Tutorial:

Use nmap to uncover security gaps in the network

Behind every open port in the network is a server program which in all likelihood contains errors. These errors may be so severe that they can be exploited to spy on the system, sabotage it or even run malicious code on it. And so the most important step you can take to greater security in your corporate network is to check the open ports and - if need be - to close them.

Introduction: Using nmap to explore the network

Using network analysis programs you can get an overview of possible weaknesses in your corporate network. This guide introduces three basic analysis techniques using the port scanner nmap. They are designed to identify terminal devices and discover which services they provide. We then show two techniques you can use to secure the found weak points.

Using the program **nmap** you can explore the structure of your network. It is essentially a hacker tool, since it uses implementation tools from various network protocols. It finds terminal devices in the network, analyzes them for running server programs and determines which programs are being used. In this way nmap provides a vast amount of details about your corporate network.

Note:
Be sure you always work with care and apply nmap only to networks you are authorized to investigate!

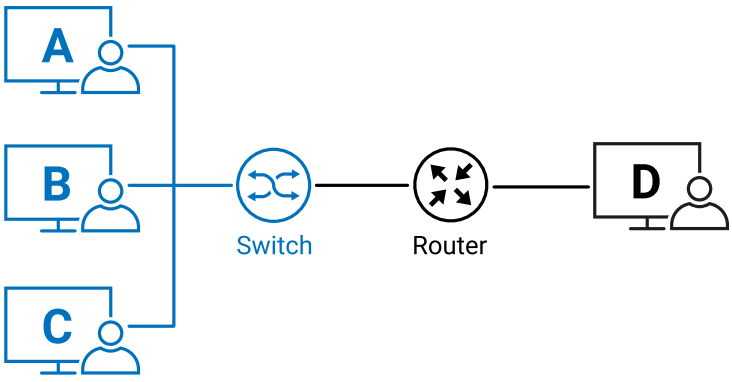
Installing and using nmap

Go to nmap.org to download the program. Whereas nmap itself is simply a command line program, Zenmap provides a graphical user interface.

In some of the techniques described below nmap sends raw data packets in the network across various layers. Since this requires bypassing the network stack of the operating system, running it requires root and administrator authorization.

Analysis technique 1: Using nmap to find terminal devices in the network

The first technique finds all the terminal devices located in the same broadcast domains. This means: All computer for example are connected to each other in the same Ethernet via switches and not separated from each other by routers (i.e. on the IP level). One could of course send an ICMP echo request (ping) to the broadcast address of the network segment, but for security reasons current terminal devices will not generally respond to this.



Computers A, B and C are connected to each other through a switch and are thus in the same broadcast domain. Computer A can find Computers B and C in the network using ARP requests. Computer D is also connected to the network through a router. ARP requests do not reach it however.

If you want to send an IP packet to another computer in the same Ethernet, this is done using an Ethernet frame which is exchanged between two network interfaces. This requires that the hardware address of the destination be known. The mechanism which assigns the associated hardware and MAC address to an IP address is provided by the Address Resolution Protocol (ARP).

Using the broadcast MAC address in the network this request is sent to all terminal devices in the broadcast domain: "Who

has the following address?". The terminal device which receives packets for this IP address then sends back its MAC address. Now the network interface can address all data to exactly this device.

When inventorying a network segment using ARP this request is made for every single IP address in the corresponding network segment. If a device is located in the network, it must reply to this request.

The following command finds all running terminal devices having IP addresses 192.168.1.1 to 192.168.1.254 in a common broadcast domain:

```
nmap -PR 192.168.1.0/24
```

Analysis technique 2:
Finding open ports without direct access to the device

If you have direct access to the destination system, you can show open ports under Linux, MacOS and Windows using the command "netstat". This is however not possible in many cases. For these situations nmap offers alternatives for searching the destination system for open ports.

Finding open TCP ports

Behind every open port is a server program which receives and processes data. In the case of TCP identifying open ports is simple: Since the protocol works based on connections, you only need to use a three-way handshake for opening the connection (more information about three-way handshakes can be found in our free book "TCP/IP Ethernet to the Web-IO"). Here a simple SYN packet is sent to the destination. Depending on its response the status of the port can be determined:

- **No reply:**
  The port is either not open or is filtered.
- **Reset:**
  The operating system does accept connection requests for the port, but returns them because there is no server application running on the port.
- **Connection accepted:**
  The port is open and may represent a security risk

Using the following command you can perform a TCP-SYN scan on the host with the address 192.168.1.200:

```
nmap -sS 192.168.1.200
```

Finding open UDP ports

Finding open UDP ports is significantly more difficult. Unlike TCP, UDP is not connection-based. Since there is no three-way handshake, the reply from the other end cannot be predicted.

In the simplest case the operating system in the destination replies that the port cannot be reached. In this case it is marked as closed. If no reply is sent, it is not possible to say whether the packet was received by a server application or rejected somewhere along the line. So no reply means that the port is either open or filtered.

A service-specific packet is sent to the ’usual suspects" among the ports. If the sender receives a reply, he knows that the port is open. This works for example in resolving computer names using DNS, when requesting an IP configuration using DHCP, or for time updates via NTP.

This command performs a simple UDP scan:

```
nmap -sU 1 192.168.1.200
```

As noted above, this scan does not necessarily find all open UDP ports, but for an initial overview it is often sufficient.

Finding open TCP and UDP ports in the same run

To find both open TCP and UDP ports at the same time you can also combine the commands:

```
nmap -sS -sU 192.168.1.200
```

Analysis technique 3:
Determining the operating system and server programs

Nmap can do more than just find terminal devices and ports: when implementing network protocols developers have certain freedoms. This ensures that each operating system has a specific fingerprint by which it can be identified.

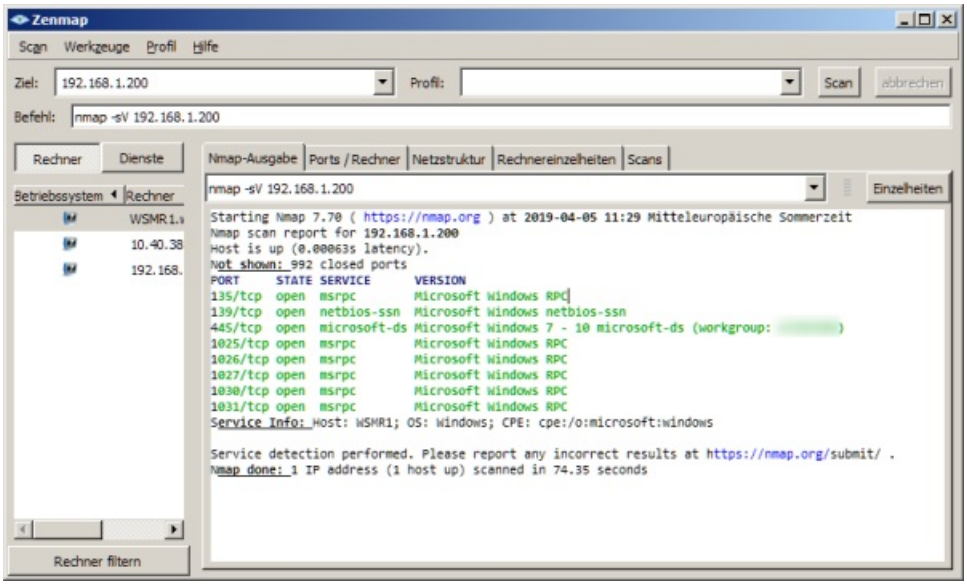
Identifying the operating system does not always work. But in most cases the result is accurate.

```
nmap -0 192.168.1.200
```

To find out which services are running on a terminal device, nmap can analyze the replies from the open ports. These programs often send out a great amount of information on their own, for example their version number or which protocols

they support. Nmap evaluates this information and summarizes it.

`nmap -sV 192.168.1.200`



Please note: If you use sV together with sU, i.e. the UDP scan, nmap sends a battery of wide ranging test packets to each individual UDP port. This can provide you with additional information about UDP ports but may in some cases take a very long time!

Closing ports, increasing security

Once any risks from open ports are known of, the next step is to minimize them.

Security technique 1:
Quit server applications

As a first step you should disable all unneeded server applications. To find out which application opens a port, you can again use the "netstat" command. In many cases quitting server applications on the terminal device for closing ports is not possible, for example without the proper access rights, in cases of embedded software, or when server applications are started dynamically as needed. Here the isolation technique described in the next section can help.

Security technique 2:
Uncomplicated and effective - isolating using Microwall

When using isolation technique potentially at-risk systems whose open ports cannot be closed are isolated in their own network segment. Communication with this segment is monitored, limited and logged. In addition a firewall router such as the W&T Microwall VPN is installed between the affected system and the surrounding network. The IP data traffic routed through this path is filtered according to rules.

- The packet filter of the Microwall filters out undesirable communication and rejects packets. Even if a port on the terminal device is open it cannot be reached.
- In NAT mode all terminal devices are hidden on the island behind the Microwall. The only thing visible in the network is the Microwall, which passes on and monitors communication.
- Isolated terminal devices are in another broadcast domain. An ARP request and other attacks are effectively suppressed.

Detailed information about the isolating strategy can be found on the topic page.

The proof of the pudding is in the eating!

We are happy to provide you with a Microwall at no charge for a period of four weeks.

Request test unit

Thomas Clever
t.clever@wut.de

You can reach our engineers by phone at +49 202/2680-110 (Mon.-Fri. 8 a.m. to 5 p.m.)

W&T
www.WuT.de

We are available to you in person:

Wiesemann & Theis GmbH
Porschestr. 12
42279 Wuppertal
Phone: +49 202/2680-110 (Mon.-Fri. 8 a.m. to 5 p.m.)
Fax: +49 202/2680-265
info@wut.de

Data Privacy