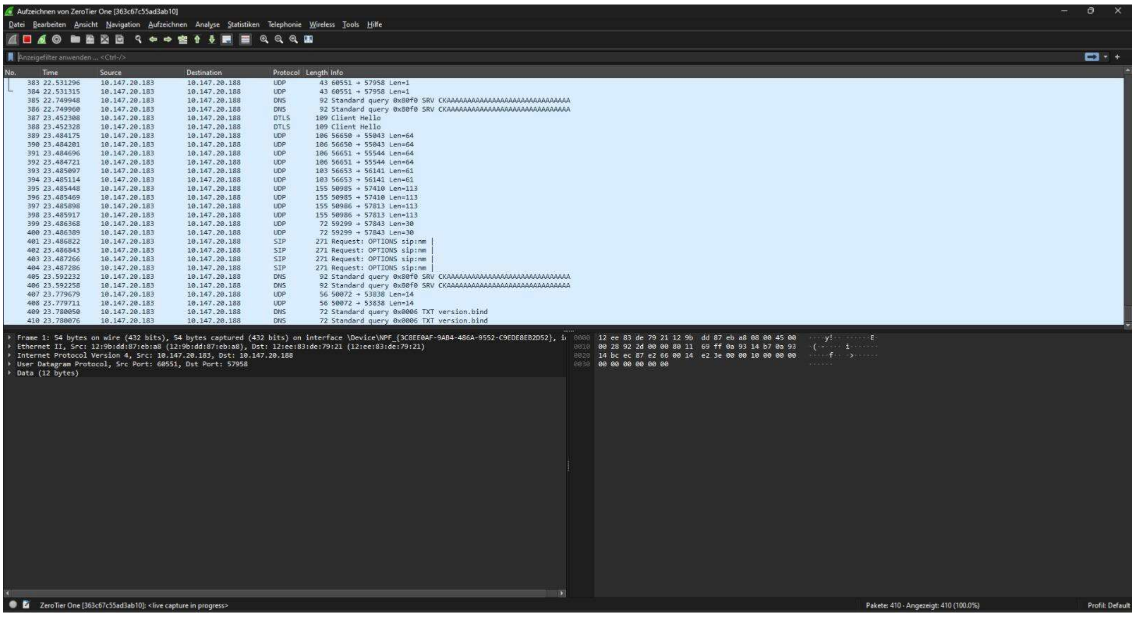
On the other hand, there is one catch that should not be ignored when backing up the RAM: This process is system-critical and can also lead to a system crash in individual cases. A consideration is always necessary here.

In the system examined, there were no critical system impairments even after multiple RAM backups.

4.5.3. WIRESHARK

Wireshark is a powerful standard tool for analysing computer networks and comprehensively storing network traffic. For capturing, the resource-saving command line version is expressly recommended, or the smaller and more powerful command line tool tcpdump.

To collect data, the tools usually have to be run with elevated rights. This is similar on Windows, MacOS or Linux. Finally, the network adapter can be selected. However, the configurations often do not provide for the selection of multiple network adapters. In real cases, this can lead to problems, so that only one network adapter can be successfully recorded.


Aufzeichnen von ZeroTier One [363c67c55ad3ab10]
Datei
Bearbeiten
Ansicht
Navigation
Aufzeichnen
Analyse
Statistiken
Telephonie
Wireless
Tools
Hilfe
Anzeigefilter anwenden ... <Ctrl-/>
ZeroTier One [363c67c55ad3ab10]: <live capture in progress>
Pakete: 410 · Angezeigt: 410 (100.0%)
Profil: Default

```
No.     Time        Source          Destination     Protocol Length Info
    383 22.531296   10.147.20.183   10.147.20.188   UDP        43 60551 → 57958 Len=1
    384 22.531315   10.147.20.183   10.147.20.188   UDP        43 60551 → 57958 Len=1
    385 22.749948   10.147.20.183   10.147.20.188   DNS        92 Standard query 0x80f0 SRV CKAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    386 22.749960   10.147.20.183   10.147.20.188   DNS        92 Standard query 0x80f0 SRV CKAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    387 23.452308   10.147.20.183   10.147.20.188   DTLS      109 Client Hello
    388 23.452328   10.147.20.183   10.147.20.188   DTLS      109 Client Hello
    389 23.484175   10.147.20.183   10.147.20.188   UDP       106 56650 → 55043 Len=64
    390 23.484201   10.147.20.183   10.147.20.188   UDP       106 56650 → 55043 Len=64
    391 23.484696   10.147.20.183   10.147.20.188   UDP       106 56651 → 55544 Len=64
    392 23.484721   10.147.20.183   10.147.20.188   UDP       106 56651 → 55544 Len=64
    393 23.485097   10.147.20.183   10.147.20.188   UDP       103 56653 → 56141 Len=61
    394 23.485114   10.147.20.183   10.147.20.188   UDP       103 56653 → 56141 Len=61
    395 23.485448   10.147.20.183   10.147.20.188   UDP       155 50985 → 57410 Len=113
    396 23.485469   10.147.20.183   10.147.20.188   UDP       155 50985 → 57410 Len=113
    397 23.485898   10.147.20.183   10.147.20.188   UDP       155 50986 → 57813 Len=113
    398 23.485917   10.147.20.183   10.147.20.188   UDP       155 50986 → 57813 Len=113
    399 23.486368   10.147.20.183   10.147.20.188   UDP        72 59299 → 57843 Len=30
    400 23.486389   10.147.20.183   10.147.20.188   UDP        72 59299 → 57843 Len=30
    401 23.486822   10.147.20.183   10.147.20.188   SIP       271 Request: OPTIONS sip:nm |
    402 23.486843   10.147.20.183   10.147.20.188   SIP       271 Request: OPTIONS sip:nm |
    403 23.487266   10.147.20.183   10.147.20.188   SIP       271 Request: OPTIONS sip:nm |
    404 23.487286   10.147.20.183   10.147.20.188   SIP       271 Request: OPTIONS sip:nm |
    405 23.592232   10.147.20.183   10.147.20.188   DNS        92 Standard query 0x80f0 SRV CKAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    406 23.592258   10.147.20.183   10.147.20.188   DNS        92 Standard query 0x80f0 SRV CKAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    407 23.779679   10.147.20.183   10.147.20.188   UDP        56 50072 → 53838 Len=14
    408 23.779711   10.147.20.183   10.147.20.188   UDP        56 50072 → 53838 Len=14
    409 23.780050   10.147.20.183   10.147.20.188   DNS        72 Standard query 0x0006 TXT version.bind
    410 23.780076   10.147.20.183   10.147.20.188   DNS        72 Standard query 0x0006 TXT version.bind
```

```
› Frame 1: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{3C8EE0AF-9AB4-486A-9552-C9EDE8EB2D52}, i
› Ethernet II, Src: 12:9b:dd:87:eb:a8 (12:9b:dd:87:eb:a8), Dst: 12:ee:83:de:79:21 (12:ee:83:de:79:21)
› Internet Protocol Version 4, Src: 10.147.20.183, Dst: 10.147.20.188
› User Datagram Protocol, Src Port: 60551, Dst Port: 57958
› Data (12 bytes)
```

```
0000  12 ee 83 de 79 21 12 9b  dd 87 eb a8 08 00 45 00   ····y!·· ······E·
0010  00 28 92 2d 00 00 80 11  69 ff 0a 93 14 b7 0a 93   ·(·-···· i·······
0020  14 bc ec 87 e2 66 00 14  e2 3e 00 00 10 00 00 00   ·····f·· ·>······
0030  00 00 00 00 00 00                                  ······
```

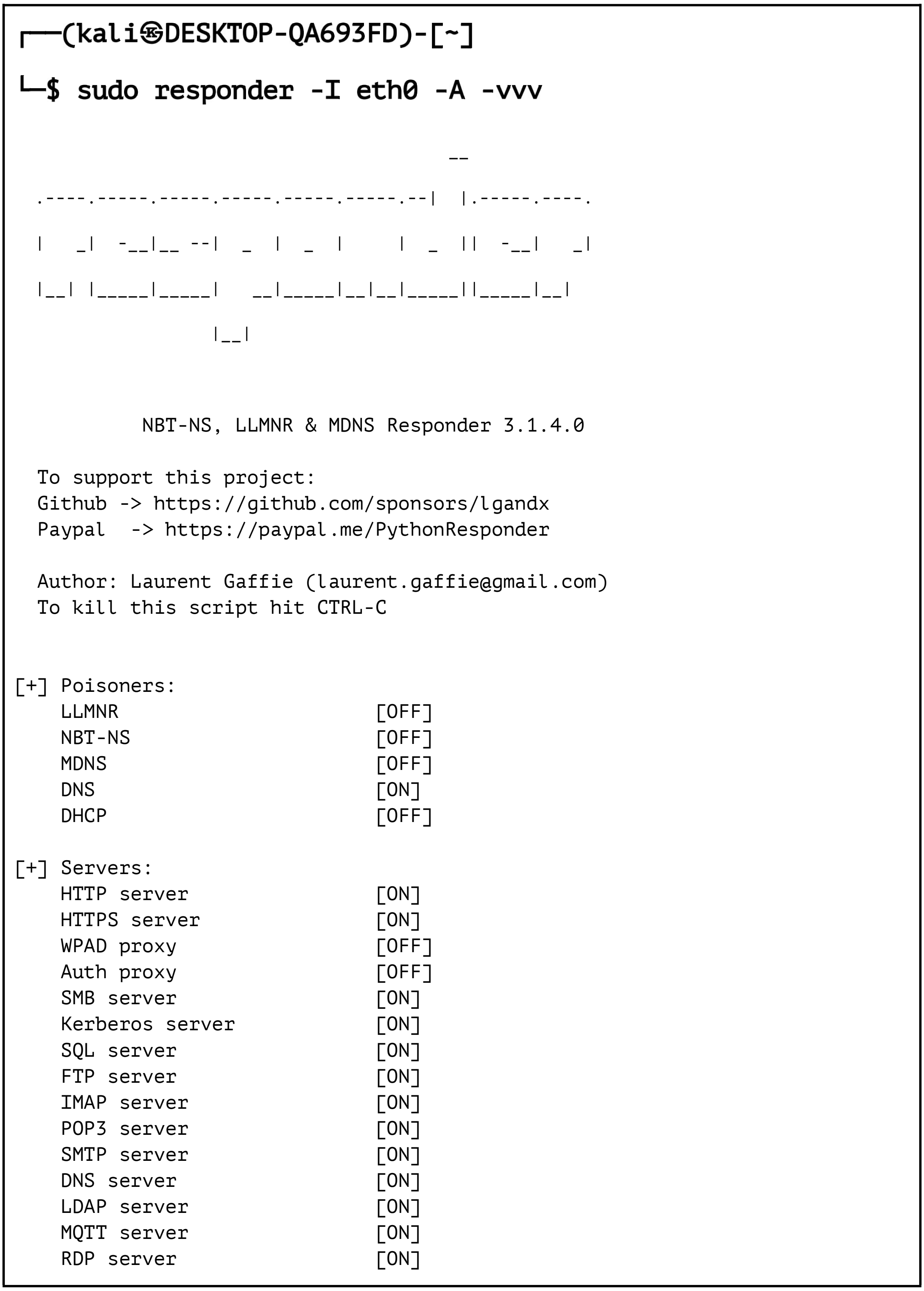
Fig. 35: Screenshot of the wireshark gui with capturing packets in the ZeroTier Network

Wireshark is known for its clear interface for viewing network traffic. There are also a number of filter options available here.

Thanks to its comprehensive protocol support, Wireshark can be used to carry out detailed network analyses if you are an advanced user. Wireshark is also a powerful tool for investigating SDN, as individual interfaces are displayed, for instance, and live analysis is possible directly. Nevertheless, Wireshark requires extensive specialist knowledge and experience in order to be used effectively.

4.5.4. Responder

With Responder in analysis mode you can also obtain useful information in SDN, as the tool in the configuration used here does not send requests, but only receives and evaluates data.



```
┌──(kali㉿DESKTOP-QA693FD)-[~]
└─$ sudo responder -I eth0 -A -vvv

                                         __
  .----.-----.-----.-----.-----.-----.--|  |.-----.----.
  |   _|  -__|__ --|  _  |  _  |     |  _  ||  -__|   _|
  |__| |_____|_____|   __|_____|__|__|_____||_____|__|
                   |__|


           NBT-NS, LLMNR & MDNS Responder 3.1.4.0

  To support this project:
  Github -> https://github.com/sponsors/lgandx
  Paypal  -> https://paypal.me/PythonResponder

  Author: Laurent Gaffie (laurent.gaffie@gmail.com)
  To kill this script hit CTRL-C


[+] Poisoners:
    LLMNR                      [OFF]
    NBT-NS                     [OFF]
    MDNS                       [OFF]
    DNS                        [ON]
    DHCP                       [OFF]

[+] Servers:
    HTTP server                [ON]
    HTTPS server               [ON]
    WPAD proxy                 [OFF]
    Auth proxy                 [OFF]
    SMB server                 [ON]
    Kerberos server            [ON]
    SQL server                 [ON]
    FTP server                 [ON]
    IMAP server                [ON]
    POP3 server                [ON]
    SMTP server                [ON]
    DNS server                 [ON]
    LDAP server                [ON]
    MQTT server                [ON]
    RDP server                 [ON]
    DCE-RPC server             [ON]
    WinRM server               [ON]
    SNMP server                [OFF]

[+] HTTP Options:
    Always serving EXE         [OFF]
    Serving EXE                [OFF]
    Serving HTML               [OFF]
    Upstream Proxy             [OFF]

[+] Poisoning Options:
    Analyze Mode               [ON]
    Force WPAD auth            [OFF]
    Force Basic Auth           [OFF]
    Force LM downgrade         [OFF]
    Force ESS downgrade        [OFF]

[+] Generic Options:
    Responder NIC              [eth0]
    Responder IP               [10.147.20.183]
    Responder IPv6             [fe80::215:5dff:fe7a:d6ea]
    Challenge set              [random]
    Don't Respond To Names     ['ISATAP', 'ISATAP.LOCAL']

[+] Current Session Variables:
    Responder Machine Name     [WIN-IG50UGJJ6O0]
    Responder Domain Name      [HVWQ.LOCAL]
    Responder DCE-RPC Port     [47868]

[+] Listening for events...

[Analyze mode: ICMP] You can ICMP Redirect on this network.
[Analyze mode: ICMP] This workstation (10.147.20.183) is not on the same
subnet than the DNS server (10.147.20.1).
[Analyze mode: ICMP] Use `python tools/Icmp-Redirect.py` for more details.
[+] Responder is in analyze mode. No NBT-NS, LLMNR, MDNS requests will be
poisoned.
```

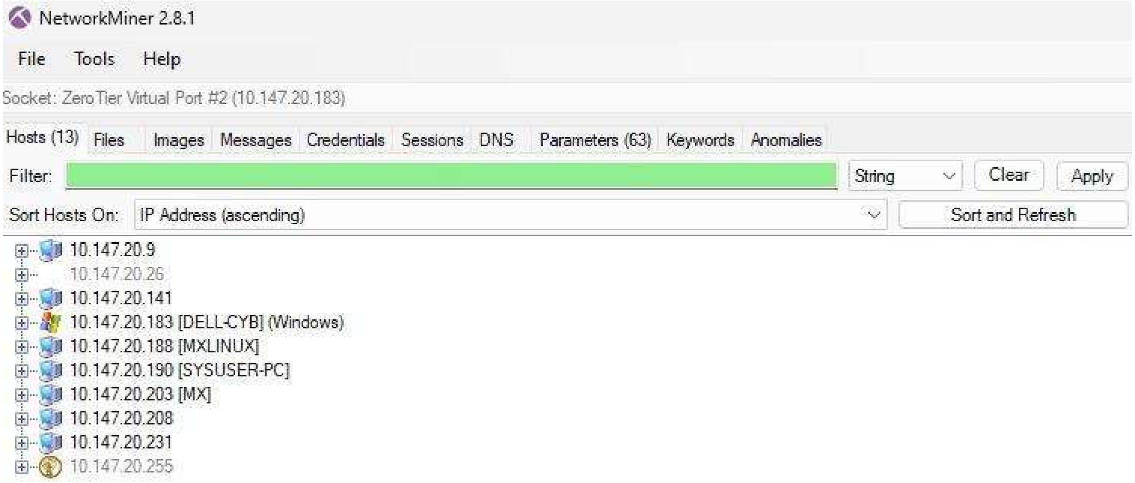
Fig. 36: Output of running Responder in the ZeroTier SDN

Responder listens for typical events in Windows networks and can interpret the traffic. In attack mode, Responder can also obtain hashes from Windows users and actively intervene in the traffic. This way you can gain access to Windows shares, which is also easily possible in the ZeroTier SDN.

4.5.5. Network Miner

Network Miner can also be used for forensic investigations on Windows. There is a free, slightly limited version, which was used here, and a paid professional version of the program, which has a wider range of features. The tool is essentially a swiss army knife for network investigations and can display computers, resources and extract additional information such as user information, files, messages, image data, etc. from network traffic in real time. In ZeroTier SDN, this tool can also be used to obtain a quick network overview. However, it turned out that no content data like files, pictures, messages, etc. was extracted from the network stream - this is probably due to the data encapsulation or encryption used in the SDN data stream.

A professional version of the tool is available to better support these functionalities.



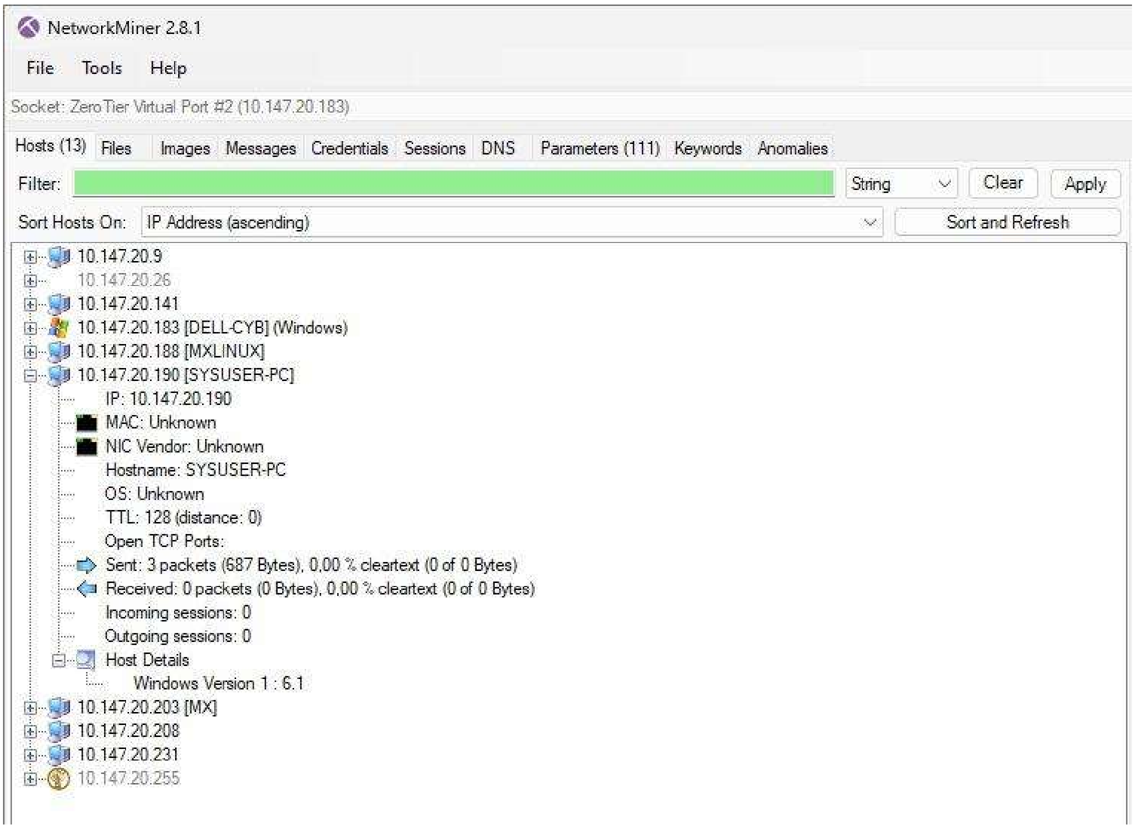
Socket: ZeroTier Virtual Port #2 (10.147.20.183)

Sort Hosts On: IP Address (ascending)

```
10.147.20.9
10.147.20.26
10.147.20.141
10.147.20.183 [DELL-CYB] (Windows)
10.147.20.188 [MXLINUX]
10.147.20.190 [SYSUSER-PC]
10.147.20.203 [MX]
10.147.20.208
10.147.20.231
10.147.20.255
```

Fig. 37: Screenshot of the scan result with Network Miner

As a result, Network Miner provided an overview of all participants in the SDN.



Socket: ZeroTier Virtual Port #2 (10.147.20.183)

Filter: String

Sort Hosts On: IP Address (ascending)

```
⊞ 10.147.20.9
⊞ 10.147.20.26
⊞ 10.147.20.141
⊞ 10.147.20.183 [DELL-CYB] (Windows)
⊞ 10.147.20.188 [MXLINUX]
⊟ 10.147.20.190 [SYSUSER-PC]
    IP: 10.147.20.190
    MAC: Unknown
    NIC Vendor: Unknown
    Hostname: SYSUSER-PC
    OS: Unknown
    TTL: 128 (distance: 0)
    Open TCP Ports:
    Sent: 3 packets (687 Bytes), 0,00 % cleartext (0 of 0 Bytes)
    Received: 0 packets (0 Bytes), 0,00 % cleartext (0 of 0 Bytes)
    Incoming sessions: 0
    Outgoing sessions: 0
  ⊟ Host Details
        Windows Version 1 : 6.1
⊞ 10.147.20.203 [MX]
⊞ 10.147.20.208
⊞ 10.147.20.231
⊞ 10.147.20.255
```

Fig. 38: Screenshot of detailed view of the ZeroTier SDN

Additional information about the hosts was also displayed, such as Windows version, open ports, etc.

These investigations were found as part of live forensics and would probably not have been able to be presented in such detail as part of post-mortem forensics or a later packet analysis.

Like many other tools, Network Miner only works in live networks, which again underlines the importance of live forensics.

Resources found via the network can also be called up immediately and analysed with a click.

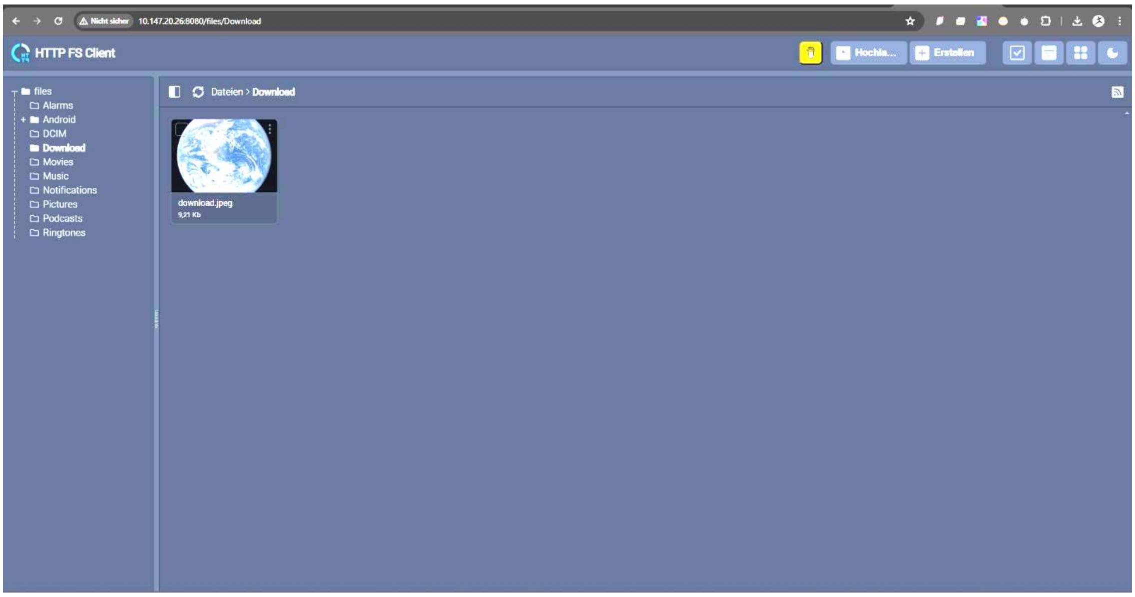


Fig. 39: Screenshot of an open http server found with Network Miner on a virtual android device in the ZeroTier SDN

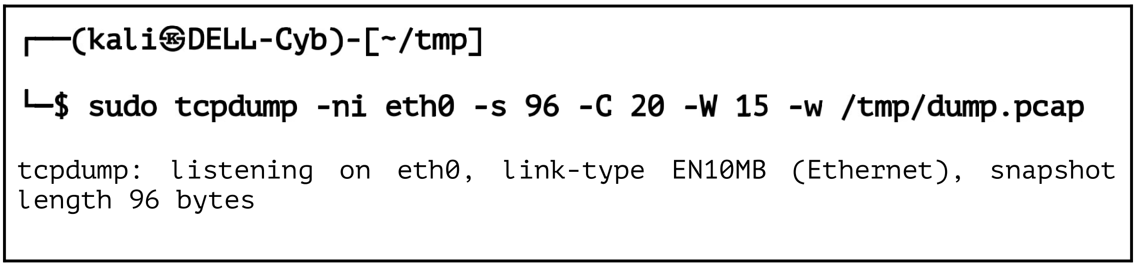
Network Miner was also able to find the intentionally open HTTP server on a virtual Android device and open the homepage. It was shown that, on the one hand, investigations are possible in ZeroTier SDN and that special servers can also be set up quickly. This configuration would be suitable for easily publishing a homepage or distributing files on an unsuspicious or virtual device.

Unfortunately, Network Miner only works effectively on Windows, so many operating systems are left out. Nevertheless, the results are surprisingly good and clearly show the most important network components in the ZeroTier SDN. The program was also able to determine the operating system information of the network participants and open ports or services well in some cases.

4.5.6. Tcpdump

The tcpdump tool is best suited to acquire network data under Linux. So you select the network resource - or select all available network adapters - and you can save the data stream into one or more files.

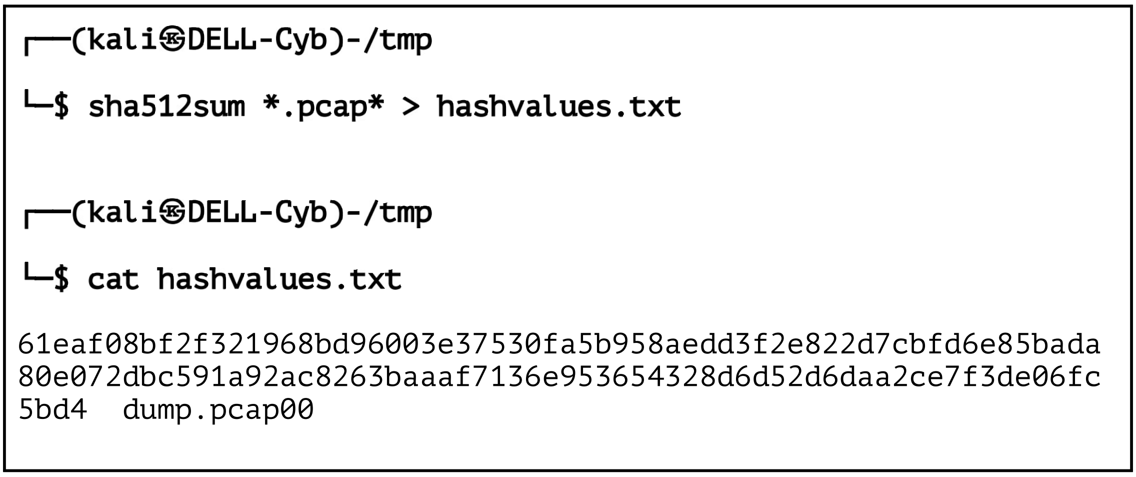
The method of regularly creating smaller files has proven successful, so that the files created can be immediately secured with hash values and can be analysed promptly.



```
┌──(kali㉿DELL-Cyb)-[~/tmp]
└─$ sudo tcpdump -ni eth0 -s 96 -C 20 -W 15 -w /tmp/dump.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot
length 96 bytes
```

Fig. 40: Working tcpdump on eth0 with creating .pcap files every 20 MB

Here, you can see that only a few parameters are needed to carry out targeted storage of network data using tcpdump. Even so, the tool requires a confident and advanced use of the command line under Linux.



```
┌──(kali㉿DELL-Cyb)-/tmp
└─$ sha512sum *.pcap* > hashvalues.txt

┌──(kali㉿DELL-Cyb)-/tmp
└─$ cat hashvalues.txt
61eaf08bf2f321968bd96003e37530fa5b958aedd3f2e822d7cbfd6e85bada
80e072dbc591a92ac8263baaaf7136e953654328d6d52d6daa2ce7f3de06fc
5bd4  dump.pcap00
```

Fig. 41: Securing the .pcap-files with sha512

Obtained data can be secured easily using hash algorithms to ensure integrity.[5]

In this example, tcpdump replaces a network TAP, as can be used in classic cable networks. In SDN, a software solution must be used.

[5] Checksums should be created using actual secure hashing algorithms.

4.5.7. Summary and assessment of IT security mechanisms

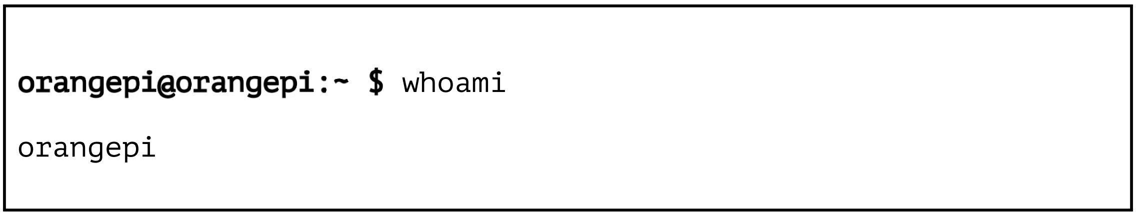
As part of the non-extensive forensic analysis of the structures of the ZeroTier network, astonishing security gaps were found in the standard settings of the network adapters.

As an example, access to an Orange Pi that was connected via the ZeroTier network from a Kali Linux outside this network will be shown here. The Kali Linux is a virtual machine in this network system.

It turned out that although the Linux distribution could not be addressed, the Orangepi could still be accessed from the Kali machine.

```
┌──(Message from Kali developers)
│
│ This is a minimal installation of Kali Linux, you likely
│ want to install supplementary tools. Learn how:
│                                                          ⇒
https://www.kali.org/docs/troubleshooting/common-minimum-setup
/
│
└──(Run: “touch ~/.hushlogin” to hide this message)
┌──(kali㉿DESKTOP-QA693FD)-[~]
└─$ ip addr show

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever

2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
group default qlen 1000
    link/ether 00:15:5d:7a:d6:ea brd ff:ff:ff:ff:ff:ff
    inet 172.26.235.79/20 brd 172.26.239.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:fe7a:d6ea/64 scope link
       valid_lft forever preferred_lft forever

┌──(kali㉿DESKTOP-QA693FD)-[~]
└─$ whoami
kali


┌──(kali㉿DESKTOP-QA693FD)-[~]
└─$ ssh orangepi@orangepi
ssh: Could not resolve hostname orangepi: Name or service not known


┌──(kali㉿DESKTOP-QA693FD)-[~]
└─$ ssh orangepi@192.168.193.21
The authenticity of host '192.168.193.21 (192.168.193.21)' can't be
established.
ED25519 key fingerprint is
SHA256:UGuKm+ebpS4WVYrdMmvfEwCgnxpBRiCfYKgS28xE6A.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.193.21' (ED25519) to the list of known
hosts.
orangepi@192.168.193.21's password:
  ___  ____ ___   _____   _   _____ ____
 / _ \|  _ \_ _| |___ /  | | |_   _/ ___|
| | | | |_) | |    |_ \  | |   | | \___ \
| |_| |  __/| |   ___) | | |___| |  ___) |
 \___/|_|  |___| |____/  |_____|_| |____/


Welcome to Orange Pi buster with Linux 5.10.75-sunxi64
```

```
System load:   1.00 1.00 1.00   Up time:       16 days 20:49
Memory usage:  11 % of 1989MB   IP:            192.168.178.92
CPU temp:      49°C
Usage of /:    32% of 29G
[ General system configuration (beta): orangepi-config ]
Last login: Wed Oct 18 09:11:31 2023 from 192.168.193.106


orangepi@orangepi:~ $ ip addr show



1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever


2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
    link/ether 02:07:de:e7:ad:f6 brd ff:ff:ff:ff:ff:ff
        inet 192.168.178.92/24 brd 192.168.178.255 scope global dynamic
noprefixroute eth0
       valid_lft 533239sec preferred_lft 533239sec
       inet6 2003:c6:5712:b800:ed1f:212f:369b:35ab/64 scope global dynamic
noprefixroute
       valid_lft 6994sec preferred_lft 1116sec
    inet6 fe80::921c:8589:24b1:f12/64 scope link noprefixroute
       valid_lft forever preferred_lft forever


3: wlan0: <NO-CARRIER,BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
pfifo_fast state DORMANT group default qlen 1000
    link/ether 3c:7a:aa:27:d8:dd brd ff:ff:ff:ff:ff:ff


4: zteb4a6h2o: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 2800 qdisc pfifo_fast
state UNKNOWN group default qlen 1000
    link/ether 62:da:ec:59:5e:d4 brd ff:ff:ff:ff:ff:ff
    inet 192.168.193.21/24 brd 192.168.193.255 scope global zteb4a6h2o
       valid_lft forever preferred_lft forever
       inet 192.168.193.188/24 brd 192.168.193.255 scope global secondary
zteb4a6h2o
       valid_lft forever preferred_lft forever
    inet6 fe80::60da:ecff:fe59:5ed4/64 scope link
       valid_lft forever preferred_lft forever
```



```
orangepi@orangepi:~ $ whoami
orangepi
```

Fig. 42: Simple SSH access from outside the SDN (kali) to a computer (orangepi) within the SDN

This example shows that SDN, here in the example of the ZeroTier network, are not silver bullets for IT security. Of course, these networks could also be configured securely, but this is obviously not always the case.

If a popular gaming network has an insecure configuration or existing security gaps, over a million computers could easily be attacked. On the other hand, SDN solutions can be made so secure using end-to-end encryption that attacks on this encryption are currently unsuccessful. The responsibility here probably lies initially with the developers of such software in order to comply with the rules of security by default.

A big advantage of SDN is its dynamism and ability to adapt to the (security) needs of users. Dynamic encryption algorithms can be implemented, as can multifactor authentication, obfuscation, decentralization or dynamic routing. Many providers of SDN (VPN) solutions advertise that their networks are extremely secure and regularly adjust their standards.

The typical approaches and tools used in network forensics also largely work in SDN. Nevertheless, the results are individual and misinterpretations can occur. Scan results are regularly influenced by network separation, and networks can be activated or deactivated with a click or automatically. Virtual environments also do not make forensic work any easier and have an impact on the further course of forensic processes. Finally, the constant software development and dynamics of modern SDN environments regularly make it difficult to adapt forensic approaches.

5. Results

Specific strategies and methods for forensic analysis in SDN were derived on the basis of the classic investigation methods and strategies of network forensics and the corresponding practical methods for investigating SDN.

The strategies developed for investigations in SDN and the methodological approaches for investigations in SDN are presented on the following pages.

5.1. Strategies for Forensics in SDN

Which strategies can be used for clarification, quick information gathering and reliable evidence preservation in SDN will be presented below.

As already mentioned, strategies in the context of this work are comprehensive plans and approaches to achieve higher-level or abstract goals and solve the associated problems. In contrast to this are the concrete methods of network forensics that can be applied specifically in an application to achieve a specific goal or solve a practical problem [Ei17].

Network forensics strategies can be divided into main strategies such as rapid information gathering, damage prevention or effective law enforcement, and sub-strategies such as no network burden, invisible investigations, maximum information gathering as quickly as possible, etc.

In addition to the main strategies, one can define sub-strategies such as invisible investigations, passive investigations and active investigations. These range from discreet and invisible investigations to targeted queries to the network in order to enable quick and high-quality information gathering. Each sub-strategy has its own characteristics and challenges that need to be considered.

Typical main forensics strategies in SDN are listed on the next pages, and their characteristics are briefly explained.

Objectives	Quick presentation of results
Focus	Rapid presentation of forensic results to enable quick decisions.
Features	● Quick data collection ● Fast analysis to identify relevant information ● Less documentation through less information collection ● Comprehensive preservation of evidence at the control interface
Disadvantages	● Short preparation phase ● Small amount of data increases the influence of small errors ● Waiver of individual quality tests ● Legally more vulnerable in hindsight ● Higher risk of network stability degradation ● Direct network access required

Table 1: Objective of quick presentation of results

The main objective of the first strategy is to achieve a quick presentation of forensic results, facilitating rapid decision-making. This involves swift data collection and analysis to identify relevant information, with less documentation required due to reduced information collection. Additionally, comprehensive preservation of evidence at the control interface is emphasised.

Nonetheless, there are several disadvantages to consider. These include a short preparation phase, which may lead to an increased risk of errors. Furthermore, the smaller amount of data collected may render the investigation legally vulnerable in hindsight. Additionally, direct network access is required, posing potential risks to network stability.

Objectives	Effective law enforcement
Focus	Collecting and analysing forensic evidence to support criminal prosecutions.
Features	● Detailed and comprehensive forensic investigation ● Secured and robust evidence ● legally secured and regularly checked ● Interim results for adjusting measures ● Sensitive control of network load through forensics
Disadvantages	● Mistakes have major legal implications ● Slow way of working ● Four eyes principle necessary ● Enormous documentation effort ● Long follow-up and analysis phase ● Very skilled reporting required ● High personnel costs ● Direct network access is often required

Table 2: Objective of effective law enforcement

The primary aim of effective law enforcement in network forensics is to gather and analyse forensic evidence to support criminal prosecutions. This entails conducting thorough and detailed investigations to procure secure and robust evidence, which is legally protected and regularly scrutinized.

Notable features include providing interim results for adjusting measures and carefully controlling network load through forensic techniques. However, there are several drawbacks to consider. Errors in this process can carry significant legal ramifications, and the approach typically involves a slow and methodical working pace, requiring adherence to the four eyes principle and necessitating extensive documentation efforts.

Moreover, the protracted follow-up and analysis phase, coupled with the need for highly skilled reporting, contribute to elevated personnel costs. Additionally, direct network access is often essential for effectively carrying out these law enforcement activities in network forensics.

Objectives	Fast damage defence
Focus	Quickly detecting, analysing and defending against network attacks in order to minimize damage.
Features	● Can be carried out during an IT damage event or cyberattack ● Focused forensic investigation ● Rapid and targeted information gathering ● Effective, automated responsiveness ● Fast delivery of results and response ● No additional network stress caused by forensic activities ● Preventing further damaging effects
Disadvantages	● Often insufficient for a detailed clarification of the incident ● Automation of processes is prone to errors and manipulation ● High demands on hardware and software ● Direct network access is absolutely required

Table 3: Objective of fast damage defence

The main goal of fast damage defence is to promptly detect, analyse, and counter network attacks to minimise harm. This involves swiftly responding to IT damage events or cyberattacks through focused forensic investigations, rapid targeted information gathering, and efficient automated responses.

Key features include the ability to act during ongoing IT damage events or cyberattacks, along with rapid and precise information collection. The strategy also emphasises quick delivery of results and responses without adding stress to the network. Moreover, it aims to prevent further damage by promptly addressing the attack.

This approach may lack detail in clarifying incidents due to its focus on speed. Automation could introduce errors and susceptibility to manipulation. Additionally, it places high demands on hardware and software resources, requiring direct network access for effective implementation.

Objectives	Proactive threat detection and response
Focus	Early detection of potential threats in SDN with automated response and secure mechanisms to damage prevention.
Features	● Continuous Monitoring ● Identifying anomalous patterns and suspicious behaviour ● Machine learning algorithms to detect and display anomalies ● Automated response to suspicious activity ● Fast delivery of results and response ● Additional network stress caused by continuous forensic activities ● Preventive measures to secure the SDN against potential attacks
Disadvantages	● Automation of processes is prone to false positive errors and manipulation ● High requirements on hardware-resources and software ● Potential disruption to normal network operations due to aggressive defence measures ● Complexity in configuring and managing proactive security measures in SDN

Table 4: Objective of proactive threat detection and response

The strategy of proactive threat detection and response in SDN aims to detect potential threats early and deploy automated responses to prevent damage. Key features include continuous monitoring, anomaly detection, machine learning algorithms, and fast response. Despite this, challenges such as false positives, hardware/software requirements, disruption to network operations, and complexity in configuration need to be addressed.

Further strategies often combine with the main strategies already mentioned. These secondary strategies are referred to as sub-strategies and are described in more detail below.

Objectives	Invisible Investigation
Focus	Discreet and invisible investigation to SDN
Features	● No queries are sent on the network ● Investigator is not a participant in the network ● Comprehensive information is collected outside the network structures (e.g. control interface) ● Suitable for long-term examinations ● Little information is available ● Legal hurdles are very low because no network data is read
Disadvantages	● No network data can be read ● The significance of the results is rather low ● Other forensic measures often have to follow ● Legal admissibility must be checked regularly ● Sparse results

Table 5: Sub-Strategy of invisible investigation

The invisible investigation strategy in SDN aims to conduct discreet and undetectable investigations. It focuses on avoiding detection while gathering comprehensive information outside the network structures, making it suitable for long-term examinations. This approach refrains from sending queries on the network and ensures that the investigator remains a non-participant. Nevertheless, due to the limited availability of information, the significance of the results may be low. Despite these advantages, there are disadvantages such as the inability to read network data, leading to sparse results. Additionally, legal admissibility must be regularly checked, and other forensic measures may need to be employed.

Objectives	Sensitive Investigation
Focus	Passive investigations in SDN
Features	● No queries are sent on the network ● No negative network impact ● Comprehensive preservation of evidence at the control interface ● Network traffic is stored ● Live Monitoring possible ● Analysis of raw data ● Normal information density in the network ● Investigator tries to remain undetected
Disadvantages	● Information gathering only during network activity ● No control over the incoming information ● Results may only be available very late

Table 6: Sub-Strategy of sensitive investigation

The objective of sensitive investigation in SDN is to conduct passive investigations, focusing on comprehensive evidence preservation at the control interface. This approach involves not sending queries on the network, thereby avoiding negative network impact. Network traffic is stored for analysis of raw data, allowing live monitoring with normal information density in the network. Additionally, the investigator aims to remain undetected during the process. On the other hand, information gathering is limited to network activity, with no control over incoming information, and results may be available only after a delay.

Objectives	Active Investigation
Focus	Active and aggressive-fast investigations in SDN
Features	● Targeted requests are sent to the network ● Fastest possible targeted network reconnaissance ● Fast and high-quality information gathering ● Can have negative network impact ● Network traffic and query responses are saved ● High interface load ● Sensitive adjustment of network requests ● Much data is collected ● Time-intensive analysis of raw data
Disadvantages	● Risk of network overload ● Requests can trigger security mechanisms and defences ● High hardware requirements ● Special hardware and software required ● Teamwork of specialists ● Legally critical information can be obtained

Table 7: Sub-Strategy of active-aggressive investigation

The objective of active investigation in SDN is to conduct active and aggressive fast investigations, focusing on rapid and targeted network reconnaissance. This approach involves sending targeted requests to the network to achieve the fastest possible information gathering.

Even so, it may lead to a negative network impact, requiring the collection of network traffic and query responses, resulting in a high interface load. The process involves sensitive adjustment of network requests, leading to the collection of large amounts of data, followed by time-intensive analysis of raw data. Disadvantages include the risk of network overload, triggering of security mechanisms and defences, high hardware requirements, the need for special hardware and software, collaborative efforts among specialists, and the potential acquisition of legally critical information.

The strategies outlined can serve as effective means for clarification, rapid information acquisition, and robust evidence preservation within SDN environments. These strategies, in the context of this discussion, encompass overarching plans and methodologies aimed at achieving broad objectives and addressing associated challenges. They provide a structured framework for navigating the complexities of SDN and optimizing forensic processes.

On the other hand, within the realm of network forensics, there exist specific methods that offer targeted solutions to particular challenges or objectives. These methods are more concrete and application-oriented, offering actionable steps for achieving specific goals or resolving practical issues within the context of SDN investigations.

By combining strategic approaches with concrete forensic methods, practitioners can enhance their ability to effectively investigate incidents, gather critical information, and preserve evidence within SDNs. This comprehensive approach ensures thoroughness, efficiency, and reliability in the forensic analysis of SDN environments, ultimately leading to more informed decision-making and better outcomes in security investigations.

5.2. Methodological Forensics in SDN

Let us now try to transfer the classic forensic methods to the respective processes in SDN using the topics discussed in network forensics. In this context, methods are very specific procedures that aim to achieve a defined goal or solve a specific problem. It can be said that the methods are more practical approaches in network forensics [Ni18].

In any case, every single point of the forensic methods must be documented in detail, even if it is not explicitly described in each method mentioned below, to ensure the principle of chain of custody [Gs06].

The figures shown on the following pages have been selected in such a way that they invite you to examine the selected process. Adaptation to the specific needs of each forensic situation is always concrete and necessary in every individual case. The processes are constantly intertwined and can reinforce or weaken each other or even temporarily deactivate each other. However, it should be borne in mind that methodology, and in particular a standardised methodology, is an important component of successful forensics in dynamic and software-based networks.

In the context of SDN, traditional forensic methods must be adapted to overcome the unique challenges posed by the abstraction, scalability and dynamic nature of these networks. Standard forensics methods include data collection, preservation, analysis and presentation. In SDN these methods must be flexible and able to function in virtualized and often volatile environments.

Classic, comprehensive data collection, integrity assurance with signatures and checksums, as well as targeted analyses and correlations play an essential role here. The following section shows how these and other methods can be successfully used in advanced SDN to deal with typical forensic scenarios.

- ➢ Preparation

It is clear that a sophisticated preparation process is required for the effective handling of IT incidents. This process should be constantly adapted and improved, even when not in use. Routine interfaces that process the results of completed forensic processes are also necessary. In methodological terms, this involves the provision of necessary resources such as hardware and software, but also essentially the provision of trained personnel and elaborated action plans.

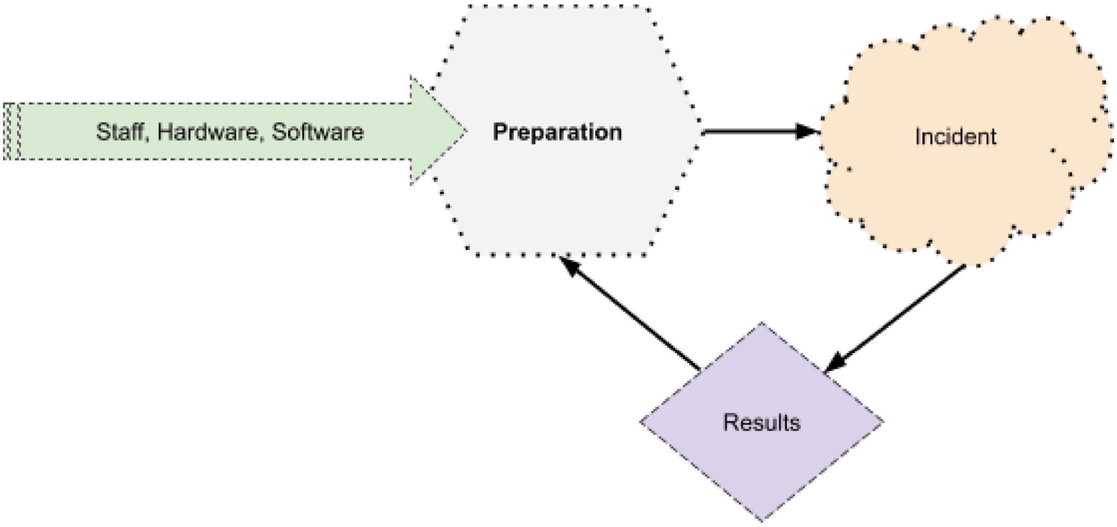


Fig. 43: Graphic of the methodological preparation process in network forensics

In the figure above, you can see that different influences affect preparation, such as staffing levels, hardware and software equipment. Basically, the available resources flow into the preparation for incidents. Furthermore, the results of the incidents are used to improve the preparation.

When preparing for incidents in SDN, special personnel, hardware and software must be available - this is already a difference to standard forensics in traditional networks.

➢ Clarification of the Preliminary Objective of the Investigation

One of the first questions that regularly arises is the question of responsibility and the legality of forensic investigations. In any case, this must be legally clarified and it must be possible to question and adapt these legalities in the further process. If the legal requirements are no longer met, this may necessarily lead to the immediate termination of the forensic activity. In other cases, powers can be extended and investigations or necessary measures can be intensified.

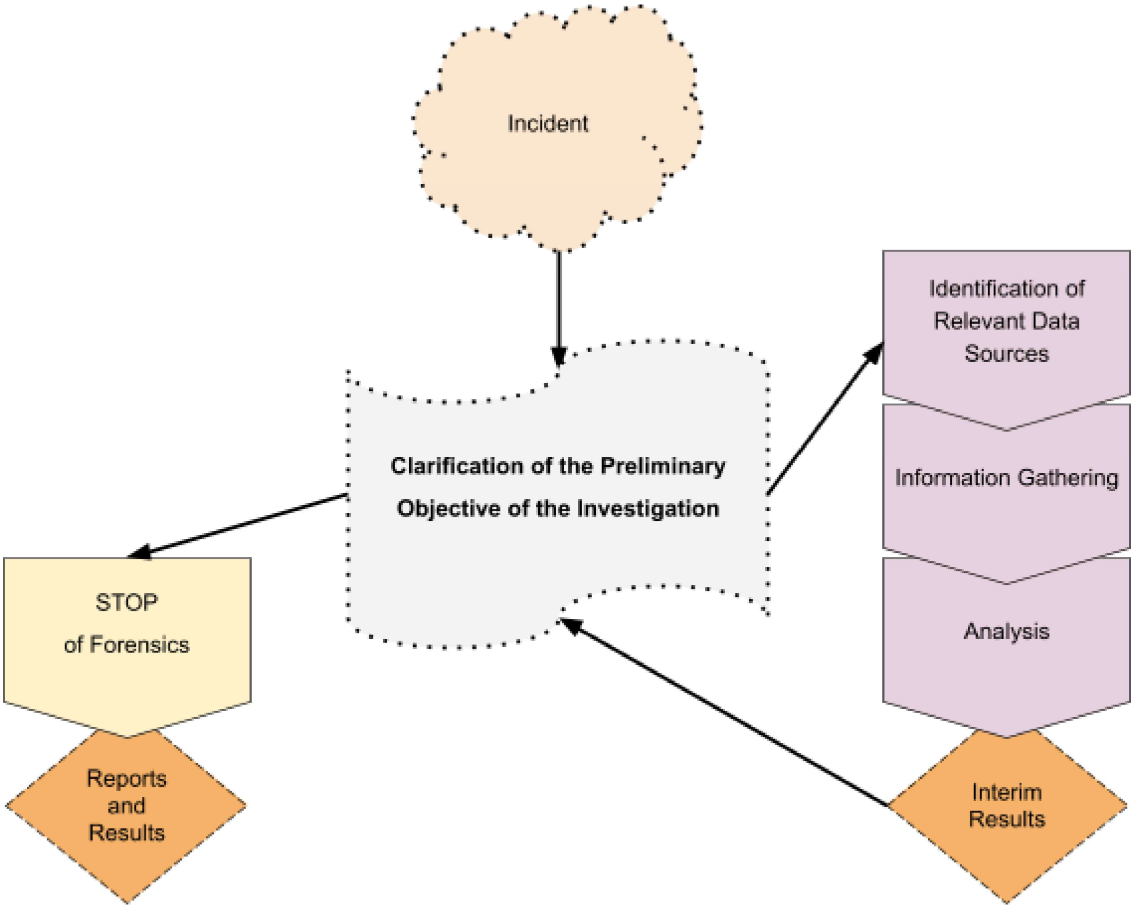

Incident
Clarification of the Preliminary Objective of the Investigation
Identification of Relevant Data Sources
Information Gathering
Analysis
Interim Results
STOP of Forensics
Reports and Results

Fig. 44: Graphic of the methodological legal consideration process

The process of reviewing legal circumstances described above is sometimes very complex and must be considered individually for each incident. In SDN, data situations become more complex and sometimes global. This requires comprehensive legal considerations that need to be reviewed from time to time in the incident.

➢ Identification of Relevant Data Sources

As a first technically practical method, it should be clarified and agreed which data is actually required to clarify the incident. This consideration sometimes has an enormous impact on the impairment of the network to be investigated and on the amount of data to be analysed and therefore also on the duration of the investigation. In particular, it must be clarified whether post-mortem forensics or live forensics should be carried out, or at which points, for example, the network traffic should be recorded. It can also be determined whether large amounts of data need to be saved live or whether backups can be used, depending on the nature of the incident.

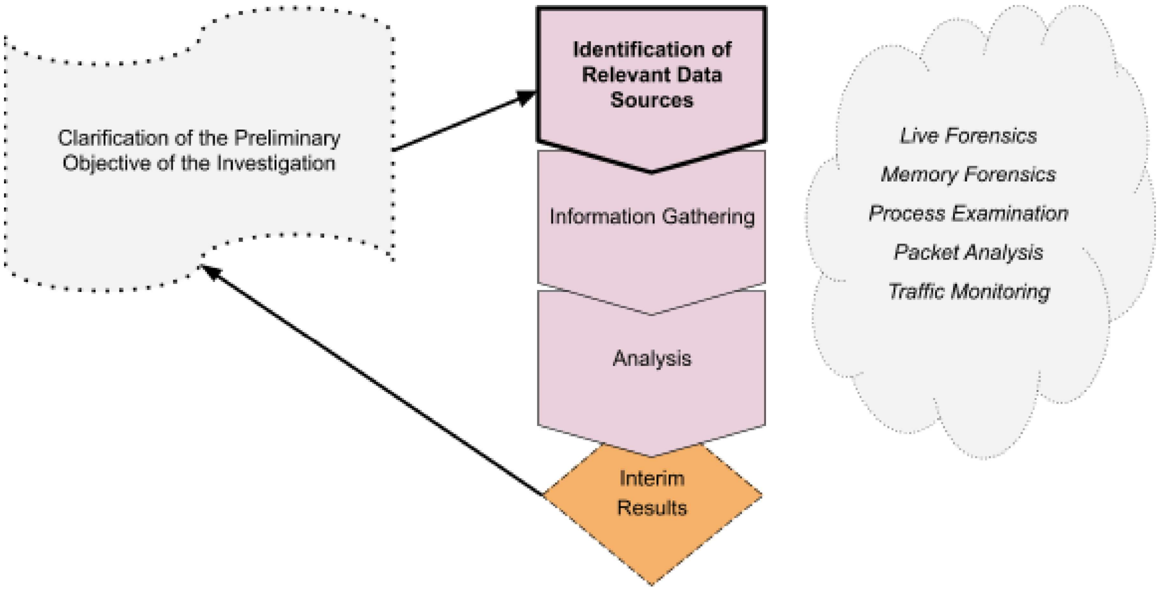


Fig. 45: Graphic of the data identification process

As shown, the identification processes of relevant data are a main aspect of the first activities of network forensics in SDN.

The consideration of the relevant data sources must be regularly reconsidered in SDN, also with regard to the legal competencies of the investigation.

➢ Information Gathering through Live Forensics

The presence of an SDN makes live forensics unavoidable. The presence and structure of an SDN can only be determined and secured through effective and meticulous data collection from a network participant. The classic procedures of live forensics are mandatory, such as creating a memory image and backing up the running processes. The experience of the forensics team involved plays a major role in the success of the forensic process and the preservation of evidence as a whole.

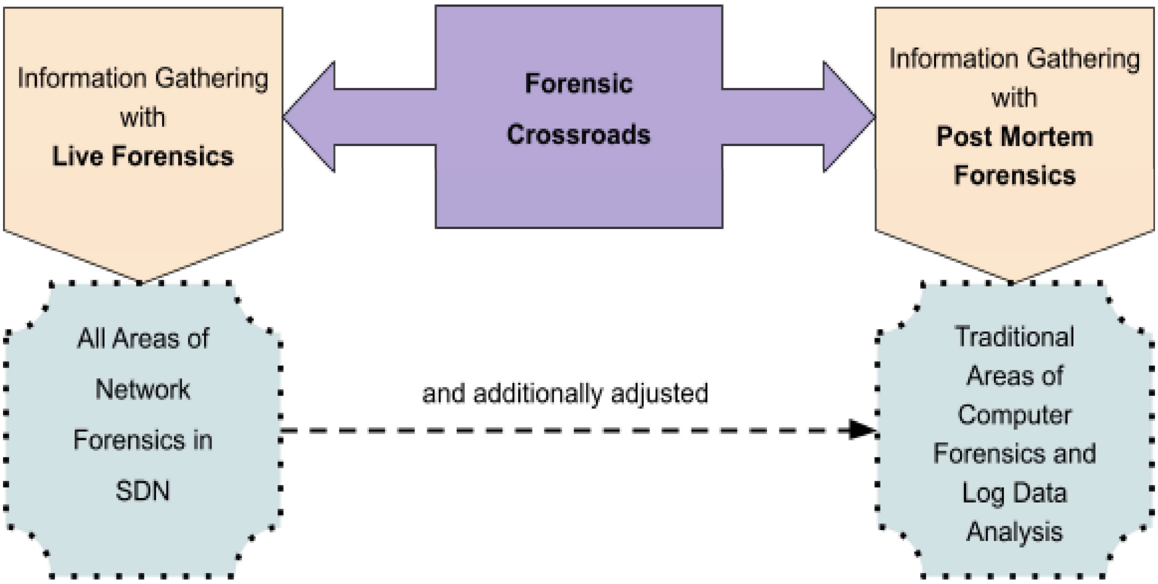

Information Gathering with Live Forensics
Forensic Crossroads
Information Gathering with Post Mortem Forensics
All Areas of Network Forensics in SDN
and additionally adjusted
Traditional Areas of Computer Forensics and Log Data Analysis

Fig. 46: Graphic of the process to live forensics

In simplified terms, it was shown here that there are also classic considerations in the investigation of SDN as to whether one should (continue to) use live forensics or post-mortem forensics. In many cases, at some point the network will be shut down and post-mortem forensics will be performed. Nevertheless, the information gained from live forensics is extremely important, especially when investigating SDN.

- ➢ Sensitive Data Collection

As SDN is a dynamic, software-controlled application, stressing these processes can severely impair their function. Therefore, passive scans with a low load should initially be selected for the investigation of the network functions.

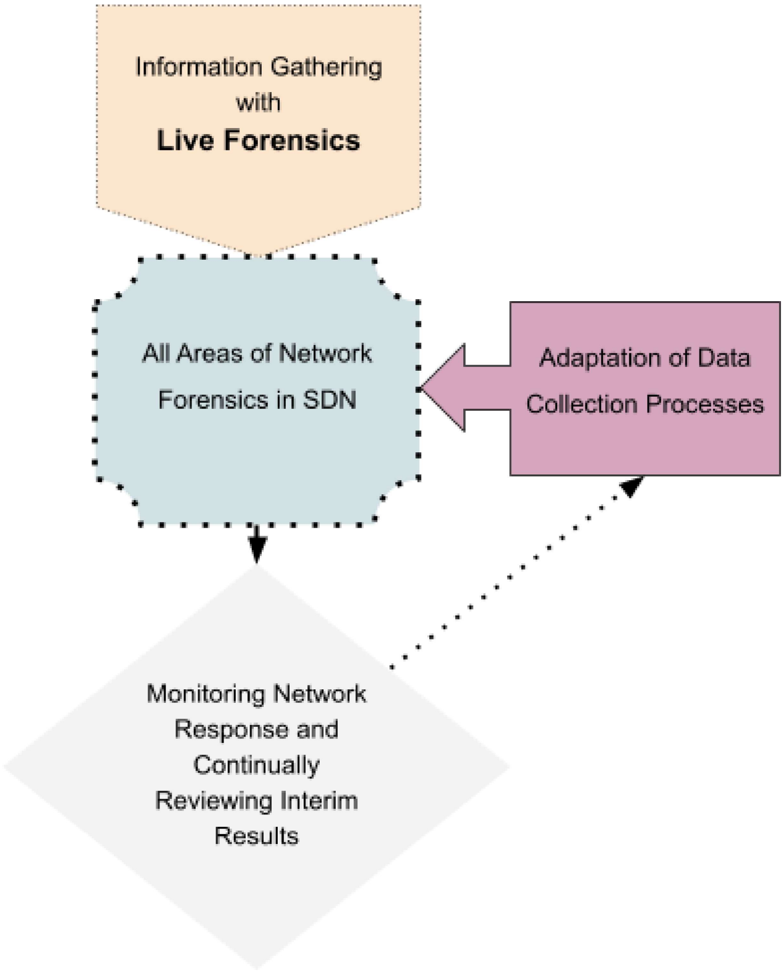


Fig. 47: Graphic of the methodologies of sensitive data collection

The graphic shows how forensic investigations can be sensitively adjusted in SDN. Simple and non-intrusive examination methods are chosen first. Then, if necessary, more active and aggressive investigation methods are used, whereby SDNs react very sensitively to classic network incidents and can partially deactivate or collapse.

➢ Verification of Data Consistency

Data must already be checked for consistency during the collection process. The data collection must therefore be viewed and checked live from time to time, as otherwise valuable evidence may not be collected at all, or may be collected partially or incorrectly. Routine processing using secure hashing procedures and formation of checksums is a key building block for verifiable data consistency.

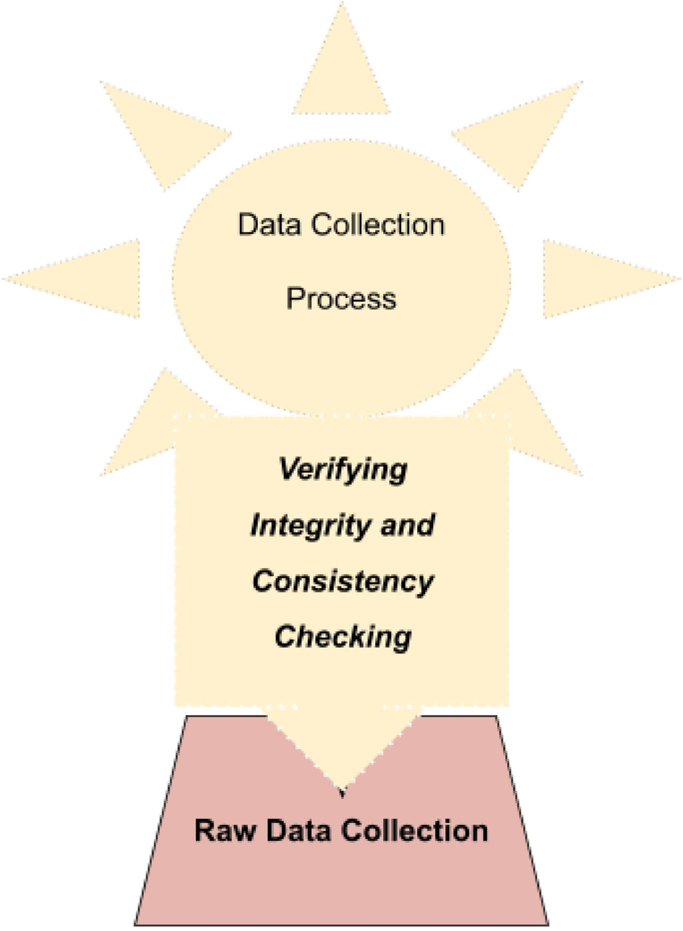


Fig. 48: Graphic of the verification process

Data verification is a rigorous process in forensic investigations. Since the data paths in SDN are often individual and data must be routed in a complicated manner, a secure process must be implemented to verify and check data consistency.

- ➢ Pre-Evaluations and Live Monitoring

The data must also be checked promptly with regard to its logical plausibility. This includes, for example, plausible time stamps, system information, etc. In this early phase, the aim is to identify and document possible sources of error or data conflicts.

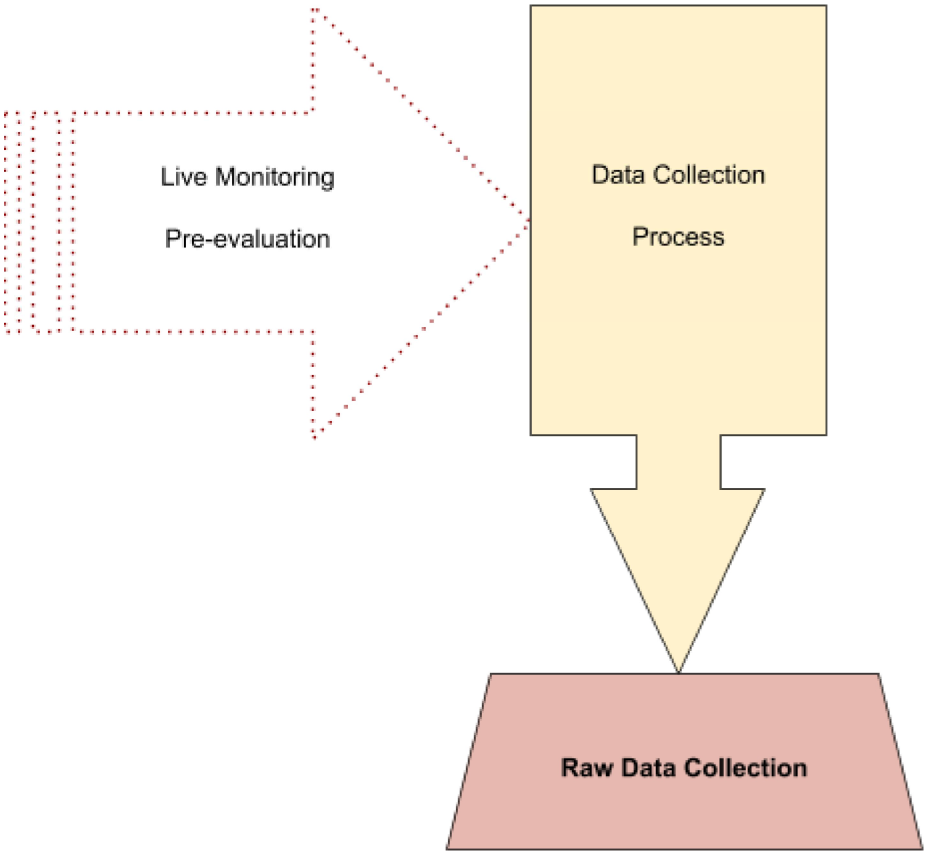


Fig. 49: Graphic of the methodology of live monitoring and pre-evaluation

Constant logical checks of the data collection must also be carried out in order to detect errors at an early stage. With live monitoring and pre-validation, the data streams must be checked and logged in order to ultimately create a valid data collection in the SDN environment.

➢ Adjustment of Forensic Investigation Objectives

As soon as the first results or tendencies in the investigation process are available, the legal requirements must also be checked at short notice. It must be determined whether the legal options have changed, i.e. whether proportionality is maintained, or whether additional goals or investigative tasks have arisen.

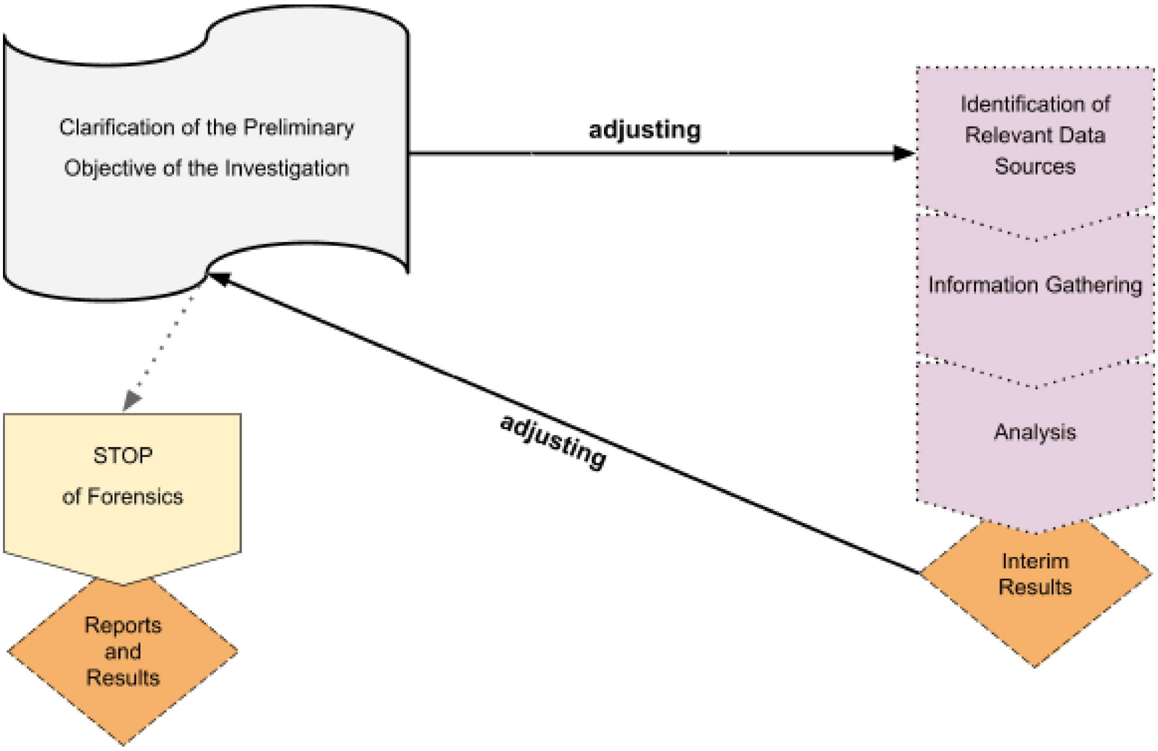

Clarification of the Preliminary Objective of the Investigation
adjusting
Identification of Relevant Data Sources
Information Gathering
Analysis
Interim Results
adjusting
STOP of Forensics
Reports and Results

Fig. 50: Graphic of the adjusting process of investigation objective

The objectives and legal framework of forensic investigations must be constantly adapted in order to adjust the information gathering process. To this end, interim results are formed, which are then reviewed in a targeted manner.

- ➢ Adjusting Data Collection Methods

As the first reliable results become available, the data collection methods can be adjusted. This means you can switch from passive to more active scans in an individually tailored manner. Additional staff, programs or hardware may also necessarily be required.

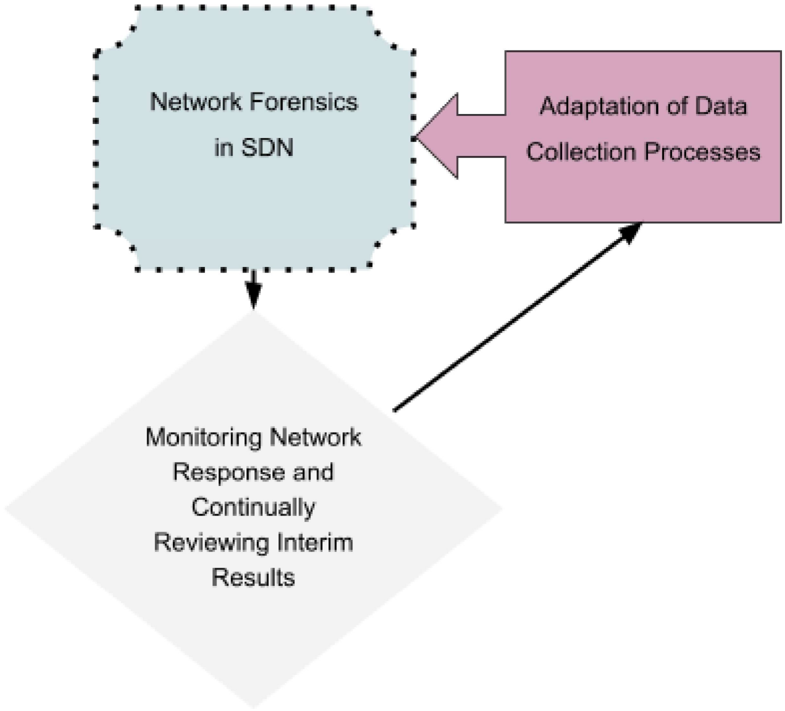

Network Forensics in SDN
Adaptation of Data Collection Processes
Monitoring Network Response and Continually Reviewing Interim Results

Fig. 51: Graphic of the data collection process

When adapting the methods in the data collection process, the quantity and quality of the data to be collected can be modified. Particular attention must be paid to the utilisation of the network, the plausibility and the present data quality. Monitoring processes are essential for this, which usually have to be monitored live by an expert in order to be able to intervene quickly if necessary.

- ➢ Temporal Assessment of the Data Collection Process

An assessment of the data collection process should be carried out regularly. Factors such as proportionality, network load, personnel effort and achieving the investigation objectives play a role here. Experience shows that data collection should only, if possible, be stopped after the results of the investigation have been verified.

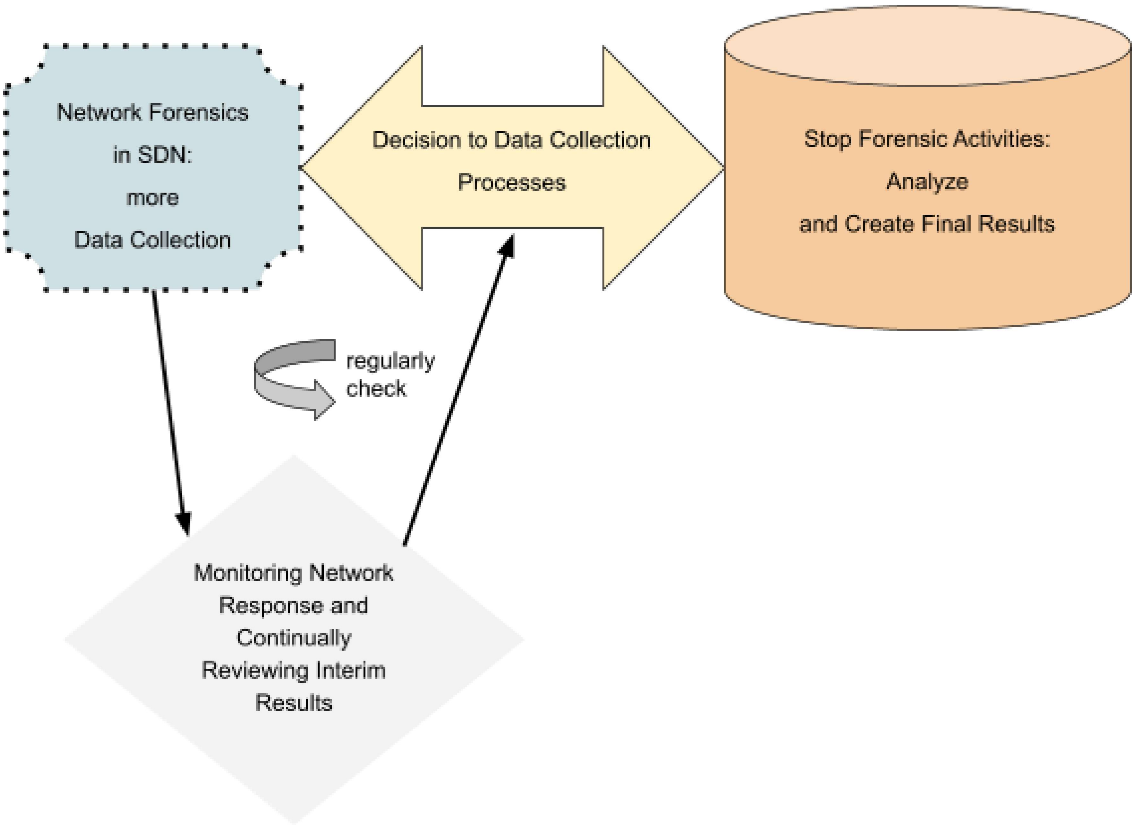


Fig. 52: Graphic of assessment of data collection process

The figure shows the process of regularly reviewing the data collection process. The decision is made each time whether the data collection process should be continued or stopped.

- ➢ Review of Initial Results of Information Gathering

Forensic experts must examine the initial results in detail to determine their reliability, both technically and logically. The further information collection process is often based on these initial results and assessments by the experts.

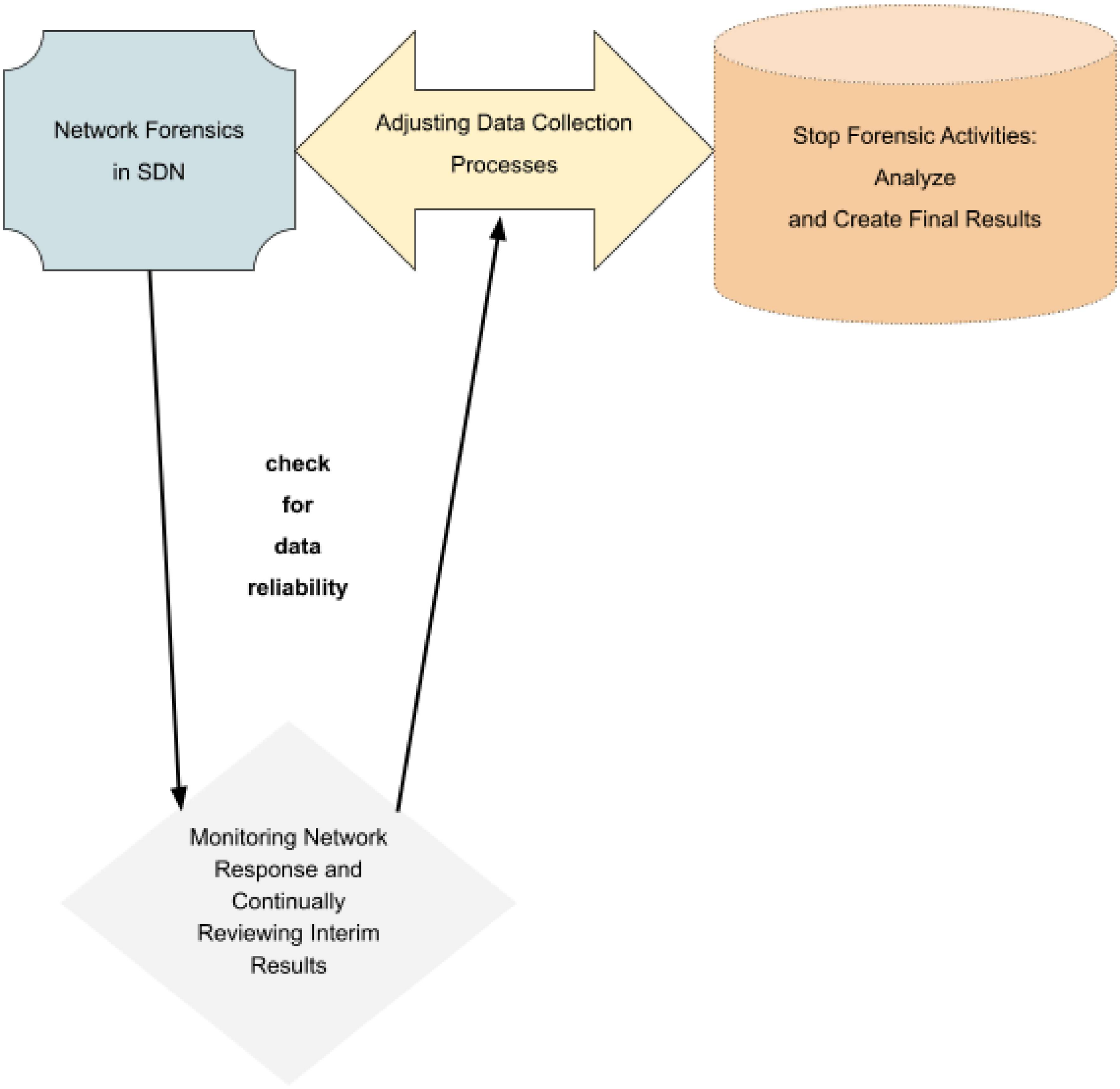

Network Forensics in SDN
Adjusting Data Collection Processes
Stop Forensic Activities: Analyze and Create Final Results
check for data reliability
Monitoring Network Response and Continually Reviewing Interim Results

Fig. 53: Graphic of data reliability check

The figure above shows the checking of the collected data for logical reliability. A reaction must also be made here in the data collection process in order to correct or recognize any data errors.

➢ Analysis of the First Data Collection

The initial results of information gathering are often essential for decision makers who are not necessarily forensic experts. This requires close consultation and timely processing of the information by IT experts. Based on this initial information, a decision should be made whether and how the forensic process should be continued.

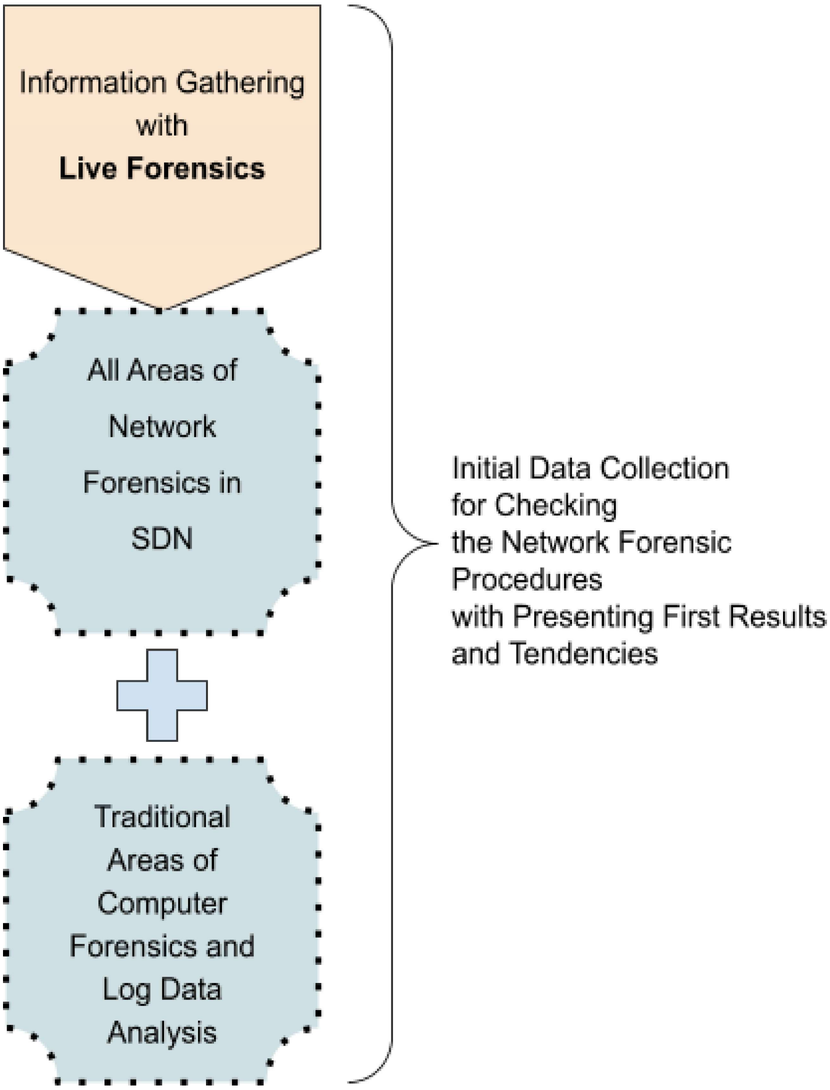

Information Gathering with Live Forensics
All Areas of Network Forensics in SDN
Traditional Areas of Computer Forensics and Log Data Analysis
Initial Data Collection for Checking the Network Forensic Procedures with Presenting First Results and Tendencies

Fig. 54: Graphic of first data collection for checking forensic procedures

This graphic shows which data should be compiled to produce an initial result. This includes the information from live forensics, the results of network forensics and the first partial results of traditional computer forensics.

- ➢ Continuation or Termination of the Data Collection Process

If information is available or absent in the medium term, the data collection process must be questioned. Should data collection be continued, intensified, or stopped? The assessment of the forensic scientists in the team as well as the considerations of superiors and those responsible play an equal role in these decisions.

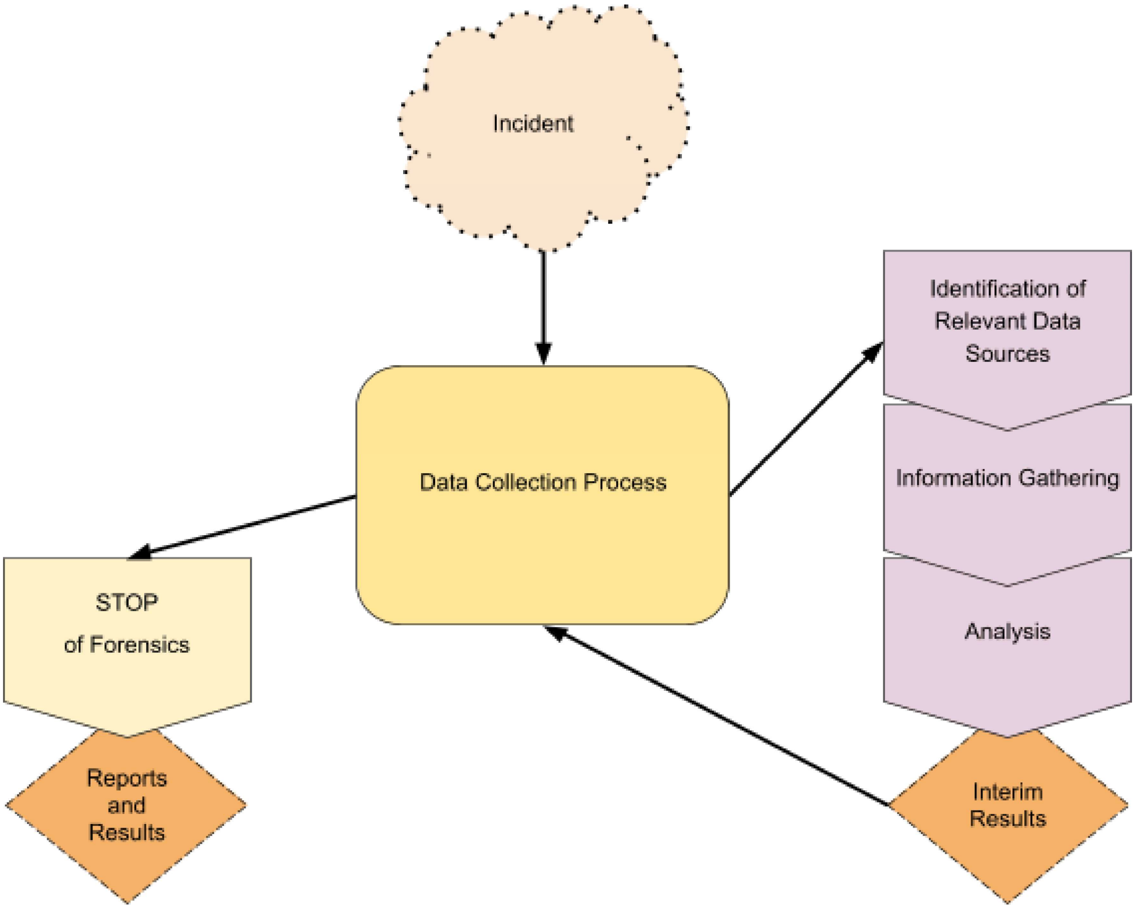


Fig. 55: Graphic of the data collection process overview

This diagram shows the central process of data collection. If no useful data is available, a weighing up must take place, relevant data sources must be checked or the forensic data collection process must be terminated.

- ➢ Preparation and Filtering of the Collected Data

Since large amounts of data are regularly collected and available for evaluation, this data must be filtered and processed. This cannot always be done automatically, but often requires manual work by forensic specialists. It should be noted that, of course, forensic copies of the original data must be used so that original raw data can be used in the event of an error.

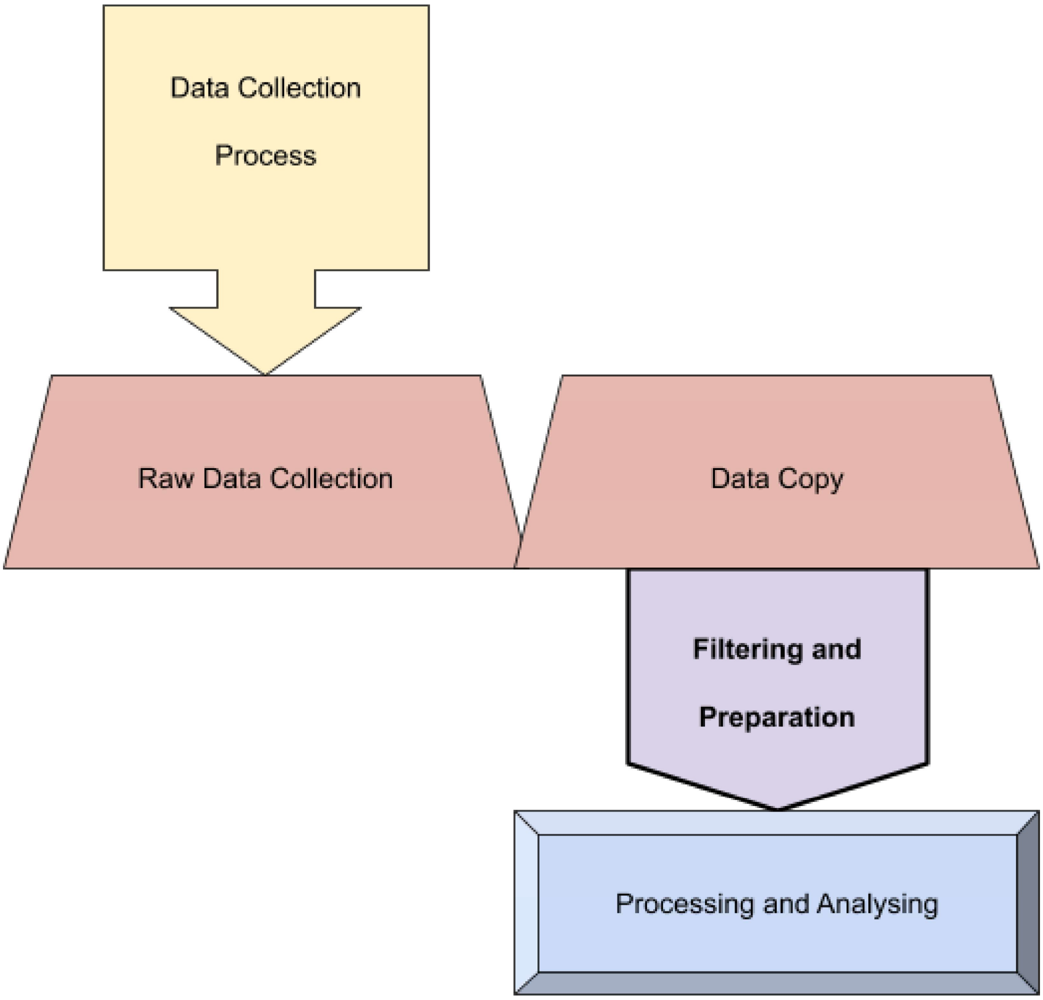


Fig .56: Preparation and filtering of the collected data

The diagram shows that the data must be filtered and processed promptly, especially in large and complex SDNs. This cannot always be done directly during data collection, so the collected raw data is processed and filtered afterwards.

- Checking Data Integrity and Conducting a Plausibility Check

Once the final data collection is available, data integrity and plausibility must also be checked and documented. As already mentioned above, collected network data must also be forensically copied and finally secured using hash values. Checks must also be carried out when creating duplicate data and after processing steps.

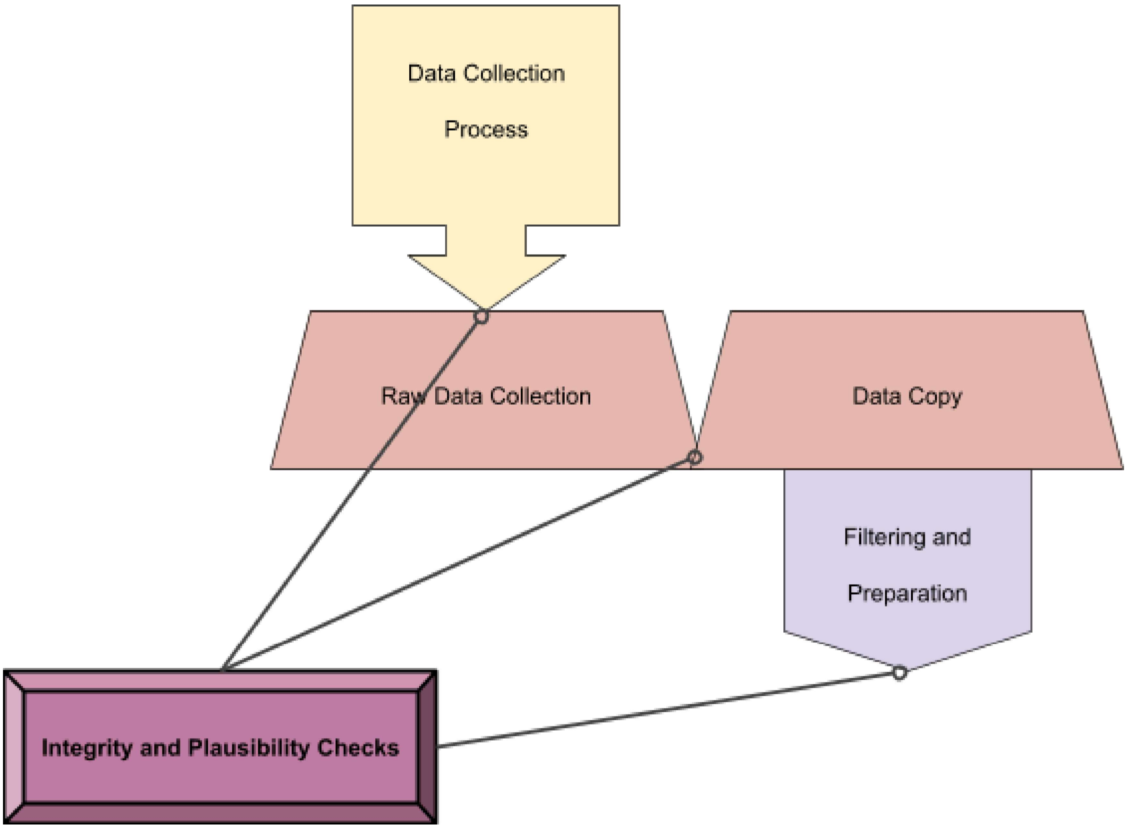


Fig. 57: Checking data integrity and plausibility

Integrity and plausibility checks should take place at different points in the investigation processes in SDN. For example, the checks should take place after data collection, between data copying processes and after filtering or preparation.

➢ Delimitation of Initial Analysis Results

An important point is the demarcation of the analysis results, and thus the separation of what is important and what is unimportant. It is important to know exactly the objectives of the investigation and the facts of the case. Typical legal issues should also be addressed, which can then be delineated within the framework of the data available.

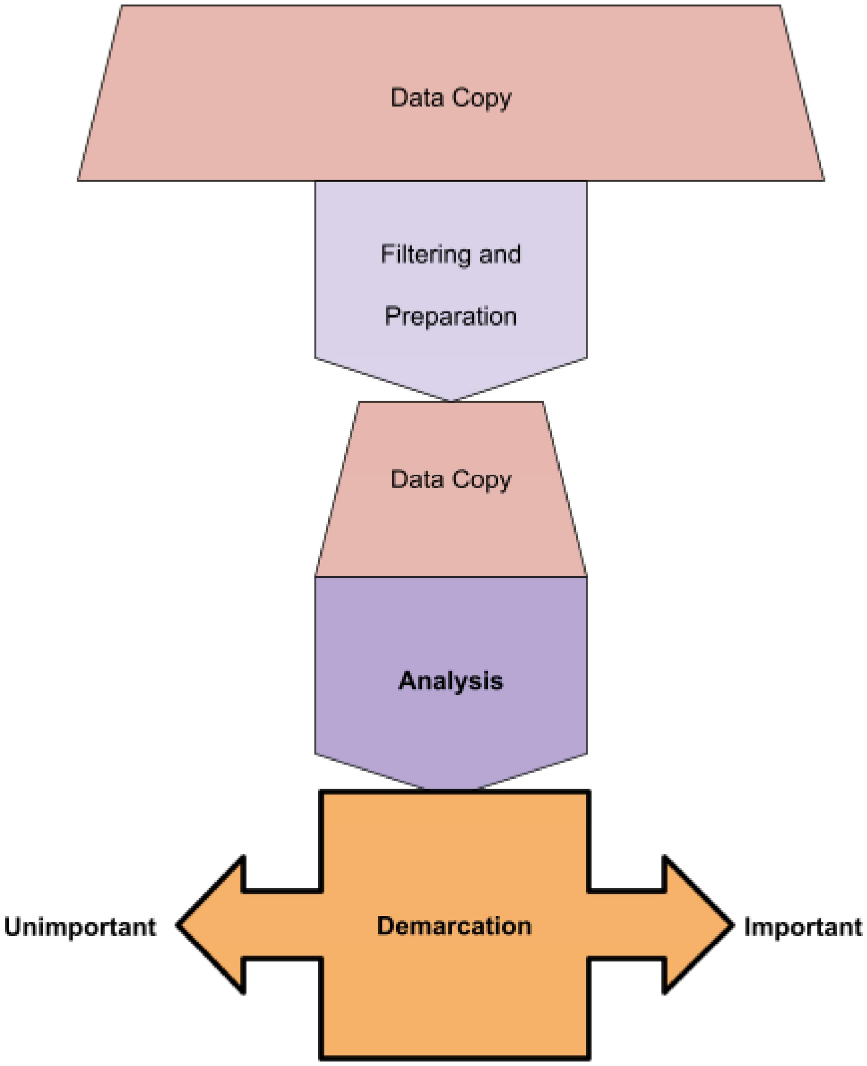


Fig. 58: Delimitation of initial analysis results

It can be seen here that after processing the data, information is generated that is important or unimportant for the investigation process. It is sometimes difficult to distinguish between the two, and no information should be completely discarded.

- ➢ Preparation of Appraisals or Reports

The preparation of official reports is a main task and usually the conclusion of the forensic process. The reporting must be based on the technical specifics of SDN. Necessary technical descriptions and details must be included and, depending on the case, described in detail. A report is created automatically under the control of a forensic scientist or manually by an expert [Bu06].

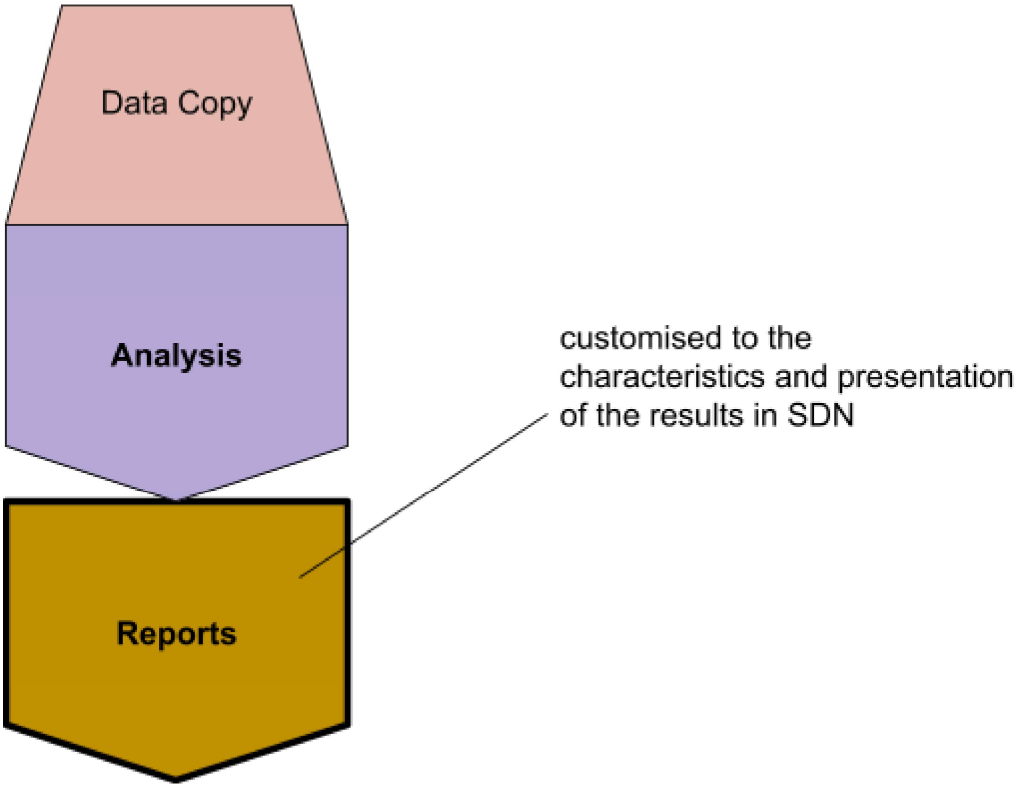

Data Copy
Analysis
Reports
customised to the characteristics and presentation of the results in SDN

Fig. 59: Preparation of reports

A standardised report is created after analysing the data with all relevant information. Legal circumstances, problems and results must be addressed in detail. Problems often arise in the automated analysis of SDN because analysis programs do not fully understand the conditions of the SDN network structures.

- ➢ Graphical Processing of the Results

Automated, graphical network overviews and technical diagrams can simplify the explanations of technical reports [Bu06].

Graphical comparisons are usually clearer to non-experts than simply presenting numbers. Drawings and illustrations can also shorten complex written elaborations and support decision-makers in the understanding process.

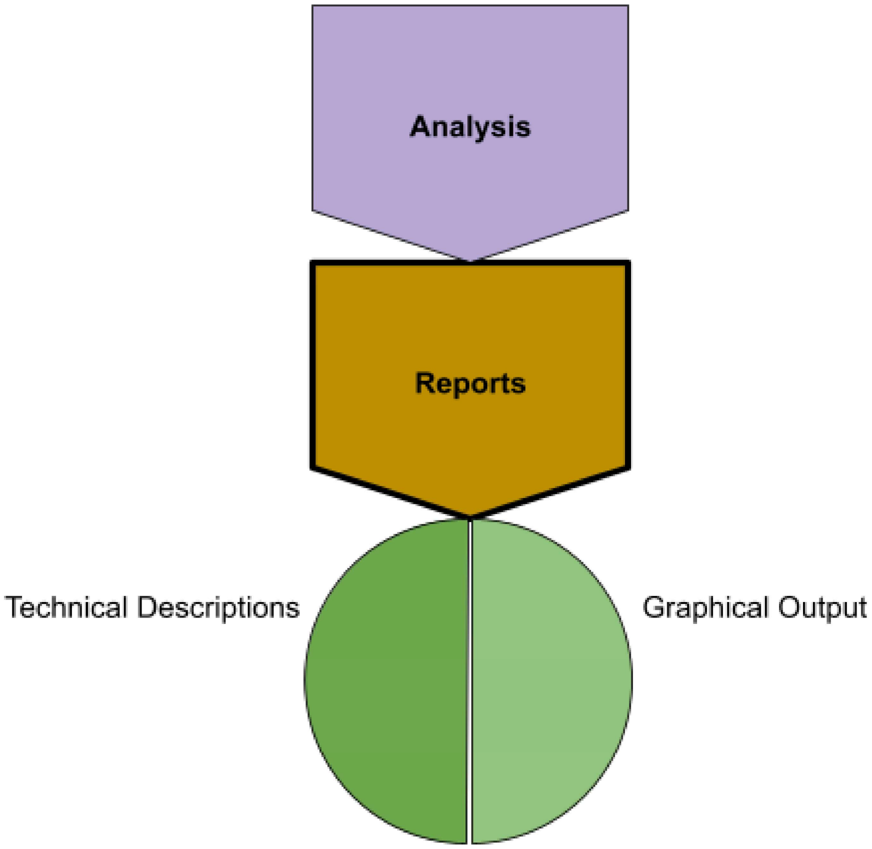


Fig. 60: Graphical Processing of the results

Reports can still be graphically enhanced. In SDN, for example, a graphical representation of the network structures is often very helpful in order to be able to clearly understand investigation steps and technical information later on.

- ➢ Developing Overall Results to Improve Forensic Workflow

Finally, the further development of forensic methods is necessary when investigating processes on dynamic and advanced SDN.

The results as well as technical and legal problems must be incorporated into the preparation process and be available at the beginning of the next incident. In this way, other experts can also learn from cases that have already been processed and continually improve the forensic process.

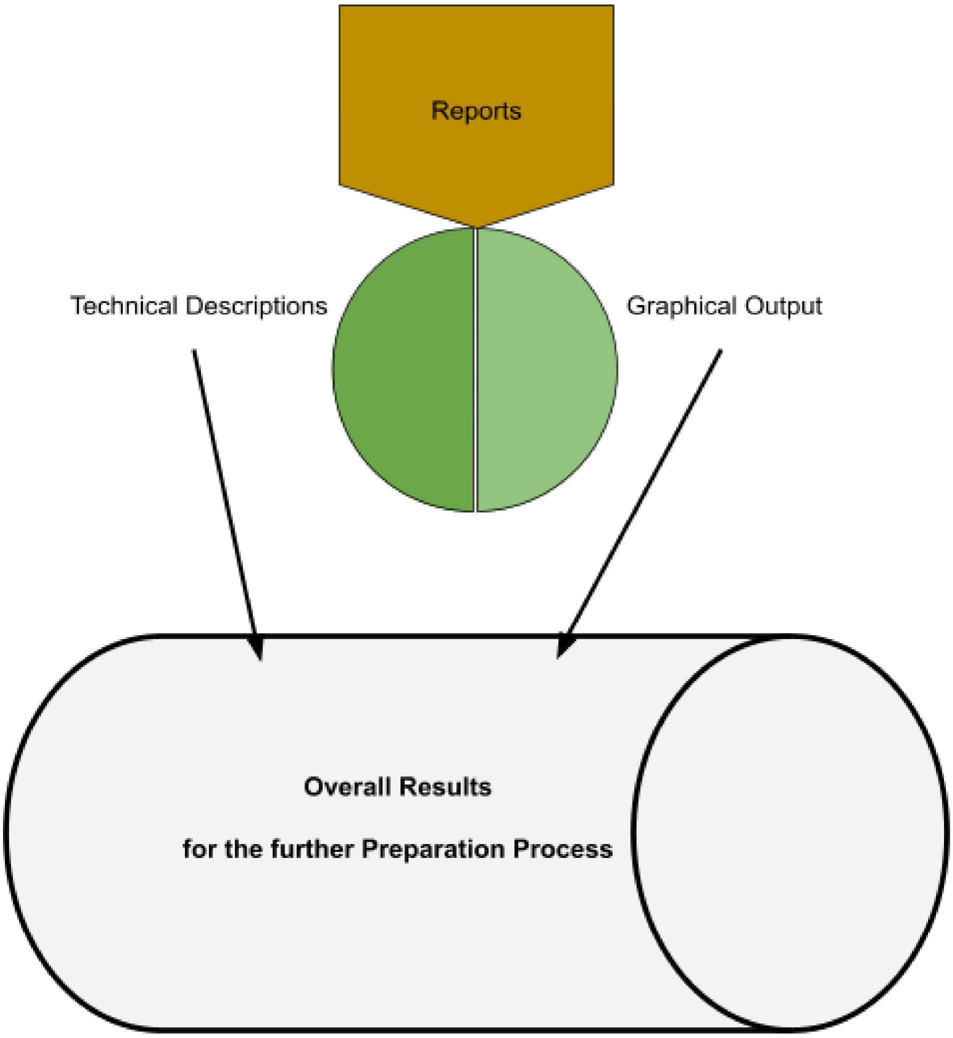


Fig. 61: Developing overall results for the further preparation process

Final results must be constantly incorporated into a broad preparation process in order to improve the quality of the procedures and to be able to adapt to the dynamics in SDN.

5.3. Discussion of the Technical Challenges

Network forensics in SDN faces a variety of technical challenges that must be taken into account. These challenges encompass the complexity of the network architecture, lack of transparency, dynamic nature, scalability, and the sheer volume and complexity of data within SDNs.

SDN networks are often complex and dynamic, making it difficult to identify and track incidents or attacks. The virtualization of resources and the dynamic adjustment of network flow further complicate the ability to maintain an accurate and comprehensive view of the network's state. Unlike traditional networks, where physical infrastructure can provide a tangible reference point for forensic analysis, SDNs abstract the physical layer, creating an additional layer of complexity.

One of the primary challenges is the lack of transparency. The abstraction from the physical infrastructure in SDNs complicates the monitoring and analysis of network traffic. Understanding data flow and communication patterns becomes particularly challenging in virtualized environments, where network configurations can change rapidly. This dynamism means that forensic investigators must be adept at understanding and interpreting virtual network topologies and configurations.

To effectively detect and respond to threats, forensic analysis in SDNs must often be performed in real time. This requires powerful monitoring tools capable of capturing and analysing data on-the-fly, as well as rapid response mechanisms to mitigate identified threats. However, the necessity for real-time analysis introduces additional challenges related to processing power and data handling capabilities. High-performance computing resources and sophisticated algorithms are essential to manage the data volumes and ensure timely detection of anomalies.

SDN networks can generate large amounts of data that must be analysed promptly to identify suspicious activities. The complexity of this data, often involving intricate and numerous connections and transactions, requires advanced analysis techniques and algorithms. Traditional forensic tools and methods may not be sufficient, necessitating the development and deployment of specialised forensic software tailored to the needs of SDNs.

Virtualization, a core aspect of SDNs, poses another significant technical challenge. The dynamic nature of virtual environments means that forensic investigators may need to reconstruct the state of the network as it was at the time of an incident, which can be exceedingly difficult. Virtual machines and virtual networks can be created, modified, or destroyed rapidly, often leaving little trace. IP or MAC addresses may be reassigned or altered, further complicating the ability to trace network activity back to its source.

This makes live forensics - conducting forensic investigations while the network is still running - particularly important in SDNs. Live forensics can provide crucial insights and evidence that may be lost once the network state changes or devices are powered down. Therefore, capturing data in real time and making accurate, immediate interpretations of that data is critical.

All in all, the technical challenges in network forensics within SDNs are multifaceted and significant. They require a combination of advanced technical skills, specialised tools, and real-time analysis capabilities. Forensic investigators must navigate the complexities of virtualized environments, manage vast and complex data sets, and adapt to the dynamic nature of SDNs to effectively identify and respond to security incidents. The role of live forensics becomes particularly vital, underscoring the importance of timely and accurate data capture and analysis.

5.4. Discussion of the Legal Situation

In addition to the technical challenges, there are significant legal frameworks that must be considered when conducting forensic investigations in SDN. These frameworks encompass various aspects such as privacy and data security, provability and integrity, liability and accountability, and other legal requirements.

Network forensics in SDN must adhere to applicable data protection laws, regulations, and policies, including the General Data Protection Regulation (GDPR) in Europe. This means that the collection and analysis of network data must always be traceable and transparent, and must respect the intellectual property rights of all network users. Investigators need to ensure that any collected data is relevant to the investigation and that the rights of individuals are not violated.

One of the primary legal challenges is the handling of personal and sensitive data. Forensic work often involves access to personal information or even sensitive medical data, which is particularly protected by law. Therefore, special technical measures such as encryption, checksums and hash protection must be taken to protect such data, and comprehensive legal authorisations or permissions should be obtained before data collection begins.

Forensic evidence collected in SDN must be usable for legal purposes, which requires maintaining the integrity of the data throughout the investigation process. This involves documenting the investigation methods thoroughly, verifying results meticulously, and cooperating closely with authorities. The chain of custody must be maintained to ensure that the evidence is admissible in court. This includes timestamps, hash values for data integrity, and detailed logs of all actions performed during the investigation.

In forensic investigations within SDN, liability and accountability are crucial aspects. Investigators must ensure that their actions do not unintentionally violate laws or policies, which could lead to legal repercussions. There must be clear accountability for all actions taken during the investigation, and roles and responsibilities should be well-defined. This is particularly important when dealing with cross-border investigations, where different jurisdictions may have varying legal requirements.

Current data protection regulations, such as the GDPR, impose strict requirements on data collection, processing, and storage. The GDPR strengthens the rights of digital participants in computer networks, which includes the right to privacy and data protection. This regulation mandates principles such as legality, good faith, transparency, purpose limitation, data minimization, accuracy, storage limitation, integrity, and confidentiality. These principles must be observed during forensic investigations to ensure compliance.

SDN investigations face additional legal challenges than traditional digital forensics due to the specific nature of data collection and analysis. Since data collection must be targeted and specific to the incident or defendant, there may be legal barriers to conducting an all-encompassing network analysis. Legal frameworks require that any forensic investigation respects the rights of individuals and complies with national and international laws.

The dynamic and virtualized nature of SDN also complicates the task of maintaining data integrity and privacy. For instance, IP and MAC addresses might change, and the virtualized environments may alter data flows, making it harder to ensure compliance with data protection laws. For example, due to the increased dynamics in SDN, innocent participants could quickly become involved in investigations or be suspected as criminals.

A significant legal challenge in forensic investigations is balancing the need for data collection with the protection of personal data. While it is essential to prevent, detect, investigate, and prosecute criminal offences, this must be done without infringing on individuals' rights. This balance requires a careful and nuanced approach, often necessitating the involvement of legal experts to guide the forensic process and ensure compliance.

Forensic experts and legal authorities must collaborate closely to navigate these challenges. Legal decisions should ideally be made by qualified lawyers who can defend the actions taken during the forensic process in court.

However, forensic experts often find themselves in the position of having to justify their actions legally, which can jeopardise the entire investigation if not done correctly.

In conclusion, the legal situation in network forensics within SDNs is complex and multifaceted. It requires a thorough understanding of data protection laws, careful documentation and maintenance of data integrity, and close collaboration between forensic experts and legal authorities. Legal clarity in this area is often lacking, leading to case-by-case decisions by courts. As such, forensic investigations must be meticulously planned and executed to ensure they meet all legal requirements and withstand legal scrutiny.

Current developments in the field of digital forensics, and in particular forensics in networks, seem to demand that forensic scientists deal with legal problems in a well-founded manner. Serious errors or wilful non-compliance with legal regulations can be blamed on the technical expert in individual cases. This can result in claims for damages, criminal prosecution or disciplinary sanctions. An appeal here could be that comprehensible regulations and uniform standards must be established and implemented.

6. Summary

In the following, the methods are presented graphically. The results obtained are abstracted and converted into a graphical model.

6.1. Graphical Presentation of Results

Now it is clear from the presentation of the strategies and methods of classic digital forensics as well as the strategies and methodology of network forensics in SDN that a classic model or a classic approach can no longer be derived. Therefore, in this case, a multidimensional model is used for the representation, which can accommodate different influences of the methods and strategies.

Ultimately, the methodology and strategy bring together technical approaches, legal requirements, methodology and strategic considerations:

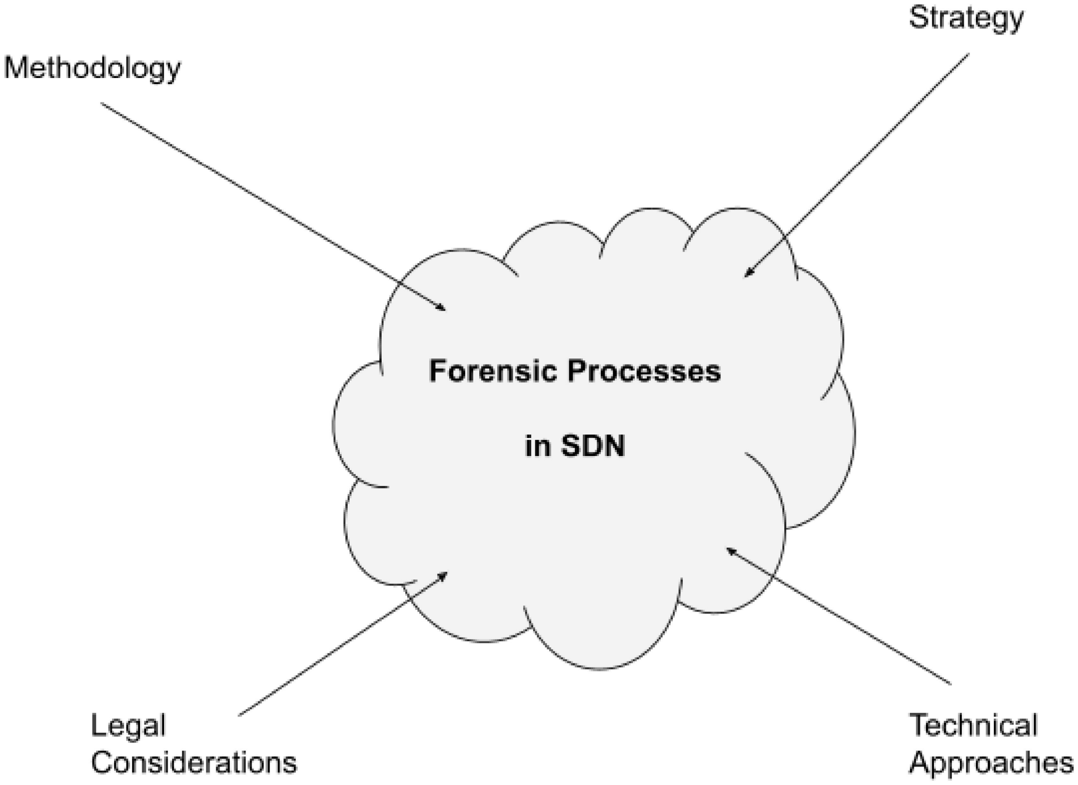


Fig. 62: Multidimensional precursor model of forensics in SDN

This matrix with n=4 can be represented mathematically as follows:

$$\begin{pmatrix} \text{Methodology} \\ \text{Strategy} \\ \text{Legal Considerations} \\ \text{Technical Approaches} \end{pmatrix}$$

Fig. 63: Abstracted quantitative formula of main forensic processes

This model then contains a variety of possible dimensions in the form of applicable strategies, methods, technical applicability and legal considerations and therefore quickly becomes large and complex.

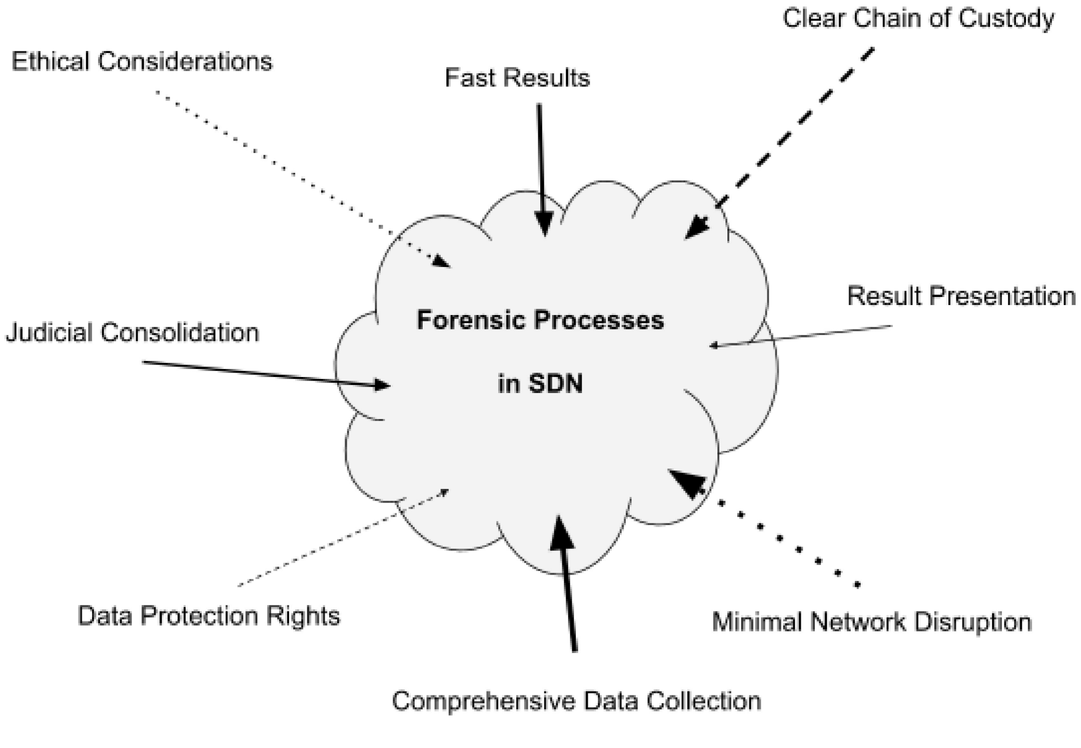


Fig. 64: Complex multidimensional model example of forensics in SDN

In the above illustration, any factors influencing the forensic process in SDN were selected and weighted in a complex multidimensional model, with various arrows representing the different weightings.

This connection ◦ of the weighted matrix with n=7 and the individual weighting factor w can be represented as follows:

$$\begin{pmatrix} w0 \circ \text{Ethical Considerations} \\ w1 \circ \text{Fast Results} \\ w2 \circ \text{Clear Chain of Custody} \\ w3 \circ \text{Result Presentation} \\ w4 \circ \text{Minimal Network Disruption} \\ w5 \circ \text{Comprehensive Data Collection} \\ w6 \circ \text{Data Protection Rights} \\ w7 \circ \text{Judicial Consolidation} \end{pmatrix}$$

Fig. 65: Abstracted quantitative formula of forensic processes

In abstract terms, the multidimensional model could be described simplified mathematically as follows:

$$w_n \circ N_n$$

N is the set of properties that play a role in the forensic process and n is their identification counter, or n+1 is the number of properties N.

We see that in the simplest model mentioned above, n=3. With few properties to consider, n=7. The complexity of the procedures regularly increases with the number of n+1 dimensions.

Analogously, other strategies and methods can also be applied to the complex multidimensional model:

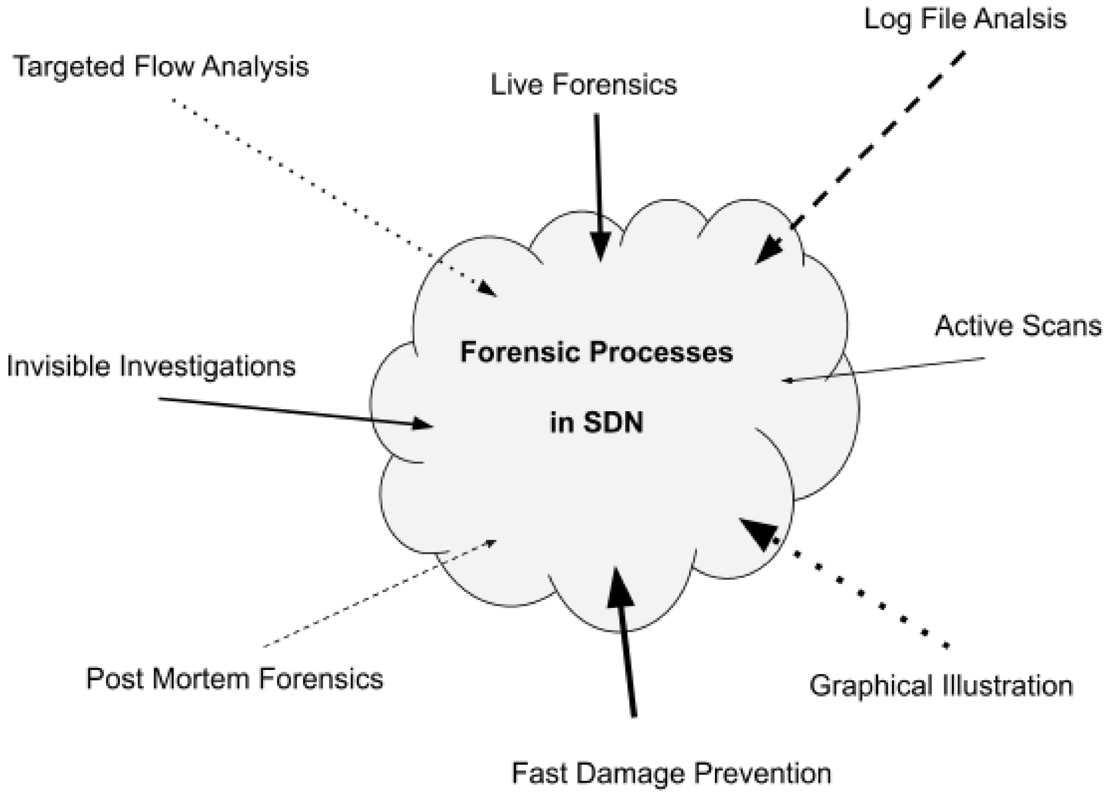


Fig. 66: Complex Multidimensional model example of forensics in SDN

This simple mathematical model shows the strategies and methods for forensics in SDN. The complexity increases as more of the methods and strategies already discussed are incorporated into the forensic process. In this way, forensic events and processes can also be assessed and evaluated retrospectively in order to adapt and improve the investigation methods and strategies.

6.2. Development of Network Forensics in SDN

So it can be seen from the discussed illustrations, strategies and methodologies, the classic forensic workflow [Ke06] with collection, examination, analysis and reporting is being put to the test, especially in the context of investigations in SDN.

Obviously, these points usually have to be dealt with, but there are other things that need to be taken into account depending on the application.

Computer network forensics is no longer about examining dead digital devices. Large computer networks are very dynamic structures that can hardly be depicted in detail. This problem becomes quickly apparent in virtual or software-defined environments.

Even when examining an SDN, intermediate steps that are not only of a technical nature must be taken into account. This makes a forensic investigation of SDN structures extremely challenging.

Forensics of SDN environments is a complex and rapidly evolving field. The development of new methods [Ni18], tools and standards as well as the training of forensic investigators in SDN technologies are necessary to effectively address the challenges of investigating these dynamic networks.

In the following, some points of forensics in SDN will be presented from a practical perspective in order to create a compact and useful workflow.

During the preparation phase, regular training of forensic specialists is essential, as current technologies are often used in SDN. In addition, at least access to technically reliable solutions for data extraction and storage is necessary. During the preparation phase, scenarios should also be practised regularly or at least discussed with other experts.

Obtaining information and ongoing documentation is another extremely important component of a forensic investigation. Especially if network monitoring takes a long time or involves large amounts of data or clients, it must be meticulously documented right from the start. Legal requirements require consistent documentation not only in the analysis phase, but also at the first contact with the network.

As soon as initial information is available and legal requirements have been clarified, initial data collection can begin. Here it must be defined which goals are achievable and important. If comprehensive data backups are necessary, a high-performance and integrity solution for data extraction must be created. Techniques for integrity assurance and data consistency checking must be used in every case.

An information analysis can and should be carried out after the first reliable information has been obtained and must be included in the further evaluation process of the forensic actions. There should be a constant comparison of the requirements with the legal circumstances, as well as an assessment of the proportionality of the data collection.

Once data collection is complete, there is often no way back from a technical and legal perspective. If SDN structures are impaired or incriminated, a shutdown is regularly considered and evidence can no longer be preserved.

Nevertheless, the data can already be analysed as soon as the first network data blocks or evidence data blocks are available. This not only serves to collect evidence in a timely manner, but is also intended to ensure logical data consistency.

Taking into account the stated objectives of the investigation, the previous results should be regularly presented to decision-makers or other experts (four eyes principle). There may be a need for legal action or other circumstances that need to be taken into account.

After data collection is complete and all active actions in SDN are finished, data integrity must be double-checked. After that, according to the applicable rules of digital forensics, you should only work with an exact copy of the collected data.

The analysis and presentation of the data collection can then be used to report on the objectives of the investigation.

6.3. Legal Problems and Conflicts

In addition to technical problems and conflicts in digital forensics (discussed in 5.3.) in network systems, legal and ethical questions also regularly arise during investigations [Rü23].

It is crucial to determine which specific data from the data collection is pertinent for further investigations. Forensic activities can grant access to personal or sensitive medical data, which is particularly protected by law and demands special technical safeguards.

These eventualities should be clarified before data collection begins, and should be legally verified at the latest when the existence of such data becomes known.

From a legal point of view, the GDPR, national data protection regulations and the various criminal procedure codes and police duty laws must be complied with in Europe.

This means that not all data may generally be stored, processed, or even viewed as part of investigations. Specific types of data, such as legal information, financial and banking details, sensitive medical records, proprietary company secrets, government information, etc., are protected by strict legal provisions.

This all shows that, as part of the forensic investigation process, it must also be considered whether all data really needs to be fully accessed. The risk of compromise or data breach increases with every reproduction of this sensitive data, especially because encryption of the raw data usually has to be avoided in the analysis process.

Balancing the required data collection is not necessarily the responsibility of the forensic scientist, but is always a joint process between technical specialists and those legally responsible. Ideally, this decision-making process always involves several different authorities, who have to critically question the purpose of the forensic activities.

In principle, these conflicts and problems should be pointed out without conclusively fixing the legal framework. As always, it all depends on the aim and purpose of the forensic investigation and the legal position of the investigators or examiners.

Another principle that must be mentioned is that the network data collected is almost always particularly sensitive and must therefore be protected from access by third parties. Unauthorised parties could analyse network vulnerabilities, blackmail, steal or publish sensitive data.

In summary, legal and ethical considerations in network forensics require a nuanced approach that balances the need for thorough forensic investigations with the stringent requirements of data protection laws. The involvement of multiple authorities and the careful evaluation of the necessity and extent of data access are crucial to conducting legally compliant and ethically sound forensic investigations. These principles help safeguard sensitive information and maintain the integrity of the forensic process while respecting individuals' data protection rights.

Ideally, transnational standards will be developed and discussed by means of dynamic, open debates in democratic processes.

List of Abbreviations

BSI	Bundesamt für Sicherheit in der Informationstechnik (Federal Office for Information Security)
Botnet	Network of Robotted Network Devices
CLI	Command Line Interface
CMD	Command Line
CSD	Computer Security Division
DDoS	Distributed Denial of Service
DFIR	Digital Forensic and Incident Response
DFWM	Digital Forensic Workflow Model
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
EAP	Extensible Authentication Protocol
ENISA	European Union Agency for Cybersecurity
GDPR	General Data Protection Regulation (German: DSGVO)
IEC	International Electrotechnical Commission
IEEE	Institute Of Electrical And Electronic Engineers
IP	Internet Protocol
IPSec	Secure Internet Protocol
ISO	International Organisation for Standardisation
LAN	Local Area Network

LLDP	Link Layer Discovery Protocol
MPPE	Point-To-Point Encryption Protocol (Microsoft)
NIC	Network Interface Connection
NIST	National Institute of Standards and Technology
OSI	Open Systems Interconnection
PPTP	Point-To-Point Tunneling Protocol
RAM	Random Access Memory
SDN	Software Defined Network
SSH	Secure Shell
SSL	Secure Sockets Layer
TAP	Test Access Point
TCP	Transport Control Protocol (Network Protocol)
TLS	Transport Layer Security (Protocol)
UI	User Interface
UDP	User Datagram Protocol
URL	Uniform Resource Locator
VDI	Verein Deutscher Ingenieure
VPN	Virtual Private Network
WAN	Wide Area Network
WLAN	Wireless Local Area Network

List of Figures

All figures without sources were created by the author Florian Weijers.

List of Tables

All tables without sources were created by the author Florian Weijers.

Bibliography

[Al11] Altheide, C. & Carvey, H. (2011). “Digital Forensics with Open Source Tools”. In Elsevier eBooks. https://doi.org/10.1016/b978-1-59749-586-8.00001-7

[As04] Ashcroft, J. & Daniels, D. J. & Hart, S. V. (2004). “Forensic Examination of Digital Evidence: A Guide for Law Enforcement” . U.S. Department of Justice, Washington. https://doi.org/10.1037/e378092004-001

[Ba18] Bartsch, M.: (2018)., “Cybersecurity Best Practices” ISBN 978-3-658-21654-2, Springer Verlag 2018

[Be16] Benzekki, K.; El Fergougui A.; Elbelrhiti Elalaoui A. (2016): “Software-Defined Networking (SDN): a Survey”, Security and Communication Networks, https://web.archive.org/web/20190724155640id_/https://onlinelibrary.wiley.com/doi/pdf/10.1002/sec.1737 accessed: 05.03.2024

[Bu06] Bunting, S. C.; Wei, W. (2006). “ENCASE Computer Forensics - The official ENCE: ENCASE Certified Examiner Study Guide”. https://openlibrary.org/books/OL18498381M/EnCase_computer_forensics accessed: 05.03.2024

[Bu11] Bundesamt für Sicherheit in der Informationstechnik, 2010. Leitfaden "IT-Forensik", Version 1.0.1 (März 2011), https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Cyber-Sicherheit/Themen/Leitfaden_IT-Forensik.pdf accessed: 01.03.2024

[Ca10] Casey, E. (2010). “Handbook of Digital Forensics and Investigation”. In Elsevier eBooks. https://doi.org/10.1016/c2009-0-01683-3

[Ch15] Chawki, M.; Darwish, A.; Khan, M. A.; Tyagi, S. (2015). “Cybercrime, Digital Forensics and Jurisdiction. In Studies in computational intelligence”. https://doi.org/10.1007/978-3-319-15150-2

[De15] Deutscher Industrie- und Handelskammertag e.V.: “Informationstechnologie 2100 - Definition des Sachgebietes - Fachliche Bestellungsvoraussetzungen” (DIHK, März 2015)

[Eb18] Ebner, S.: “Datenschutz bei Polizei und Justiz” (2018) Diplomarbeit JKU Linz 2018

[Ei17] Eibensteiner, S. (2017) “Modell einer IT-Strategie”. Masterarbeit. JKU Linz S. 11 ff

[Eu19] European Union Agency for Cybersecurity: “Introduction to Network Forensics Handbook” (ENISA, 2019) ISBN 978-92-9204-288-2

[Eu23] European Union's Seventh Framework Programme; European Informatics Data Exchange Framework For Courts And Evidence: “The Digital Forensics Tool Catalogue”, 2023, https://www.dftoolscatalogue.eu/ accessed: 30.04.2024

[Fe14] Feamster, N.; J. Rexford, J.; E. Zegura, E.:, “The Road to SDN: An Intellectual History of Programmable Networks,” ACM SIGCOMM Computer Communication Review, 2014 https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9331325/ accessed: 21.03.2024

[Fi06] Fischer, R.: “Netzwerk Forensik”, https://www.kriminalwissenschaft.de/Lehrmaterialien/IT-Security.pdf accessed: 02.03.2024

[Fr13] Frantzen, A.: “Einsatz der Netzwerkforensik für Ermittlungsbehörden” (LKA Hessen, 2013), https://cdn.netzpolitik.org/wp-upload/2013-12-19_CAST-Frantzen-Netzwerkforensik.pdf accessed: 23.10.2023

[Ga12] Ghabban, F.; Alfadli, I.;, Ameerbakhsh, O.; Amer AbuAli, A.; Al-Daquim, A.; Al-Khasawneh, M.: “Comparative Analysis of Network Forensic Tools and Network Forensics Processes” (International Conference on Smart Computing and Electronic Enterprise ICSCEE2012)

[Ga20] Gasior, D. (2020) “Basic SDNs. In: Resource Allocation for Software Defined Networks” Springer, Briefs in Computer Science. Springer, doi 10.1007/978-3-030-59098-7_2

[Gr17] Grote, M. (2017): “Netzwerkanalyse mit Wireshark - Datenverkehr scannen, protokollieren und analysieren” (heise.de, 2017) https://www.heise.de/download/blog/Netzwerkanalyse-mit-Wireshark-3806147?hg=1&hgi=1&hgf=false accessed: 01.03.2024

[Gs06] Gschonneck, A. (2006) “Computer Forensik in der Praxis” Heise CeBIT Forum 2006

[In19] Infosec: “Computer forensics: Network forensics analysis and examination steps [2019]”, https://resources.infosecinstitute.com/topics/digital-forensics/computer-forensics-network-forensics-analysis-examination-steps/ accessed: 24.10.23

[It23] IT-Forensik Wiki der HS Wismar: “Netzwerk-Forensik” https://it-forensik.fiw.hs-wismar.de/index.php/Netzwerk-Forensik accessed: 01.03.2024

[Ja21] Jaiswahl, A.: “Wirtual Private Network (VPN) & Its Type” Meteorid, 22.09.2021, https://cybermeteoroid.com/virtual-private-network-vpn-its-type/ accessed: 30.04.2024

[Jo16] Joshi, R.C., Pilli, E. S. (2016): “Fundamentals of Network Forensics” (Springer, London) ISBN 978-1-4471-7297-0

[Ke06] Kent, K.; Chevalier, S.; Grance, T.; Dang, H.:: “NIST SP 800-86 Guide to Integrating Forensic Techniques into Incident Response” NIST, 2006 https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-86.pdf accessed: 22.10.2023

[Kr14] Kraft, P.;Weyert, A.: “Network Hacking” Franzis Verlag, ISBN 978-3645603171, 2014

[La17] Labudde, D.; Spranger, M.: “Forensik in der digitalen Welt” Springer-Spektrum 2017, ISBN 978-3-662-53800-5

[La20] Lambertz, M.; Hilgert, J.N.;: “Einführung in die Netzwerkforensik” Fraunhofer FKIE, 2020, https://www.cybersicherheit.fraunhofer.de/content/dam/zv/cybersicherheit-zv/documents/brosch%C3%BCren/einzelbrosch%C3%BCren/it-forensik/Einf%C3%BChrung%20in%20die%20Netzwerkforensik_V1.pdf accessed: 02.05.2024

[Ma07] Marcella, A.; Menendez, D. (2007). “Cyber Forensics: A Field Manual for Collecting, Examining, and Preserving Evidence of Computer Crimes”, Second Edition. https://openlibrary.org/books/OL23611593M/Cyber_forensics accessed: 01.05.2024

[Mo23] Morales, C.: “What is Network Forensic and why it is important” ip2location, 19.12.2023, https://medium.com/ip2location/what-is-network-forensic-and-why-it-is-important-3e2c0a8328a8 accessed: 29.04.2024

[Ni18] Ningsih, S.; Prayudi, I. R. Y. (2018). “Digital Forensics Workflow as A Mapping Model for People, Evidence, and Process in Digital Investigation”. International Journal Of Cyber-security And Digital Forensics, 7(3), 294–304. https://doi.org/10.17781/p002463

[Pi10] Pilli, E.S.; Joshi, R.C.;, Niyogi, R.: “Network forensic frameworks: Survey and research challenges” (Indian Institute of Technology Roorkee, 2010), https://www.academia.edu/79202692/Network_forensic_frameworks_Survey_and_research_challenges accessed: 23.10.2023

[Ra17] Rat der Europäischen Union: "Praktische Umsetzung und Durchführung europäischer Strategien zur Verhütung und Bekämpfung von Cyberkriminalität" (Brüssel, 2017) https://www.parlament.gv.at/dokument/XXV/EU/144067/imfname_10721489.pdf accessed: 23.10.2023

[Ri20] Rico, J. (2020): “ZeroTier - A smart p2p VPN solution” Rico´s blog, (25.10.2020) https://blog.rico-j.de/zerotier-one/ accessed on 30.04.2024

[Rü23] Rüdiger, T.G., Bayerl P.S. (2023):“Handbuch Cyberkriminologie 1 - Theorien und Methoden” (2023) Teil III: Rechtliche Grundlagen, Springer VS, ISBN 978-3-658-35438-1

[Sa16] Sachowski, J. (2016): “Implementing Digital Forensic Readlines”, Syngress 2016, ISBN 978-0-12-804454-4

[Sc21] Schreiber, F. (2021): “Network Forensics - State of the Art” Diplomarbeit, FH St. Pölten 2021

[Sc24] Schlottag, S.; Uslenghi, F.; Langner, J-M. (2024): “Rekorde bei Steam: Die 10 Spiele mit den höchsten Spielerzahlen“ (GameStar, 25.01.2024) https://www.gamestar.de/artikel/rekorde-steam-10-spiele-hoechsten-spielerzahlen,3324096,seite2.html accessed: 03.03.2024

[So20] Song, J.; Li, J. (2020): “A Framework for Digital Forensic Investigation of Big Data” 2020, 3rd International Conference on Artificial Intelligence and Big Data (ICAIBD), OI: 10.1109/ICAIBD49809.2020.9137498

[Sp17] Spiekermann, D. (2017): “Netzwerkforensik in virtuellen Umgebungen”, (Dissertation, FernUniversität in Hagen, BoD, 2017) https://ub-deposit.fernuni-hagen.de/rsc/viewer/mir_derivate_00001226/Diss_Spiekermann_Netzwerkforensik_2017.pdf?page=1 accessed: 12.04.2024

[St22] Stubbig, M. (2022): ZeroTier: The Internet as a Global Switch” OpenSourceForU.com (07.03.2022), https://www.opensourceforu.com/2022/03/zerotier-the-internet-as-a-global-switch/ accessed: 30.04.2024

[Th10] The Daily Telegraph (2010): “FBI arrests 'mastermind' of Mariposa botnet computer code“ (The Daily Telegraph, London, 28.07.2010) https://www.telegraph.co.uk/technology/7913767/FBI-arrests-mastermind-of-Mariposa-botnet-computer-code.html accessed: 01.03.2024

[Va13] Vaarand, R.; Niziński, P. (2013): “A Comparative Analysis of Open-Source Log Management Solutions for Security Monitoring and Network Forensics”, Tallinn 2013, https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.428.6404#citations accessed: 02.10.2023

[Vo08] Volonino, L. & Anzaldua, R. (2008) “Computer Forensics for Dummies”. Wiley. (2008). Part III

[Ze24] ZeroTier Inc. (2024): “ZeroTier Documentation”, 2024 https://docs.zerotier.com/ accessed: 02.05.2024