# Traffic Monitoring

Reading

45 minutes

✓ Status | Incomplete

## Introduction

Before you start monitoring a network for Cyber Security, you must first understand how network administrators monitor the network infrastructure services.

When networks start to flick or fail, the great flow of traffic information required by applications and business operations can stop completely.

The network administrators are continually requested to add new services, users, technologies and applications to the network. Still, these changes can affect your ability to provide consistent and predictable network performance. When network problems arise, network administrators are hard-pressed to verify the problem's root cause before it impacts the entire network.

> TIP: An intermittent network performance issue is difficult to replicate and diagnose.

## Reading

In this reading you will learn about the five essential functions of a **Network Monitoring System (NMS)** and why they are important.

An **NMS** provides five essential functions:

- Discover
- Map
- Monitor
- Alert
- Report

# Discover

**NMSs** discover the devices on the network, such as routers, switches, firewalls, servers, printers, and running services. The **NMSs** can include a library of monitoring templates, which define which part or how to monitor a device and its services.

All these functions are typically vendor-specific. For example, when you monitor a Cisco router, it will be different than when you monitor an Enterasys router, so when an NMS completes the discovery process, it automatically assigns an appropriate device role to each discovered device, independent of the brand.

A monitoring system with Layer 2/3 discovery will discover **port-to-port connectivity** between devices on the network, including **MAC** and **IP addresses**. For effective network monitoring, more information is needed to know what is working on the network, and it is important for you to understand how and where everything is connected.

Why? Because a performance issue with one device or a port can affect the performance of another connected device. For example, when a switch fails, all devices connected to that switch cannot communicate over the network.

# Map

NMSs generate network maps, a powerful first-response tool that allows network administrators to visualize their networks. Visualizing your network can save you hours and even days of troubleshooting network issues. Unfortunately, network wiring can get complex and messy, limiting the network's administrative ability to view the network and preventing troubleshooting.

Many NMSs require significant manual processing to create a network map but there is some software that provides a drawing tool. To fully utilize this type of software, a network administrator's knowledge and time will be important to map the network topology.

# Monitor

NMSs expose network administrators to many monitors and templates. As a starting point, network administrators want to monitor the five necessary items for any device on the network, such as **availability, latency, CPU, memory, disk, and interface utilization**. Most network monitoring tools monitor other hardware components such as power supplies, temperature, and fan speed.

When properly configured, an NMS will usually provide a visual dashboard of how the devices on the network are functioning, which will provide real-time awareness of failures, and may provide advance warning of imminent failure. Visual or audio feedback (flashing lights, colour changes, audible alarms, or text/email warnings) will alert administrators to the hardware issues that may result in physical failure of the network.

In addition, activity on a device that is outside the scope of its ordinary usage may indicate an attack or other threat. A little-used server that suddenly sees a huge spike in its uptime or CPU usage warrants investigation into the cause of the change in the device's behaviour.

# Alert

The central and essential feature is monitoring and notifying when something goes wrong. Performance metrics like CPU, memory and interface utilization can fluctuate throughout the day, but they may exceed the limits by a few seconds or minutes during peak usage periods. **Threshold-based alerting** allows network administrators to respond to issues before they impact users, applications, or the business. For example, the monitoring system is configured to issue an alert when CPU utilization on a router exceeds 80%, which allows the network administrator to investigate and respond before the router fails.

**Central Alert Management**: NMSs provide real-time and historical monitoring data to support the lifecycle. This information allows network administrators to:

- Validate that network projects are delivering desired results.
- Expose trends that could impact the network's ability to deliver the performance required by users, applications and businesses.
- Quickly isolate and fix performance issues.
- Prove the availability of the network and the connected devices.

> ℹ TIP: Most NMSs are customizable, and you can create profile-based dashboards for managers, line-of-business owners, Help Desk, and applications. NMSs differ in their capabilities for each function, but they have the same result.

# How Network Monitoring Tools Collects Information

Network monitoring tools collect information on the network by using poll connection over the network devices and servers. They receive information about the performance data using standard protocols such as:

- Agents: using specific agents in the case of servers
- SNMP
- **Windows Management Instrumentation (WMI)**
- Sniffing the traffic and flow of pieces of information
- Remote SSH for Unix and Linux servers

**WMI** implementation is a proprietary protocol for Windows-based systems and applications of Web-Based Enterprise Management, a software industry initiative to develop a standard for accessing management information across the enterprise. This protocol creates an OS interface that receives information from devices running a WMI agent. WMI gathers details about the OS, hardware or software data, the status and properties of remote or local systems, configuration and security information, and process and service information. It then passes all these details to the network management software, which monitors the network's health, performance, and availability.

> ⚠ Microsoft has changed its policy regarding the code it uses to execute WMI, which is important if PowerShell is being used. There is no indication what the PowerShell users should be aware of. Some WMI commands in PowerShell are no longer supported, and they should be replaced with CIM cmdlets (commandlets) code.

# Monitoring the Network with Wireshark

Wireshark allows the user to analyze packets transmitted over the network, displaying the information in detail in a graphical interface. With this, it is possible to identify, with more details about everything that happens in the data exchange between the client and host. The program's graphical interface is the main difference between Wireshark and the **TCPDUMP** (command-line interface) tool.

TCPDUMP provides the same details of network packet information while Wireshark simply presents the data in a dashboard visual. TCPDUMP is included in the OS as a command-line tool, meaning it's free to use.

> Network administrators can therefore choose which option suits their style and budget.

Wireshark offers a series of facilities depending on the user's knowledge level rather than the graphic resources. You can also run it using command lines. Both solutions use the same library **(libpcap)**, which is specialized for capturing and sending packets or files over a network. Therefore, Wireshark supports **TCPDUMP** capture documents.

With a wide range of functionalities and features, Wireshark has boost functionality, such as:

- Real-time data reading: All packet contents sent/received by the server are captured in real time by Wireshark. User interaction with your network can be monitored anytime. This is an essential feature for tracking suspicious events.
- Availability and interoperability for multiple OSs: All popular OSs like Linux, Mac and Windows have a compatible version of Wireshark.

> The tool is also available for FreeBSD, NetBSD, Solaris, HP-UX, etc.

- Easy and intuitive filters to understand the total traffic and network communication.

# Applying Filters

Among the most remarkable things you can do in Wireshark is to apply content filters. All you have to do through them is type the terms following a syntax or use certain filters that make the process easier. Here are five practical examples of how to inspect traffic closely.

## Example 1

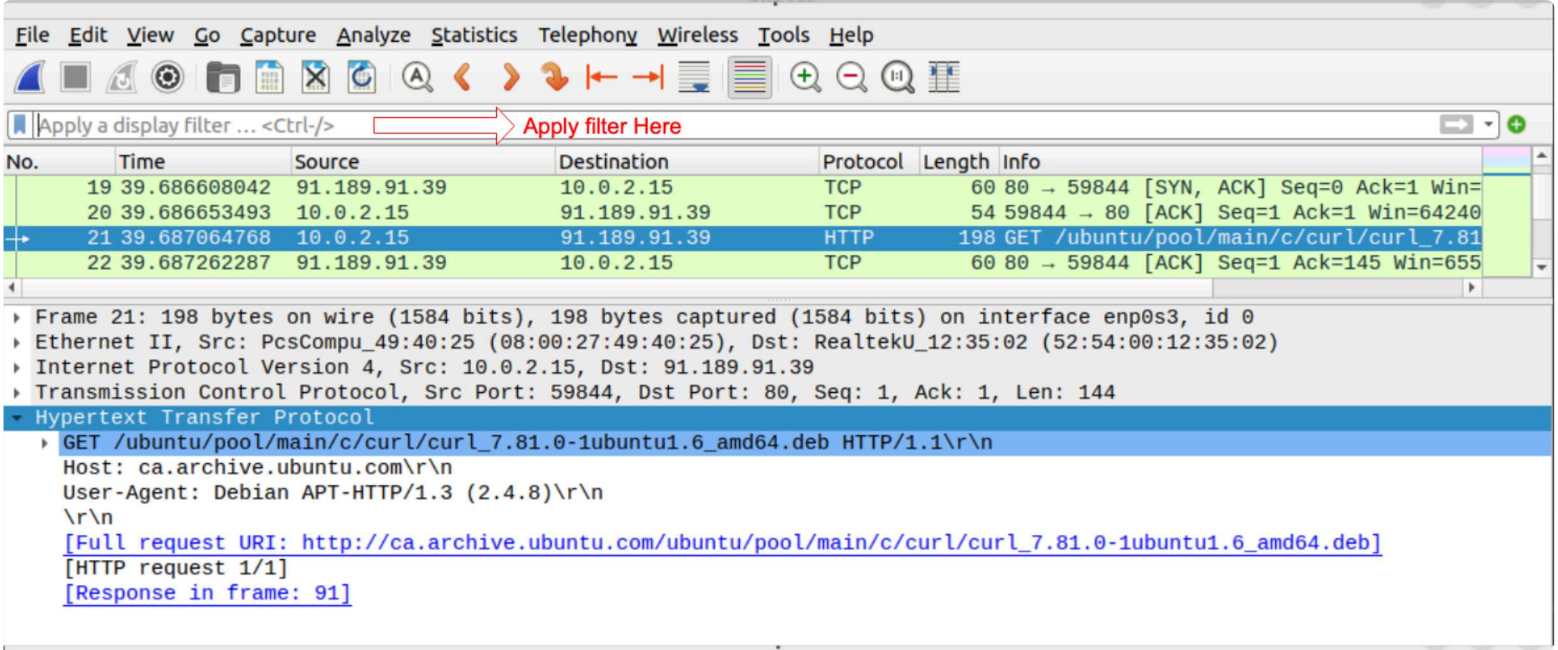
If you wanted to filter packets by source/destination IP or monitor traffic from an IP address

In Wireshark, we can do this via the source and/or destination IP, as per the examples

> There is a handy filter to refine the search, restricting it to local network traffic, thus reducing the amount of information displayed in the results.

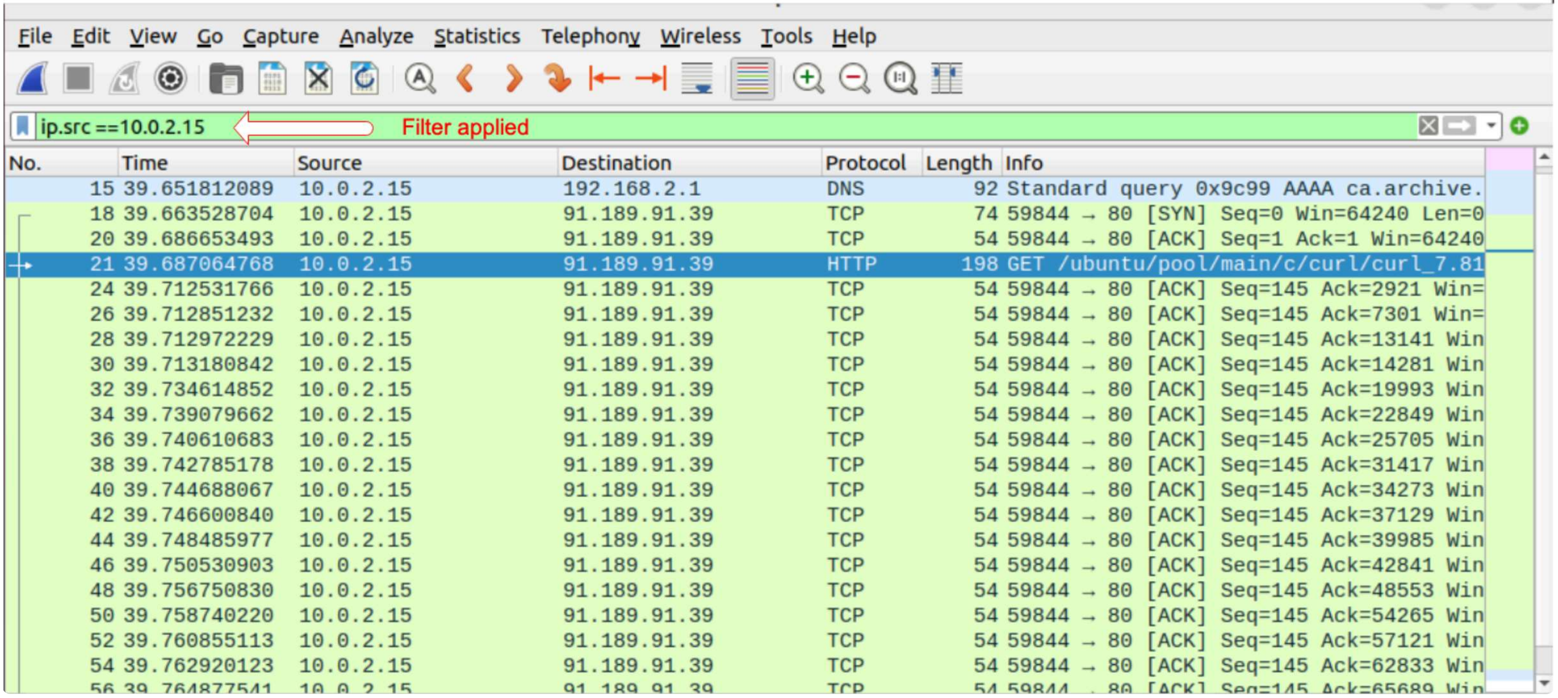
```
ip.src==192.168.1.15


ip.dst==192.168.16.8
```


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Apply filter Here

```
No.   Time          Source          Destination     Protocol Length Info
   19 39.686608042  91.189.91.39    10.0.2.15       TCP          60 80 → 59844 [SYN, ACK] Seq=0 Ack=1 Win=
   20 39.686653493  10.0.2.15       91.189.91.39    TCP          54 59844 → 80 [ACK] Seq=1 Ack=1 Win=64240
   21 39.687064768  10.0.2.15       91.189.91.39    HTTP        198 GET /ubuntu/pool/main/c/curl/curl_7.81
   22 39.687262287  91.189.91.39    10.0.2.15       TCP          60 80 → 59844 [ACK] Seq=1 Ack=145 Win=655
```

```
▸ Frame 21: 198 bytes on wire (1584 bits), 198 bytes captured (1584 bits) on interface enp0s3, id 0
▸ Ethernet II, Src: PcsCompu_49:40:25 (08:00:27:49:40:25), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
▸ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 91.189.91.39
▸ Transmission Control Protocol, Src Port: 59844, Dst Port: 80, Seq: 1, Ack: 1, Len: 144
▾ Hypertext Transfer Protocol
  ▸ GET /ubuntu/pool/main/c/curl/curl_7.81.0-1ubuntu1.6_amd64.deb HTTP/1.1\r\n
    Host: ca.archive.ubuntu.com\r\n
    User-Agent: Debian APT-HTTP/1.3 (2.4.8)\r\n
    \r\n
    [Full request URI: http://ca.archive.ubuntu.com/ubuntu/pool/main/c/curl/curl_7.81.0-1ubuntu1.6_amd64.deb]
    [HTTP request 1/1]
    [Response in frame: 91]
```


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ip.src ==10.0.2.15
Filter applied

```
No.   Time          Source          Destination     Protocol Length Info
   15 39.651812089  10.0.2.15       192.168.2.1     DNS          92 Standard query 0x9c99 AAAA ca.archive.
   18 39.663528704  10.0.2.15       91.189.91.39    TCP          74 59844 → 80 [SYN] Seq=0 Win=64240 Len=0
   20 39.686653493  10.0.2.15       91.189.91.39    TCP          54 59844 → 80 [ACK] Seq=1 Ack=1 Win=64240
   21 39.687064768  10.0.2.15       91.189.91.39    HTTP        198 GET /ubuntu/pool/main/c/curl/curl_7.81
   24 39.712531766  10.0.2.15       91.189.91.39    TCP          54 59844 → 80 [ACK] Seq=145 Ack=2921 Win=
   26 39.712851232  10.0.2.15       91.189.91.39    TCP          54 59844 → 80 [ACK] Seq=145 Ack=7301 Win=
   28 39.712972229  10.0.2.15       91.189.91.39    TCP          54 59844 → 80 [ACK] Seq=145 Ack=13141 Win
   30 39.713180842  10.0.2.15       91.189.91.39    TCP          54 59844 → 80 [ACK] Seq=145 Ack=14281 Win
   32 39.734614852  10.0.2.15       91.189.91.39    TCP          54 59844 → 80 [ACK] Seq=145 Ack=19993 Win
   34 39.739079662  10.0.2.15       91.189.91.39    TCP          54 59844 → 80 [ACK] Seq=145 Ack=22849 Win
   36 39.740610683  10.0.2.15       91.189.91.39    TCP          54 59844 → 80 [ACK] Seq=145 Ack=25705 Win
   38 39.742785178  10.0.2.15       91.189.91.39    TCP          54 59844 → 80 [ACK] Seq=145 Ack=31417 Win
   40 39.744688067  10.0.2.15       91.189.91.39    TCP          54 59844 → 80 [ACK] Seq=145 Ack=34273 Win
   42 39.746600840  10.0.2.15       91.189.91.39    TCP          54 59844 → 80 [ACK] Seq=145 Ack=37129 Win
   44 39.748485977  10.0.2.15       91.189.91.39    TCP          54 59844 → 80 [ACK] Seq=145 Ack=39985 Win
   46 39.750530903  10.0.2.15       91.189.91.39    TCP          54 59844 → 80 [ACK] Seq=145 Ack=42841 Win
   48 39.756750830  10.0.2.15       91.189.91.39    TCP          54 59844 → 80 [ACK] Seq=145 Ack=48553 Win
   50 39.758740220  10.0.2.15       91.189.91.39    TCP          54 59844 → 80 [ACK] Seq=145 Ack=54265 Win
   52 39.760855113  10.0.2.15       91.189.91.39    TCP          54 59844 → 80 [ACK] Seq=145 Ack=57121 Win
   54 39.762920123  10.0.2.15       91.189.91.39    TCP          54 59844 → 80 [ACK] Seq=145 Ack=62833 Win
   56 39.764877541  10.0.2.15       91.189.91.39    TCP          54 59844 → 80 [ACK] Seq=145 Ack=65689 Win
```

A green display filter bar indicates a successful filter.

## Example 2

Below, we apply the conditional and, as a reference, the IP 192.168.0.0/24 common in local networks.

As in the examples above, it is only sometimes necessary to resort to filter fields. Wireshark can perform **string** searches, which speeds up the work considerably.

For this, you use the `contains` operator to perform this search.

```
ip.src==192.168.0.0/24 and ip.dst==192.168.0.0/24
```

## Example 3

Suppose you are looking for a particular package in an extensive set of packages.

To find strings over the packet: Wireshark will identify the packages that contain the string to confirm the existence and check the inner contents of the packet.

Knowing that the URL of our blog is present in the transaction, you can use it as a reference by typing:

```
http contains "www.lighthouselabs.ca"
```

## Example 4

To filter packets from TCP or UDP ports: Searching for packets using as a reference a port (TCP or UDP) used in the manipulation is very simple.

In the example below, you use the filter port:

```
tcp.port==357

udp.port==4113
```

## Example 5

Monitor traffic on Apache and MySQL networks: Wireshark allows us to condition the use of resources.

In a network installed by the LAMP suite, for example, know the corresponding server networks by the LAMP and web database (MySQL or MariaDB).

```
tcp.port==80 || tcp.port==3306
```

Suppose you are interested in the study of Wireshark filters. In that case, it is recommended that you base yourself on this document from the official website, which contains all references to elements usable in filters.

# Further Reading

Wireshark

# References

1. Paessler
2. Nagios
3. About Cacti

# Review Questions

Answer all of the questions below to review your understanding. Try to answer them in your own words.

? Imagine you are a network administrator using an NMS. You receive an alert that the CPU utilization on a core router has exceeded 80%. What steps would you take to investigate and resolve this issue before it impacts the network?

**Your Answer**

Type in your answer here and Compass will let you reveal our answer below. Compass will auto-save your answer as you type. Once you click Toggle Answer below, your answer cannot be changed.

Toggle Answer

? You are using Wireshark to monitor network traffic and notice a sudden spike in CPU usage on a normally idle server. What steps would you take to investigate whether this spike indicates a potential security threat?

**Your Answer**

Type in your answer here and Compass will let you reveal our answer below. Compass will auto-save your answer as you type. Once you click Toggle Answer below, your answer cannot be changed.

Toggle Answer

✓ Mark Completed

← Previous
Logs

Next
Network Monitoring with Wireshark →

## How well did this activity help you to understand the content?

Let us know how we're doing

☆ ☆ ☆ ☆ ☆

# W01D5

Fri Jun 28

## Lectures (1)

## Work (11)

**7 hrs**

- Traffic Monitoring
- Network Monitoring with Wireshark
- Intro to Network Baselines
- Visualizing What's Happening Lab - Windows Logging
- Group Share and Feedback
- Recap Terminal Commands: Windows and Linux
- Understand the environment and Wireshark
- Check the Commands
- Tech Interviews
- TIs: How it Works ✓
- TIs: What to Expect ✓

## Other (1)

W01D5 Schedule »