UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

Analýza principů a funkcí Firewall na IPv6

Bc. Michal Třetina

Diplomová práce
2015

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: **2014/2015**

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Michal Třetina**
Osobní číslo:	**I13454**
Studijní program:	**N2646 Informační technologie**
Studijní obor:	**Informační technologie**
Název tématu:	**Analýza principů a funkcí Firewall na IPv6**
Zadávající katedra:	**Katedra softwarových technologií**

Zásady pro vypracování:

Cílem diplomové práce je podrobně představit principy, funkcionality a nedostatky firewallů fungujících nad IPv6. V teoretické části autor podrobně představí principy Firewallů, jejich možnosti a nasazení. Dále autor provede analýzu možných nedostatků fungování firewallů nad IPv6, jež jsou spojeny právě s využíváním IPv6 protokolu na síťové vrstvě modelu ISO/OSI. Získané poznatky autor prakticky ověří na standardních firewall řešeních pro operační systémy Windows a GNU/Linux.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

MORAES, Alexandre Matos da Silva Pires de. Cisco firewalls. Indianapolis: Cisco Press, c2011, xxii, 889 s. ISBN 978-1-58714-109-6.
FRAHIM, Jazib. Cisco Asa: all-in-one next-generation firewall, IPS, and VPPN services. 3rd Ed. Indianapolis: Cisco Press, c2011, pages cm. ISBN 978-158-7143-076.
SHINDER, Thomas W. Best damn firewall book period: all-in-one next-generation firewall, IPS, and VPPN services. 2nd ed. Oxford: Elsevier Science [distributor], c2007, 1123 p. ISBN 15-974-9218-3.

Vedoucí diplomové práce: **Mgr. Josef Horálek, Ph.D.**
Katedra softwarových technologií

Datum zadání diplomové práce: **31. října 2014**
Termín odevzdání diplomové práce: **15. května 2015**

prof. Ing. Simeon Karamazov, Dr.
děkan

L.S.

prof. Ing. Antonín Kavička, Ph.D.
vedoucí katedry

V Pardubicích dne 15. listopadu 2014

Prohlášení autora

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 5. května 2015 Bc. Michal Třetina

Poděkování

Děkuji Mgr. Josefu Horálkovi, Ph.D. za vedení diplomové práce a veškerý jeho čas. Děkuji také rodičům za jejich podporu během celé doby mého studia.

Anotace

Cílem práce je podrobně představit principy, funkcionality a nedostatky firewallů fungujících nad IPv6. V teoretické části autor podrobně představí principy firewallů, jejich možnosti a nasazení. Dále autor provede analýzu možných nedostatků fungování firewallů IPv6, jež jsou spojeny právě s využíváním protokolu IPv6 na síťové vrstvě modelu ISO/OSI. Získané poznatky autor prakticky ověří na standardních firewall řešeních pro operační systémy Windows, GNU/Linux a další operační systémy typu Unix.

Klíčová slova

Firewall, IPv6, síťová bezpečnost, Scapy

Title

An analysis of principles and functions of Firewall over IPv6

Annotation

The aim of this thesis is to introduce principles, functionality and pitfalls of firewalls in the IPv6 environment. In the theoretical part, the author presents principles, scope and deployment of firewalls in detail. Next, the author performs an analysis of possible deficiencies of IPv6 firewalls, which are connected with the usage of the IPv6 protocol on the network layer of the ISO/OSI model. These findings were verified in practice on standard firewall solutions for operating systems Windows, GNU/Linux and other Unix or Unix-like systems.

Keywords

Firewall, IPv6, network security, Scapy

Obsah

Úvod 13

1 Přehled současného stavu problematiky 14
1.1 Request for Comments 14
1.2 Firewally a bezpečnost v prostředí IPv6 17
1.3 Shrnutí 18

2 Obecné principy a klasifikace firewallů 20
2.1 Firewally a model ISO/OSI 21
2.1.1 Paketový filtr 22
2.1.2 Stavový firewall 22
2.1.3 Aplikační brána 23
2.2 Hrozby 24
2.3 Firewally v síťové architektuře 25
2.3.1 Architektura Single-box 25
2.3.2 Architektura Screened Host 25
2.3.3 Architektura Screened Subnet 26
2.3.4 Architektura Split-Screened Subnet 26
2.3.5 Architektura Independent Screened Subnets 26
2.3.6 Host-Based Firewall 27

3 Firewally a protokol IPv6 28
3.1 Filtrování na základě adresy IPv6 28
3.2 Multihoming 29
3.3 Internet Control Message Protocol version 6 31
3.4 Rozšiřující hlavičky IPv6 33
3.4.1 Hlavičky Volby pro cíl, Volby Hop-by-hop 35
3.4.2 Hlavičky Autentizace a Šifrování obsahu 36
3.4.3 Hlavička Fragmentace 36
3.5 Mobile IPv6 38
3.6 Tunely 40

4 Praktické posouzení firewallů IPv6 44

4.1 Přehled použitých operačních systémů ... 44
4.2 Popis testovaných firewallů ... 45
4.2.1 Windows Firewall ... 45
4.2.2 Netfilter/iptables ... 45
4.2.3 IP Firewall ... 46
4.2.4 Packet Filter ... 47
4.2.5 IP Filter ... 48
4.3 Použitý hardware ... 48
4.4 Testovací scénáře a nástroj firewall6 ... 49
4.5 Praktický test firewallů pomocí firewall6 ... 52
4.6 Testovací scénáře a nástroj Scapy ... 53
4.7 Praktický test firewallů pomocí Scapy ... 59

Závěr ... 63

Použitá literatura ... 64

A Příloha I – Skript ... 71
A.1 NestedFragments ... 71

Seznam tabulek

1 Hrozby 24
2 Srovnání IPv4 a IPv6 28
3 Adresy IPv6 k blokaci na perimetru 30
4 Vybrané zprávy ICMPv6 32
5 Rozšiřující hlavičky IPv6 34
6 Operační systémy 44
7 Hardwarová konfigurace zdroje paketů 48
8 Sada testů č. 1 49
9 Získané výsledky – Firewall6 52
10 Sada testů č. 2 54
11 Sada testů č. 3 58
12 Získané výsledky – Scapy 59

Seznam obrázků

1 Nasazení firewallu 20
2 Model ISO/OSI a typy firewallů 21
3 Paketový filtr 22
4 Stavový firewall 23
5 Webová proxy 24
6 Alokace prefixů v IPv6 29
7 Hlavička IPv6 33
8 Řetězení rozšiřujících hlaviček IPv6 34
9 Hlavičky Voleb pro cíl a Hop-by-hop 35
10 Hlavička Fragmentace 37
11 Schéma komunikace Mobile IPv6 39
12 Tunelování 6in4 40
13 Testovací topologie 1 49
14 Testovací topologie 2 53

Seznam zkratek

ACK	Acknowledgement
ACL	Access Control List
API	Application Programming Interface
ASA	Adaptive Security Appliance
CLI	Command Line Interface
DNS	Domain Name System
DNSSEC	Domain Name System Security Extensions
DMZ	Demilitarizovaná Zóna
ESP	Encapsulating Security Payload
FTP	File Transfer Protocol
GNU	GNU's Not Unix!
ICMP	Internet Control Message Protocol
ICMPv6	Internet Control Message Protocol Version 6
IETF	Internet Engineering Task Force
IDS	Intrusion Detection System
IP	Internet Protocol
IPS	Intrusion Prevention System
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
ISO	International Standards Organization
ISP	Internet Service Provider
MTU	Maximum Transmission Unit
NAT	Network Address Translation
ND	Neighbor Discovery
OSI	Open Systems Interconnection
RFC	Request for Comments
SLAAC	Stateless Address Autoconfiguration

SYN	Synchronize
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
URL	Uniform Resource Locator
VPN	Virtual Private Network

Úvod

Vyčerpání adresního prostoru protokolu IPv4 je obecně známý a dlouho diskutovaný problém, nyní aktuálnější než kdy dříve, neboť 3. února 2011 byly alokovány poslední zbývající IPv4 adresy z jejich centrálního fondu [1]. Tím vzniká motivace pro přechod k protokolu IPv6, který adresní prostor rozšiřuje z bitů 32 na bitů 128 (např. mezi uživateli služeb Google již konektivita IPv6 přesáhla 4 % [2]). Zatímco jednotlivé mechanismy tohoto přechodu byly zevrubně diskutovány, implementace IPv6 v softwarových balících a hardware nejsou nezbytně tak vyspělé, otestované a popsané jako jejich IPv4 protějšky.

Velkou otázkou je bezpečnost tohoto nově vznikajícího prostředí. Přestože bezpečnost hrála při návrhu protokolu IPv6 významnou roli, ten je stále potenciálním vektorem pro nové typy síťových útoků. Nese s sebou i další potenciální důsledky pro síťovou bezpečnost, příkladem je odstranění potřeby překladu síťových adres vzhledem k dostatečně rozsáhlému adresnímu prostoru a konceptu end-to-end transparence.

Tato práce si bere za cíl popsat dopady změn v protokolu IPv6 vůči protokolu IPv4 na síťová zařízení typu firewall (obecně se může jednat o hardwarový nebo softwarový firewall – tato práce se bude soustředit zejména na softwarové implementace firewallů nad operačními systémy Windows, GNU/Linux a dalšími systémy typu Unix). V kapitole 1 práce shrnuje současný stav problematiky s pomocí vybrané literatury. Kapitola 2 definuje nezbytné teoretické pojmy a představí architekturu, taxonomii firewallů a principy, na nichž pracují. V kapitole 3 je provedeno hodnocení konkrétních nových vlastností protokolu IPv6 a jejich důsledků pro oblast firewallů a síťové bezpečnosti. Kapitola 4 se prakticky zabývá testováním firewallů IPv6 v rámci laboratorní, virtualizované topologie s použitím vybraných nástrojů. Závěr práce shrnuje a hodnotí získané poznatky a navrhuje další možné cesty výzkumu. Přílohy obsahují jeden převzatý skript nástroje Scapy.

1 Přehled současného stavu problematiky

V této kapitole bude představena a shrnuta relevantní literatura a výzkum, týkající se problematiky firewallů v prostředí IPv6. Pojmy zavedené v této kapitole budou detailněji definovány v kapitolách následujících.

1.1 Request for Comments

Request for Comments (dále RFC) se v oblasti počítačových sítí rozumí dokument publikovaný Internet Engineering Task Force (IETF), jenž popisuje specifikace, chování, procedury a protokoly Internetu. Autory jsou typicky výzkumníci v oblasti informatiky a počítačových sítí. Tyto dokumenty jsou recenzovány a mohou být přijaty jako standardy Internetu.

Jedním ze záměrů návrhu IPv6 je skoncovat s potřebou překladu síťových adres (NAT). Přes své nevýhody NAT také funguje jako bezpečnostní prvek [3] a jeho vynětím vzniká mezera, kterou je potřeba zaplnit. RFC 4864 [4] diskutuje základní techniky ochrany integrity síťové architektury IPv6. Firewall jakožto stavový filtr paketů zde může NAT plně zastoupit při obraně před nevyžádanými pokusy o spojení z vnějšku sítě a skrývání síťové topologie. Navíc nabízí explicitní a podrobný management této funkce, znamená to však také nutnou základní úroveň konfigurace firewallu.

Přechod k IPv6 však neproběhne ihned a najednou, po nějakou dobu budou protokoly IPv4 a IPv6 koexistovat. O bezpečnosti tohoto prostředí se (mimo jiné) hovoří v RFC 4942 [5]. Známými problémy vyplývajícími z dvojakosti tohoto prostředí jsou např. IPv4-mapované adresy IPv6, kdy útočník je schopen tvorbou cíleně formátovaných paketů obejít pravidla pro filtrování IPv4 paketů a překonat firewall. Pozornost je věnována i tunelování IPv6 skrze sítě IPv4. Zatímco např. IPv6-over-IPv4 (zkráceně 6over4) tunelování využívající portu 41 lze snadno povolit či zakázat, IPv6 zapouzdřené v UDP (tzv. Teredo) je nesnází, neboť průchod UDP musí být na firewallu typicky povolen. Ve firewallu tak vzniká bezpečnostní mezera [6]. Problematické to může být zejména pro sítě domácí a sítě malých podniků/kanceláří bez zkušených administrátorů. RFC 6092 [7] poskytuje sadu doporučení pro základní zabezpečení bran IPv6 také pomocí funkcí paketových filtrů pro výrobce

zařízení právě pro tyto uživatele. Filtrováním paketů IPv6 ve velkých „enterprise“ sítích, plánovaných původně jako pouze IPv4, se zabývá RFC 7123 [8]. Jako bezpečnostní scénáře vztahující se k problematice firewallů, které je nutné v takové síti vzít v úvahu, uvádí:

- Firewall IPv4 nemusí být vůbec schopen prosazovat stejnou bezpečnostní politiku pro provoz IPv6.
- Firewall může podporovat IPv4 i IPv6, ale nemusí být pro kontrolu provozu IPv6 korektně konfigurován.
- Výše zmíněné problémy s IPv4/IPv6 přechodovými mechanismy (Teredo).

Navrhuje také řízení bezpečnosti nativního provozu IPv6 a provozu IPv6 tunelovaného pomocí IPv4 v podobě politik filtrování pro blokování nežádoucího provozu. Dále diskutuje způsoby filtrování konkrétního typu síťového provozu a s ním spojené problémy.

Zmíněný dokument RFC 4942 [5] je poměrně obsáhlý a z hlediska čistě IPv6 problémů zmiňuje i problematiku směrovacích hlaviček (definovány v RFC 2460 [9]) a jejich možného zneužití pro obejití firewallu nebo pro amplifikační útok. Takový útok je zavážný a RFC 5095 [10] doporučuje všem IPv6 uzlům odstranit podporu směrovací hlavičky varianty 0 („Type 0 Routing Header“) a firewallům tento typ hlavičky filtrovat (nelze však obecně filtrovat veškeré směrovací hlavičky!). Potenciálně lze firewall obejít i pomocí směrovací hlavičky varianty 2 („Type 2 Routing Header“), která souvisí s podporou Mobile IPv6 (definuje RFC 6275 [11]), nelze ji však použít pro amplifikační útok. Podvržené pakety v chybových zprávách ICMPv6, filtrování provozu s adresami anycast, IPSec, postupy pro zpracování rozšířených nebo neznámých hlaviček firewally, obranu firewallem před zneužitím voleb Pad1 a PadN nebo adres místních linek jsou v RFC 4942 také představeny, včetně problematiky fragmentace. Problémem pro bezpečnost jsou zejména pakety s překrývajícími se fragmenty (umožní obejít pravidla firewallu). Podrobný popis problému nabízí RFC 5722 [12] a navrhuje úpravu specifikace IPv6 zákazem používání překrývajících se fragmentů. Specifikace IPv6 také povoluje použít hlavičku fragmentace v paketech, které ve skutečnosti fragmentovány nejsou. Jak zmiňuje RFC 6946 [13], tyto tzv. atomické fragmenty často vznikají reakcí hostitele na zprávu ICMPv6 „Packet Too

Big“. Útočník může tyto zprávy vyrábět uměle a proti tomuto provozu využít libovolných útoků na fragmentaci založených. Navrhuje se tedy atomické fragmenty zpracovávat nezávisle na ostatních fragmentech. RFC 6980 [14] potom přímo zakazuje použití fragmentace ve zprávách objevování sousedů (ND). Fragmenty by také pokud možno měly mít nepredikovatelné identifikátory (součást fragmentační hlavičky nutná pro znovusestavení původního paketu), opačného případu lze zneužít pro odhalení pravidel firewallu, skeny portů, útoky DoS atd. (pro zajímavost: systémy rodiny MS Windows začínají počítat od hodnoty 1).

IPv6 s sebou přináší novou implementaci Internet Control Message Protocol (ICMP), tzv. ICMPv6, jak definováno v RFC 4443 [15]. ICMPv6 je nezbytnou součástí pro správnou funkci IPv6, nekontrolován však tento protokol představuje potenciální bezpečnostní riziko. Obdobná situace platí pro protokol ICMP nad IPv4, avšak stejné strategie filtrování nelze na jeho IPv6 ekvivalent uplatnit. RFC 4890 [16] obsahuje doporučení pro filtrování ICMPv6 firewally, neboť je třeba hledat rovnováhu mezi příliš agresivní (negativní dopad na komunikaci IPv6) a příliš benevolentní (otázka bezpečnosti) filtrovací politikou. Pravidla filtrování musí také vzít v úvahu konkrétní typ zprávy ICMPv6. Například chybové zprávy ICMPv6 musí být firewallem puštěny skrz, ale pokud firewall také slouží jako směrovač, zprávy ICMPv6 sloužící pro nastavení jeho rozhraní by již neměl předávat dále. Doporučení pro filtrování RFC 4890 dále dělí dle cíle provozu ICMPv6 na pravidla pro provoz směřující na rozhraní firewallu a pravidla pro provoz tranzitní. Zprávy ICMPv6 kategorizuje do tříd:

- Zprávy, které nesmí být odhozeny.
- Zprávy, které by neměly být odhozeny.
- Zprávy, které mohou být odhozeny.
- Zprávy, které administrátoři sítě mohou chtít odhodit v závislosti na lokální bezpečnostní politice.
- Zprávy, jejichž odhození by administrátoři měli zvážit.

Zprávy jsou v rámci svých tříd filtrovány dle jejich typu a druhu zdrojové a cílové adresy. Oproti původnímu vztahu ICMPv4 vs. firewall je tedy třeba vytvořit gra-

nulárnější sadu pravidel. Typicky firewall monitorováním ICMPv6 pomáhá chránit před útoky Denial-of-Service, přesměrováním nebo přečíslováním a také skenováním sítě.

RFC 6437 [17] popisuje mimo jiné riziko tzv. skrytého kanálu, který zneužívá značek toku a zmiňuje firewall jako potenciální obranu, pokud je riziko považováno za významné (firewall sám je nicméně nucen v takovém případě vykazovat vůči značkám toku nežádoucí chování). Uveďme i dokument RFC 4487 [18], který hodnotí interakce Mobile IPv6 a firewallů – ty mohou bránit optimalizaci směrování nebo dokonce samotnému navázání komunikace.

Firewally je také třeba adaptovat pro takzvaný multihoming – protokol Shim6 (mezi jinými vyvíjenými protokoly), definovaný v RFC 5533 [19]. Ten nabízí hostiteli možnost využívat najednou několik adres IPv6. Firewall tak musí korektně udržovat stav v situaci, kdy hostitel využije ke stejnému spojení jinou adresu IP. I o tomto problému hovoří RFC 6629 [20].

1.2 Firewally a bezpečnost v prostředí IPv6

Alangar a Swaminathan [21] hovoří o problémech firewallů při zpracování dlouhých řetězců hlaviček IPv6 („Header Chain“), kdy firewall musí rozebrat mnohočetné rozšiřující hlavičky pro provedení hloubkové inspekce paketů. Kombinace rozšiřujících hlaviček a fragmentace pak může filtrování paketů úplně zabránit. Útočník může dokonce použít velké množství rozšiřujících hlaviček jako vektoru Denial-of-Service útoku.

Firewall nemusí být schopen dlouhý řetězec celý zpracovat a dovolí tak útočníkovi obejít pravidla filtrování – RFC 7112 [22] navrhuje řešení, kdy první fragment (tzn. s nulovým offsetem) paketu musí obsahovat kompletní řetězec hlaviček. Praktické testy problémů fragmentace a směrovací hlavičky lze nalézt v [23], kde se autorům takto skutečně podařilo firewall obejít. Vlastní řešení problému navrhli již dříve Lim, Kim a Kim [24] jakožto tzv. „Detour Protection Algorithm“.

Problémy s fragmentací a překrývajícími se fragmenty zůstávaly ve výchozích firewallech populárních operačních systémů i nadále, viz Atlasisovu studii z roku 2012 [25]. Tento stav Atlasis zdůvodňuje nedostatečným dodržováním doporučení

souvisících RFC (zejména RFC 5722 [12]) v implementacích firewallu. Své závěry aktualizoval v roce 2013 [26], kde konstatuje, že přes jistá zlepšení (zejména co se týče systému GNU/Linux) všechny moderní operační systémy stále akceptují velmi malé fragmenty, čehož lze za specifických podmínek zneužít pro vedení útoku. Žádný ze systémů se dle jeho zjištění stále plně neřídí doporučeními RFC.

Důležitost správné konfigurace pravidel pro filtrování fragmentovaných paketů ukazuje [27] příkladem, kdy odpovědi DNSSEC (rozšíření pro DNS) na jmenné dotazy mohou být pro svou velikost fragmentovány a na firewallu odhozeny. Pro klienty ležícími za takto konfigurovaným firewallem se pak zóna Internetu stává nedostupnou kvůli nemožnosti jmenného překladu na adresu IP.

Problémem rozšiřujících hlaviček/fragmentace je také jejich praktické použití a nasazení – jak zjistila studie [28], rozšiřující hlavičky jsou v reálném prostředí velmi často odhazovány. IETF diskutuje možné zapovězení fragmentace v IPv6 jako takové.

Lai, Jiang a Yang [29] zkoumali možnosti spolupráce firewallů hostitelů a firewallů síťových pro zabezpečení sítě IPv6. Jelikož pakety, které používají hlavičku ESP, jsou šifrovány a nelze tak provádět jejich inspekci firewally (obsah paketu je znám pouze původci a cíli, jenž vlastní tajný klíč), navrhují nasazení duálního firewallu řízeného jedním centrálním serverem. Jedná se o zajímavý přístup k řešení problematiky bezpečnosti v IPv6 aktualizací myšlenky tzv. distribuovaných firewallů [30] pro IPv6.

Autoři Hoffman a Scarfone z National Institute of Standards and Technology zpracovali komplexní manuál a praktický úvod do vývoje bezpečnostních politik a nasazování firewallů [31].

1.3 Shrnutí

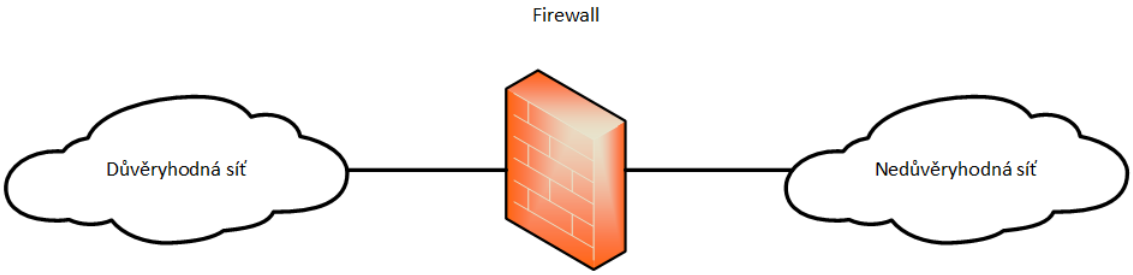
Problematika firewallů v IPv6 je značně komplexní. Nejen, že je třeba se potýkat s již vyřešenými problémy IPv4, které se v IPv6 objevují znovu (překrývání fragmentů…), IPv6 přináší i otázky nové, související s unikátními vlastnostmi tohoto protokolu, zmínili jsme např. ICMPv6, multihoming a jiné. Některé nové vlastnosti IPv6 (např. směrovací hlavičky typu 0) už navíc stačily nabýt status „deprecated“ – tedy jejich použití je třeba se, pokud možno, vyhnout. K tomu připočtěme problémy vyplývající z nutné koexistence IPv4 a IPv6 během přechodové doby adaptace IPv6

a výsledek nelze považovat za snadno uchopitelný.

RFC poskytují obsáhlý balík opatření a doporučení, jak se v tomto prostředí chovat. Jak ale zmiňuje Atlasis [26], systémy nejsou s RFC plně kompatibilní, což je předpokladem potenciálních budoucích bezpečnostních problémů. V době psaní této práce navíc není k dispozici plný oficiální seznam požadavků na firewally IPv6, pouze jeho rozpracování organizací IETF. Je patrné, že tato oblast se stále prudce vyvíjí. Jako pozitivní lze hodnotit to, že relativně pomalé přijetí IPv6 poskytuje prostor pro dostatečné vyzrátí nových technologií.

2 Obecné principy a klasifikace firewallů

Termín firewall byl vypůjčen z oblasti protipožární ochrany, kde označuje bariéru bránící šíření požáru. Podobně úkolem síťového firewallu je bránit neautorizovanému a nekontrolovanému síťovému provozu ve vstupu z jedné sítě do jiné a chránit tak citlivá data a služby. RFC 2647 [32] podává definici firewallu jakožto „zařízení nebo skupiny zařízení, jež prosazují politiku kontroly přístupu mezi sítěmi“. Pro tento úkol má definována pravidla, která určí jaký provoz bude povolen, odmítnut či zahozen (při odmítnutí narozdíl od zahození podá firewall chybovou zprávu). Typicky je nasazen na rozhraní důvěryhodné a nedůvěryhodné sítě (například na rozhraní intranet/Internet), viz Obrázek 1.



Obrázek 1: Nasazení firewallu (*Zdroj: autor*)

Je třeba podotknout, že firewall může stát na hranici sítě s mnoha hostiteli jakožto specializované zařízení, jež centralizuje některá bezpečnostní opatření, aby nebylo třeba zabezpečovat každého hostitele zvlášť – takový firewall označujeme jako *síťový* a bývá jedním z klíčových prvků, ovlivňujících návrh sítě. Může ovšem být přímo součástí samotného hostitele a zabývat se ochranou pouze této jedné stanice. Takové firewally označujeme jako *osobní* nebo *hostitelské*.

Firewall může být implementován čistě v software, nebo využívat speciální hardware (zejména při vysokých nárocích na výkon). Jako příklad čistě softwarového firewallu uveďme např. modul filtrování paketů frameworku Netfilter v jádře systému Linux. Cisco ASA [33] je naopak příkladem firewallu se speciálním hardware. Tento typ firewallu bývá proprietární povahy a také finančně náročný.

Bez ohledu na typ, firewally poskytují následující služby [34]:

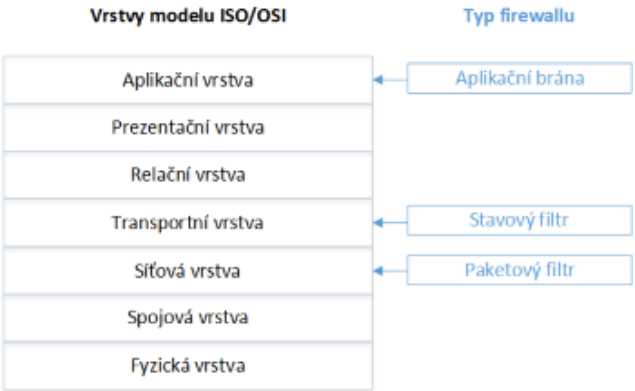
- **Předávání síťového provozu** Mnoho firewallů se chová jako směrovače, aby

spolu mohly komunikovat odlišné sítě (chování lze dosáhnout např. pomocí iptables, programu CLI, který spolupracuje s Netfilter v jádře systému Linux).

- **Vymezení sítě** Použití firewallu je způsobem, jak vytvořit hranici mezi sítěmi. Tato hranice pomáhá řídit a organizovat síťový provoz.
- **Ochrana před skenováním sítě, útoky DoS a sniffingem** Firewall monitoruje odchozí a příchozí provoz a umožňuje vybraný provoz omezovat.
- **Filtrování portů a adres IP** Jedná se o schopnost přijmout nebo odmítnout provoz na základě adresy IP a čísla portu.
- **Filtrování obsahu** Server proxy jakožto typ firewallu umožňuje inspekci např. URL. Vhodnou konfigurací lze dosáhnout blokování obsahu, který administrátor považuje za nevhodný.
- **Autentizace a šifrování** Firewall může mít schopnost ověřit uživatele a šifrovat spojení mezi sebou a firewallem v jiné síti.
- **Tvorba logů** Firewall umožňuje zpětné zkoumání detailů provozu, který jím prošel.

2.1 Firewally a model ISO/OSI

Referenční model ISO/OSI poskytuje základ pro dělení firewallů dle jednotlivých vrstev modelu, na nichž jednotlivé typy firewallů pracují, více Obrázek 2. Z pohledu této diplomové práce pro nás budou zásadní zejména transportní a síťová vrstva, resp. stavový a paketový filtr.

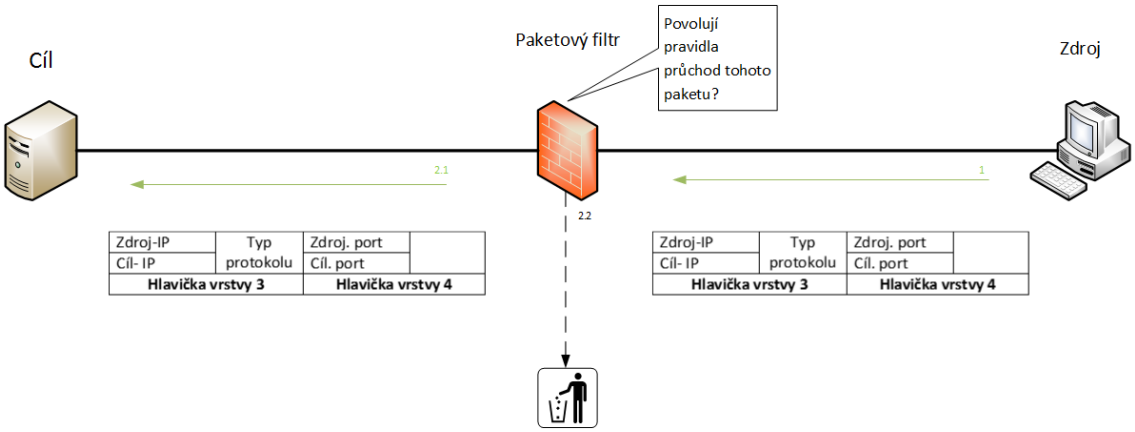


Obrázek 2: Model ISO/OSI a typy firewallů (*Zdroj: upraveno dle* [34])

2.1.1 Paketový filtr

Paketový filtr rozhoduje o dalším předání nebo odhození paketu pouze na základě adresy IP nebo čísla portu (příp. typu použitého protokolu – UDP, TCP, ICMP...). Tradičně se o tomto typu firewallu hovoří jako o zařízení třetí vrstvy modelu ISO/OSI, přestože čísla portu asociujeme s vyšší (čtvrtou) vrstvou modelu ISO/OSI. Jedná se totiž v podstatě o směrovač s určitou rozhodovací pravomocí a pracuje téměř stejnou rychlostí jako pouhý směrovač, neboť neprovádí hlubší inspekci. Musí nicméně zpracovat všechny pakety vždy jednotlivě, i u ustavených datových toků, neboť ty tento typ firewallu neumí rozlišit.

Výhodou tohoto typu firewallu je rychlost a jednoduchost. Nevýhodou je slabá nebo žádná schopnost bránit se podvrženým paketům (takové pakety jsou adresovány falešnými informacemi) nebo útokům na vyšších vrstvách. Vyžaduje také vytvoření kvalitní sady pravidel bezpečnostní politiky. Obrázek 3 zpodobňuje prostý paketový filtr.



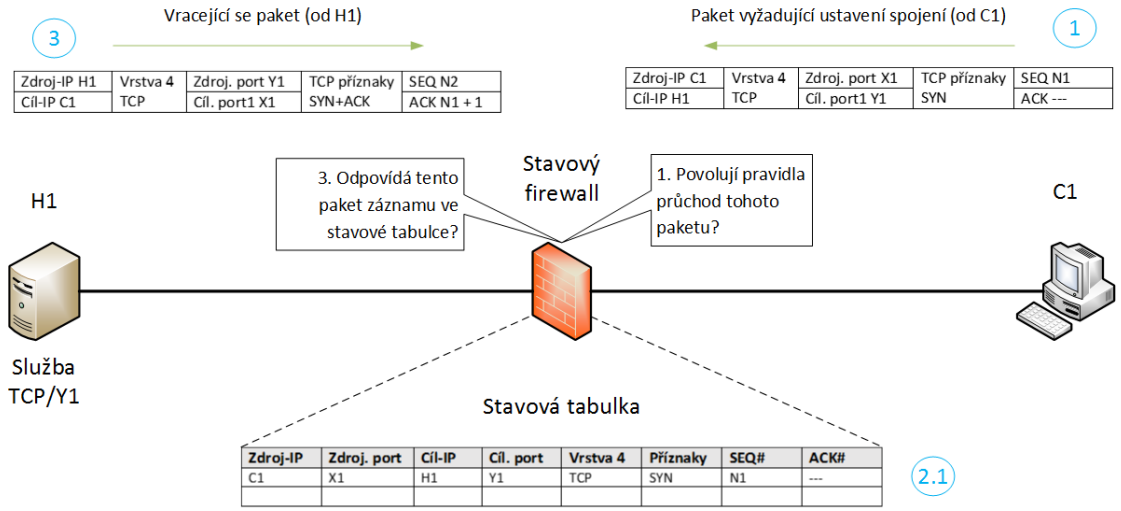
Obrázek 3: Paketový filtr (*Zdroj: upraveno dle* [35])

2.1.2 Stavový firewall

Tento typ firewallu považujeme za zařízení 4. vrstvy modelu ISO/OSI, rozšiřuje schopnosti a bezpečnost paketových filtrů. Jeho podstatou je provádění tzv. stavové inspekce paketů – firewall je schopen sledovat síťová sezení, takže když obdrží paket protokolu TCP typu ACK, může ověřit jeho legitimitu ve stavové tabulce. Ta uchovává informace pro identifikaci sezení jako např. zdrojovou a cílovou adresu, čísla

portů, příznaky, pořadová čísla paketů, typ protokolu. Záznam je do tabulky přidán (pokud to pravidla firewallu povolí), když firewall spatří první paket SYN, který otevírá sezení TCP. Tento záznam je poté referencován pro následující pakety sezení [34], a to v obou směrech. Při ukončení spojení je záznam z tabulky odstraněn. Tím je zabráněno vzniku tzv. děr. Pro nestavový protokol jako je UDP lze využít heuristiky[1] – pakety ze stejné zdrojové adresy IP cílící na stejnou adresu IP v určitém časovém úseku lze považovat za jedno sezení.

Tento typ firewallu stále nabízí vysoký výkon (obecně je rychlejší než aplikační brána, ale pomalejší než jednoduchý paketový filtr), náročnou operací je pouze ustavení sezení. Obrázek 4 schématicky zobrazuje práci stavového firewallu.



Obrázek 4: Stavový firewall (*Zdroj: upraveno dle* [35])

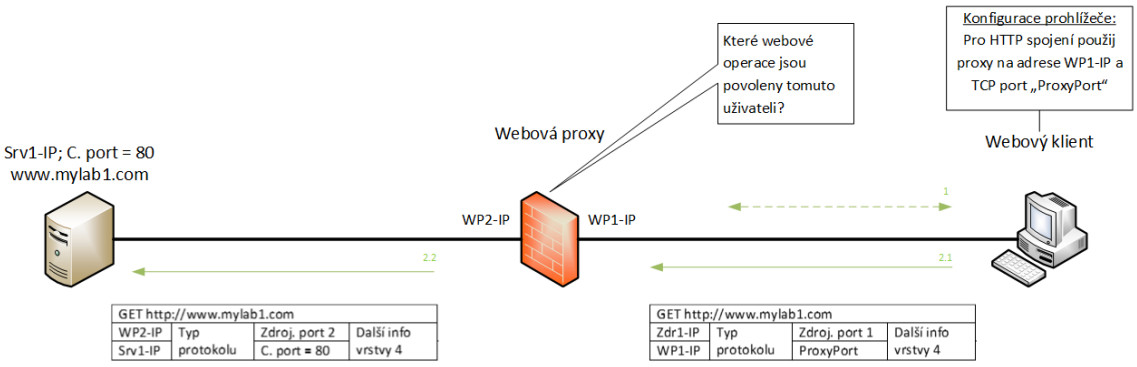
2.1.3 Aplikační brána

Také nazývána aplikační proxy. Při použití tohoto typu firewallu není možná přímá komunikace mezi sítěmi, proxy na sebe bere úlohu prostředníka, přes kterého je komunikace v obou směrech předávána. Skrývá tak detaily o síti, kterou chrání. Tento typ firewallu provádí výpočetně nejnáročnější operace z uvedených typů. Pracuje z nich také nejvýše, na 7. vrstvě modelu ISO/OSI. Díky tomu má k dispozici nejdetailnější informace o tocích dat. Toho lze dále využít mimo jiné k tvorbě detailních

[1] Technika řešení problémů poskytující „dostatečně dobré“ řešení

logů a cachování obsahu, typickými operacemi jsou ale např. filtrování nežádoucích URL nebo bezpečnostní omezení protokolu FTP. Aplikační brány se používají obvykle ve spolupráci s paketovým filtrem, např. kvůli obraně před útoky DoS.

Aplikační brány se specializují na jednu službu. Obrázek 5 představuje práci takové aplikační brány s protokolem HTTP, tzv. webové proxy.



Obrázek 5: Webová proxy (*Zdroj: upraveno dle* [35])

2.2 Hrozby

Dokument [36] poskytuje přehled hrozeb, kterým je ve vztahu k firewallům třeba věnovat pozornost. Shrnuty jsou v Tabulce 1.

Tabulka 1: Hrozby

Hrozba	**Popis**
Podvržená adresa	Vnitřní či vnější uzel použije „falešnou“ či nepřiřazenou adresu.
Opakované pokusy o autentizaci	Schopnost vnějšího uzlu opakovaně se pokoušet o přístup k vnitřním zdrojům.
Nedovolený vtok informací	Neautor. infor. pocházející z vnějšího zdroje.
Nedovolený odtok informací	Vnitřní zdroj neoprávněně poskytuje informace vnějšímu cíli.
Krádež identity	Vnější entita se maskuje jako autor. uživatel.
Selhání zaznamenávání	Vnější entita vyčerpá schopnost firewallu zaznamenávat události.
Útok „replay“	Vnější entita odposlechne a použije validní autorizační data.
Poškození uložených dat	Neautorizovaný vnější uživatel falšuje záznamy o událostech.

2.3 Firewally v síťové architektuře

Firewally nestojí v síti osamoceny a jakožto pomyslné brány do sítě značně ovlivňují podobu síťové topologie. Způsobů, jak firewally v síti nasadit, je přitom více a ten pravý je potřeba vždy volit podle konkrétních požadavků.

2.3.1 Architektura Single-box

Nejjednodušší architekturou je taková, kde je zodpovědnost firewallu přiřazena jedinému objektu na perimetru sítě. Z tohoto postavení plynou jak výhody (jediný objekt, kterému je třeba zajistit kofiguraci), tak nevýhody (bezpečnostní opatření jsou koncetrovaná na jediném místě). Takovéto řešení upřednostňuje praktičnost před bezpečností a je vhodnější pro menší sítě [37].

Jako firewall pro celou síť může posloužit pouhý paketový filtr tzv. *skrýnujícího směrovače*. Jedná se o řešení upřednostňující cenové hledisko, neboť směrovač je tak jako tak vyžadován pro připojení k Internetu. Z povahy paketového filtru však může být problém provozovat některé služby. Lze také použít tzv. *dual-homed hostitele*, což je počítač s alespoň dvěma síťovými rozhraními. Hostitel funguje jako prostředník, přes nějž systémy z různých sítí (vnitřní/vnější) komunikují. Toto odpovídá principu aplikační brány a platí tedy, že získáváme vysokou úroveň kontroly nad provozem za cenu výkonu. Obě zařízení lze spojit v tzv. *multi-purpose box* [37].

2.3.2 Architektura Screened Host

Narozdíl od použití dual-homed hostitele, připojeného k vnitřní a vnější síti, architektura Screened Host poskytuje služby z hostitele, který je součástí pouze vnitřní sítě. Tomuto hostiteli říkáme opevněný [37]. Na rozhraní vnitřní a vnější sítě přitom stojí směrovač, který poskytuje základní filtrování paketů. Vnějším systémům je povoleno otevírat spojení pouze k opevněnému hostiteli. Systémy z vnitřní sítě mohou otevřít spojení směrem ven pouze s pomocí opevněného hostitele jakožto prostředníka. Zvláštní postavení opevněného hostitele je třeba reflektovat v úvahách o bezpečnosti – provozovat na něm jen nezbytné služby a autorizovat jej jen k nezbytným úkonům, neboť se nutně stává úzkým hrdlem bezpečnosti. Při zkompromi-

tování opevněného hostitele je zkompromitována celá síť. Obecně nicméně takováto architektura poskytuje vyšší úroveň zabezpečení oproti architektuře Single-box [37].

2.3.3 Architektura Screened Subnet

Tato architektura přidává další bezpečnostní vrstvu k architektuře Screened Host vytvořením podsítě, jakéhosi nárazníku, která dále odděluje vnější a vnitřní sítě. Této podsíti se říká demilitarizovaná zóna (DMZ) nebo perimetr. Tato architektura odstraní problém jediného bodu selhání architektury Screened Host izolací opevněného hostitele (hostitelů) v této podsíti [37]. Při jeho úspěšném napadení z vnější sítě není kompromitována vnitřní síť. Pro úspěšné napadení vnitřní sítě by útočník musel úspěšně obejít kromě zařízení na hranici vnější síť/perimetr ještě další zařízení, stojící mezi perimetrem a vnitřní sítí [37].

2.3.4 Architektura Split-Screened Subnet

Architektura vychází z architektury Screened Subnet, perimetr je však dále rozdělen dual-homed hostitelem pro jemnější kontrolu nad datovými toky než pouhé filtrování paketů [37]. Lze tak např. zajistit administrativní přístup (administrativní provoz je separován, což kromě bezpečnosti může zlepšit i výkon) ke strojům, které provozují služby dostupné z Internetu nebo pouze pro docílení kvalitní vícevrstvé ochrany.

2.3.5 Architektura Independent Screened Subnets

Opět vychází z architektury Screened Subnet, přidáním dalších směrovačů na hranicích sítě vytvoříme více nezávislých perimetrů a dosáhneme redundance nebo oddělíme odchozí služby (jež umožňují uživatelům připojit se k Internetu, jako např. webová proxy) a příchozí spojení/služby (např. z Internetu přístupný webový server) [37]. Takovým oddělením spojení lze dosáhnout silného zabezpečení.

2.3.6 Host-Based Firewall

Zatím zmíněné architektury chrání v místě své činnosti (na perimetru sítě) vnitřní síť jako celek. Často musí zvládat velké množství protokolů a služeb, což znamená komplexní konfiguraci. Komplexní konfigurace logicky vytváří velký prostor pro chybu. Obrana perimetru navíc z podstaty nemůže chránit před hrozbami uvnitř sítě (např. samovolně se šířící škodlivý software). Je možné přidat další linii obrany (stejně jako v již představených architekturách může být třeba vytvořit kvalitní sadu pravidel) provozováním firewallu přímo na zvolených hostitelích (*Host-Based Firewall*), typicky použitím specializovaného softwarového balíčku. Tyto balíčky často nabízí i další funkce kromě filtrování odchozích/příchozích spojení jako např. blokování spojení ve vztahu ke specifické aplikaci běžící na hostiteli [38]. Je možné, aby instance firewallu na jednotlivých hostitelích byly spravovány z centrálního serveru, pak hovoříme o *Agent-Based* firewallech. Mnohé operační systémy jsou distribuovány s vlastní implementací firewallu.

3 Firewally a protokol IPv6

Tato kapitola se zabývá dopady konkrétních vlastností protokolu IPv6 na firewally. Cílem je tyto dopady vyjmenovat a teoreticky popsat. Tabulka 2 poskytuje stručný přehled hlavních rozdílů mezi protokoly IPv4 a IPv6. Plnou originální specifikaci IPv4 lze nalézt v RFC 791 [39]. Specifikaci IPv6 lze nahlédnout v RFC 2460 [9].

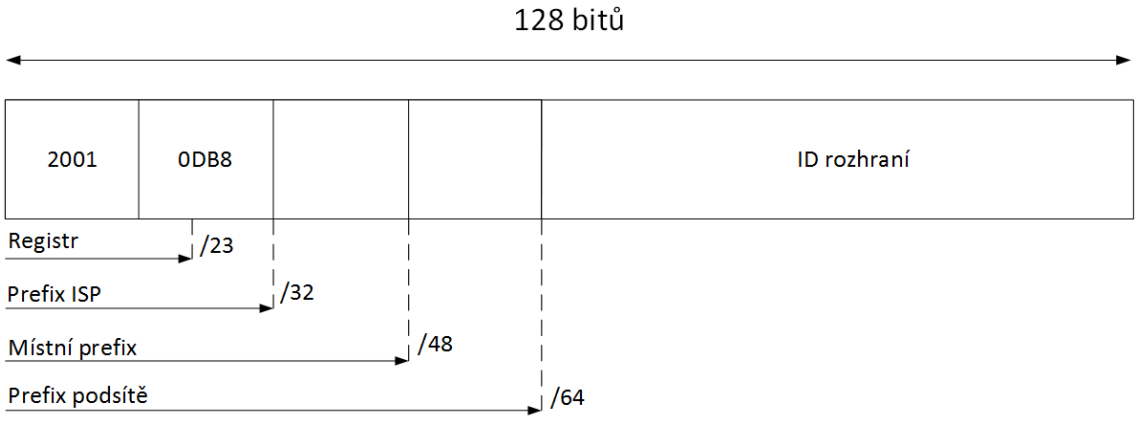
Tabulka 2: Srovnání IPv4 a IPv6

Vlastnost	**IPv4**	**IPv6**
Délka adresy	32 bitů	128 bitů
Nezbytnost NAT	Ano	Nikoliv
Nastavení adresy IP	Staticky, DHCP	Staticky, DHCPv6, Auto
Řešení adres vrstvy 2	ARP	ICMPv6 Neighbor Discovery
Počet adres/rozhraní	Jedna	Neomezeně
Podsítě	Proměnná délka	16+ bitů
Všesměrové vysílání	Ano	Nikoliv
Multicast	Malá role	Zásadní role
MTU	576 bajtů	1280 bajtů
Fragmentace	Směrovače, hostitelé	Hostitelé
ICMP skrze firewall	Volitelně	Nezbytně
Složitost hlaviček	Proměnná	Konstatní/proměnná (při řetězení)
Formální mechanismus rozšiřujících hlaviček	Nikoliv	Ano
IPSec	Volitelně	Volitelně (pův. povinně)

3.1 Filtrování na základě adresy IPv6

Nestavové filtrování paketů na základě zdrojové nebo cílové adresy funguje v IPv6 na stejném principu jako v IPv4, přináší však vyšší komplexnost, neboť jedno síťové rozhraní může mít více adres IPv6. Na to je vždy třeba myslet při konfiguraci, neboť špatné nastavení paketového filtru by mohlo vést k jeho nežádoucímu obe-

jití pomocí rozhraní, které nemá nastavena patřičná pravidla. Problém také nastává při změně poskytovatele připojení (ISP), neboť v prvních 64 bitech adresy IPv6, tzv. síťovém prefixu, dojde ke změně (pro příklad způsobu alokace prefixů v IPv6 viz Obrázek 6). Hostitelé mohou mít naráz novou i starou, „deprecated“ adresu po určitou stanovenou dobu (během této přechodné doby může být síť obsluhována dvěma poskytovateli, jde o tzv. *multihoming*), během níž jsou staré adresy vyřazovány [40]. Administrátor paketového filtru musí na tuto situaci patřičně reagovat úpravou pravidel filtrování.



Obrázek 6: Alokace prefixů v IPv6 (*Zdroj: upraveno dle* [41])

Firewall také nesmí předávat pakety s nelegální zdrojovou nebo cílovou adresu mimo perimetr, tyto případy shrnuje Tabulka 3 [42]. Dodejme, že ačkoliv budeme blokovat předávání adres lokální linky na perimetru, např. hostitel chráněný firewallem uvnitř LAN musí rozsah `fe80::/10` povolit pokud chce využít decentralizovaného mechanismu bezestavové konfigurace (SLAAC), jak definován v RFC 4862 [43]. Ze stejného důvodu (a také kvůli protokolu Neighbor Discovery) též musí povolit multicastový rozsah `ff00::/8`.

3.2 Multihoming

Jako multihoming označujeme stav, kdy je síť připojena k více poskytovalům připojení a je žádoucí a typický zejména v případě, kdy je požadována vysoká spolehlivost připojení (připojení jsou redundantní). Multihoming může být pouze dočasný v případě, kdy dochází ke změně poskytovatele, viz výše. Nejedná se o vlastnost, která by nebyla obsažena již v IPv4, v IPv6 má však nové otazníky.

Tabulka 3: Adresy IPv6 k blokaci na perimetru

Pakety k zablokování	**Adresy**
Nespecifikovaná adresa	`::`
Adresa zpětné smyčky	`::1`
IPv4-kompatibilní adresa	`::/96`
IPv4-mapovaná adresa	`::ffff:0.0.0.0/96` `::/8`
Automaticky tunelované pakety s kompatibil. adresami	`::0.0.0.0/96`

Další kompatibilní adresy	`::224.0.0.0/100`
	`::127.0.0.0/104`
	`::0.0.0.0/104`
	`::255.0.0.0/104`

Falešné pakety 6to4	`2002:e000::/20`
	`2002:7f00::/24`
	`2002:0000::/24`
	`2002:ff00::/24`
	`2002:0a00::/24`
	`2002:ac10::/28`
	`2002:c0a8::/32`

Adresy lokální linky (legitimní pouze v LAN)	`fe80::/10`
Lokální místní adresy	`fec0::/10`
Lokálně-unikátní pakety	`fc00::/7`
Multicastové adresy jako zdrojové adresy	`ff00::/8`
Dokumentační adresy	`2001:db8::/32`
Adresy 6Bone	`3ffe::/16`

Navrženým řešením multihomingu pro IPv6 je protokol Shim6, jak definován v RFC 5533 [19]. Ten je schopen komunikaci odklonit do patřičných cest pro využití multihomingu, to však vytváří problematický vztah s firewally. Stavový firewall vytváří pro každé spojení stav. Jelikož Shim6 mění tzv. lokátory, jenž komunikující hostitelé používají pro spojení a které neadaptovaný stavový filtr používá pro identifikaci konkrétního toku, je potřeba firewall adaptovat tak, aby na změnu dokázal reagovat. Administrátor nestavového paketového filtru musí vzít v úvahu nové typy paketů, které Shim6 generuje. Podrobně se touto problematikou zabývá práce [44].

Pro funkci Shim6 jej musí podporovat hostitelé na obou koncích komunikace. Mnoho firewallů navíc pakety s rozšiřující hlavičku Shim6 odhazuje. To mohou být překážky v jeho praktickém nasazení.

LinShim6 implementuje podporu Shim6 pro jádro Linux [45]. Poslední vydaná verze však pochází z roku 2009 [46]. Zdá se, že pozornost byla přesunuta k protokolu Multipath TCP [47], jenž si bere za cíl pracovat jak s IPv6, tak IPv4. Multipath TCP má své vlastní problémy ve vztahu k firewallům (použití voleb TCP, které jsou na firewallech často filtrovány). Související RFC 6824 [48] je vedena jako experimentální a vývoj Multipath TCP stále probíhá.

Jiný přístup nabízí RFC 7157 [49], doporučuje řešení na bázi DHCPv6 s tím, že přechodně bude třeba použít tzv. IPv6-to-IPv6 Network Prefix Translation (NPTv6), tedy nestavovou IPv6 obdobu NAT pro překlad prefixů IPv6. Jedním z cílů návrhu IPv6 přitom bylo vyhnout se používání překladu adres.

Je patrné, že tato oblast stále prochází vývojem a jak multihoming ovlivní oblast firewallů je prozatím otevřenou otázkou.

3.3 Internet Control Message Protocol version 6

„ICMPv6 je uzly IPv6 používán k hlášení chyb, vzniklých při zpracovávání paketů, a zvládání dalších funkcí vrstvy Internet[1], jako je diagnostika (ping ICMPv6). ICMPv6 je integrální částí IPv6 a základní protokol (veškeré zprávy a chování vyžadované specifikací) MUSÍ být plně implementován každým uzlem IPv6.“ [15]

Internet Control Message Protocol version 6 (ICMPv6) je nástupcem protokolu

[1] Odpovídá 3. vrstvě modelu ISO/OSI

ICMPv4 pro prostředí IPv6. S pomocí protokolu Neighbor Discovery, jenž definuje pět zpráv ICMPv6 [50], nahrazuje také protokol ARP, jímž se v IPv4 získává adresa 2. vrstvy modelu ISO/OSI uzlu z jeho adresy 3. vrstvy. Dále ICMPv6 přebírá úkol protokolu Internet Group Management Protocol (IGMP), který se používá pro registraci ve skupinách multicast.

ICMPv6 používá pro svou práci zpráv, jež můžeme rozdělit do dvou skupin – na chybové a informační. Typ zprávy je určen hodnotou v poli Typ hlavičky zprávy ICMPv6. Pole Typ je 8 bitů dlouhé, nejvyšší bit pro chybové zprávy je vždy nastaven na hodnotu 0 (rozsah možných hodnot 0–127), pro zprávy informační naopak vždy na hodnotu 1 (rozsah 128–255). Na správné funkci ICMPv6 také závisí protokol Path MTU Discovery for IP version 6 [51]. Tabulka 4 představí některé exponované zprávy protokolu.

Tabulka 4: Vybrané zprávy ICMPv6

Typ	**Význam**
1	Destination Unreachable
2	Packet Too Big
3	Time Exceeded
4	Parameter Problem
100	Parameter Problem
127	Reserved For Expansion of ICMPv6 Error Messages
128	Echo Request
129	Echo Reply
133	Router Solicitation
134	Router Advertisement
135	Neighbor Solicitation
136	Neighbor Advertisement
137	Redirect
255	Reserved For Expansion of ICMPv6 Informational Messages

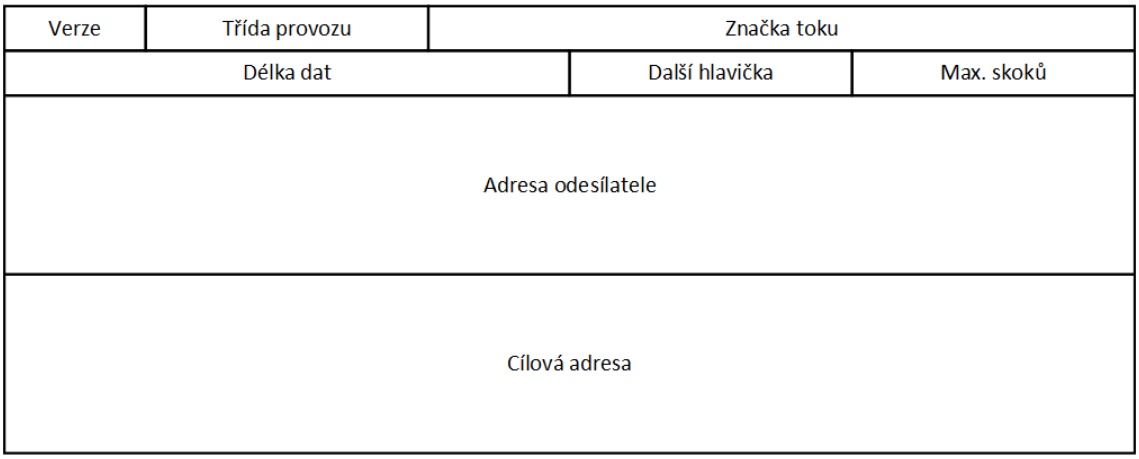
Připomeňme, že filtrování ICMPv6 vyžaduje granulárnější sadu pravidel firewallu a zprávy z pohledu firewallu dělíme dle RFC 4890 [16] na:

- Zprávy, které nesmí být odhozeny.
- Zprávy, které by neměly být odhozeny.
- Zprávy, které mohou být odhozeny.
- Zprávy, které administrátoři sítě mohou chtít odhodit v závislosti na lokální bezpečnostní politice.
- Zprávy, jejichž odhození by administrátoři měli zvážit.

Bezpečnostní opatření v síti musí vždy dovolit úspěšné provozování Neighbor Discovery (s výjimkou demilitarizované zóny, kde lze spoléhat na statické záznamy v cache sousedů) a umožnit průchod zpráv „Destination Unreachable“ a „Parameter Problem“. To platí i pro Path MTU Discovery, jmenovitě zprávu „Packet Too Big“. Některý síťový provoz (např. Router Solicitation a Router Advertisement) je užitečný pouze v lokální síti a jeho vstup do sítě zvnějšku by měl být firewallem zakázán. Experimentální a nedefinované zprávy ICMPv6 by měly být odhozeny [52].

3.4 Rozšiřující hlavičky IPv6

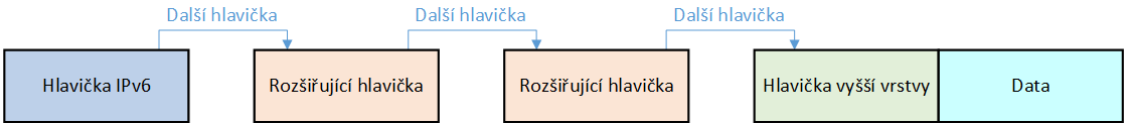
Paket IPv6 se skládá ze dvou hlavních částí – hlavičky a nesených dat. Prvních 40 bajtů paketu patří právě hlavičce, jejíž strukturu znázorňuje Obrázek 7. Mezi hlavičku a data pak mohou být vloženy a řetězeny rozšiřující hlavičky, princip znázorňuje Obrázek 8.



Obrázek 7: Hlavička IPv6 (*Zdroj: upraveno dle* [9])

Každý firewall IPv6 musí být schopen zpracovat řetězec rozšiřujících hlaviček IPv6, neboť jejich funkce může být nežádoucí z pohledu bezpečnostní politiky dané or-

ganizace. Minimem je schopnost rozšiřující hlavičky přeskočit a přejít k hlavičce vyšší vrstvy. Firewall by měl být schopen nastavit a aplikovat pravidla díky pouhé přítomnosti některé z rozšiřujících hlaviček v paketu.



Obrázek 8: Řetězení rozšiřujících hlaviček IPv6 (*Zdroj: upraveno dle* [53])

V Tabulce 5 lze nalézt rozšiřující hlavičky, jak definovány v RFC 2460 [9], a hlavičku mobility [54]. Udány jsou sestupně v pořadí, v jakém RFC 2460 doporučuje jejich použití. Jediným pevným požadavkem podle RFC 2460 je, že pokud je přítomna hlavička voleb Hop-by-hop, musí být na prvním místě v řetězci.

Dle RFC 5095 [10] je třeba se vyhnout použití hlavičky Směrování typu 0 (lze ji zneužít např. pro útok DoS mezi dvěma uzly, obejití určitých konfigurací firewallu, vyčerpání šířky pásma, odhalování topologie sítě). Firewall by s touto hlavičkou měl zacházet jako s neznámou hlavičkou, na vstupu se doporučuje tyto hlavičky filtrovat.

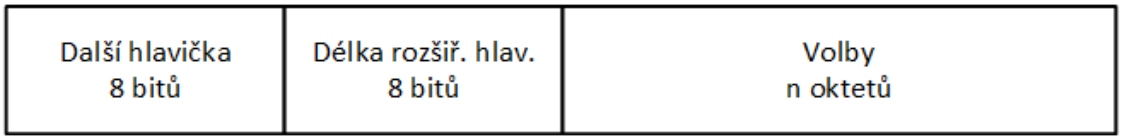
Tabulka 5: Rozšiřující hlavičky IPv6

Hlavička	**Typ**	**Význam**
Volby Hop-by-hop	0	Informace pro všechny uzly na cestě paketu
Volby pro cíl	60	Informace určené příjemci paketu
Směrování	43	Předepisuje uzly, které má paket navštívit
Fragmentace	44	Informace pro znovusložení paketu po fragmentaci
Autentizace (AH)	51	Ochrana integrity IPSec
Šifrování obsahu (ESP)	50	Ochrana integrity a důvěrnosti IPSec
Volby pro cíl	60	Informace určené příjemci paketu
Mobilita	135	Pro potřeby Mobile IPv6

Hodnota 59 v poli Další hlavička znamená, že za hlavičkou už nic nenásleduje. Hlavička Volby pro cíl se může objevit dvakrát (jednou před hlavičkou Směrování a jednou před hlavičkou vyšší vrstvy), všechny ostatní pouze jednou. Zda tomu tak je (tj. paket je validní) by firewall měl být schopen ověřit.

3.4.1 Hlavičky Volby pro cíl, Volby Hop-by-hop

Pravidla firewallu se nemusí vztahovat pouze na celé rozšiřující hlavičky, lze předávat nebo blokovat pakety i na základě specifických voleb uvnitř hlaviček Volby pro cíl a Volby Hop-by-hop. Struktura hlaviček je představena na Obrázku 9, samotné volby uvnitř jsou strukturovány v podobě typ-délka-hodnota a je tak možné specifikovat pravidla i pro volby neznámé [5]. Pakety s neznámými volbami mohou představovat bezpečnostní riziko a administrátor je typicky bude chtít na firewallu zahazovat, musí dbát ovšem na to, aby nezahazoval také např. užitečné, nově standardizované volby.



Obrázek 9: Hlavičky Volby pro cíl a Volby Hop-by-hop (*Zdroj: upraveno dle* [9])

Firewall by měl být schopen aplikovat pravidla alespoň vzhledem k následujícím volbám ve zmiňovaných hlavičkách:

- „Jumbo Payload“
- „Tunnel Encapsulation Limit“
- „Router Alert“
- Domácí adresa[1]

Tyto volby mohou mít specifické požadavky na zarovnání, pro tento účel jsou k dispozici volby Pad1 (vložení 1 oktetu) a PadN (vložení 2 a více oktetů). Jejich bity musí být vždy nastaveny pouze na hodnotu 0 a je úkolem firewallu tuto skutečnost ověřit, neboť je v opačném případě může útočník zneužít pro vytvoření tzv. skrytého kanálu. Skrytým kanálem rozumíme takový kanál, který vůbec nebyl určen pro přenos informací a není tak pod kontrolou správce. Pad1 a PadN by také neměly následovat ihned po sobě a vkládat více než 5 oktetů.

„Jumbo Payload“ [55] dovoluje použít IPv6 pakety větší než 65 535 bajtů, je ovšem

[1] Pro Mobile IPv6.

relevantní pouze ve speciálních, vysoce výkonných systémech. Běžnou praxí by mělo být odhození paketů s touto volbou firewallem.

Samotný návrh IPv6 nijak nelimituje maximální možný počet voleb v jednom paketu a nevylučuje ani jejich opakování. Zjevně tak vzniká potenciální vektor útoku DoS. Všechny volby kromě Pad1 a PadN by se měly v paketu vyskytnout pouze jednou, jinak je žádoucí, aby firewall pakety s opakujícími se volbami odhazoval.

3.4.2 Hlavičky Autentizace a Šifrování obsahu

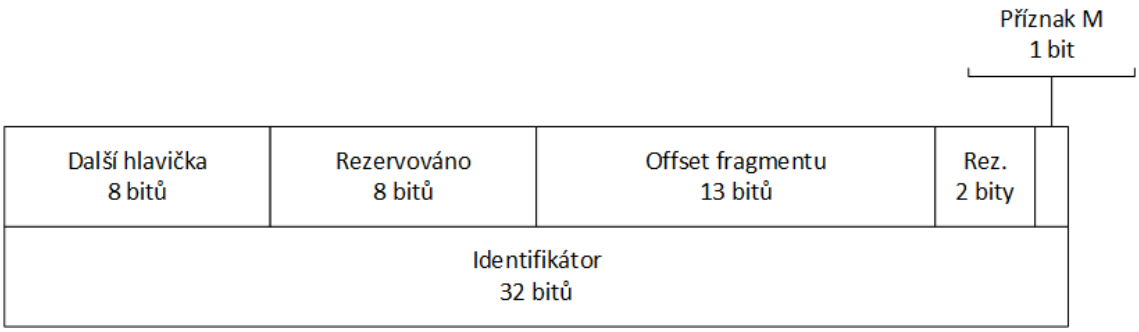
Hlavičky Autentizace a Šifrování obsahu jsou součástí frameworku IPSec [56]. Strategie IPSec pro zabezpečení komunikace je odlišná od jejího filtrování firewallem. Použitím kryptografických mechanismů se snaží dosáhnout bezpečné identifikace odesílatele a zlepšit bezpečnost sítě omezováním bezpečnostních slabin hostitelů uvnitř sítě. Odlišnost těchto strategií přináší určité problémy při jejich interakci. Například pokud paket obsahuje hlavičku Autentizace (AH), jejíž primární funkcí je poskytovat autentizaci zdroje komunikace a integritu zpráv, může firewall rozhodnutí o odhození paketu ponechat až koncovému uzlu. Větší překážkou je hlavička Šifrování obsahu (ESP), která poskytuje autentizaci, integritu a důvěrnost paketu, jejž zašifruje. Tím však znemožní jeho inspekci firewallem. Je na zvážení administrátora, jak v takovém případě postupovat. Při použití IPSec VPN připadá v úvahu povolit na firewallu hlavičky AH a ESP pouze směrem z a na vybrané adresy vnitřní sítě – povolené koncové body VPN [31]. Jinou možností je, aby firewall byl schopen výměny šifrovacích klíčů a pakety pro inspekci vždy dešifroval. Lze také nešifrovat celý paket a nechat pro firewall potřebné hlavičky čitelné. V obou případech je však narušena end-to-end povaha IPSec.

Nelze zatím předpovědět, jak masového nasazení se IPSec v prostředí IPv6 dočká, návrhem IPv6 je však velmi podporován.

3.4.3 Hlavička Fragmentace

Proces fragmentace paketů v IPv6 na rozdíl od IPv4 probíhá pouze na hostitelích. Ty pakety fragmentují tak, aby bylo možné dodržet MTU trasy (minimum bylo sta-

noveno na 1280 bajtů). K rozložení paketu na menší používá hlavičku Fragmentace, zpodobněnou na Obrázku 10. Cílový uzel tyto fragmenty znovu sestaví.



Obrázek 10: (*Zdroj: upraveno dle* [9])

Fragmentace představuje pro firewall několik problémů:

- **Získávání informací z hlaviček paketů** Aby firewall mohl plnit svoji funkci, potřebuje mimo jiné informace obsažené v hlavičkách paketů. Jedná se např. o hlavičku TCP a informace o portu, které budou obsaženy pouze v jednom z mnoha fragmentů. Pakety bez této informace by firewall typicky zahodil, zde však budou potřeba pro rekonstrukci původního paketu. Pokud by nějaký fragment chyběl, musel by být rekonstruovaný paket po vypršení času přiděleného na rekonstrukci odhozen. Jednotlivé části potřebné informace se teoreticky mohou dokonce nacházet ve více fragmentech [57], více níže. RFC 7112 [22] takové fragmentační praktiky zapovídá, nicméně jak již bylo zmíněno, různé systémy se RFC ne vždy plně řídí a útočník je může vytvářet záměrně.

- **Překrývání fragmentů** Pro rekonstrukci fragmentovaného paketu se používá délky fragmentu a pole hlavičky Fragmentace offset fragmentu. Vždy musí platit, že součet offsetu fragmentu **n** a délky fragmentu **n** je roven offsetu fragmentu **n+1** minus 1 bajt. Hodnota offsetu určuje, kam bude fragment umístěn v zrekonstruovaném paketu a nikdy by neměl existovat žádný překryv – pokud ovšem nebyl vytvořen záměrně útočníkem. Při vedení útoku s pomocí překrývajících se fragmentů se tyto jeví firewallu jinak, než jak bude vypadat znovusložený paket na hostiteli (pokud firewall posuzuje pouze první fragment pro uplatnění definované bezpečnostní politiky). První fragment se např. může tvářit jako paket TCP portu 80 (http) a projít firewallem. Následující fragment

pak přepíše port na 23 (Telnet), který by byl za normálních okolností na firewallu odhozen [57]. RFC 5722 [12] doporučuje tiché odhození překrývajících se fragmentů.

- **Drobné fragmenty** RFC nedefinují, jak nakládat s pakety (fragmenty) menšími než 1280 bajtů (tedy menšími než definované minimální MTU). Drobnými fragmenty rozumíme takové fragmenty, které přenáší méně než 1200 bajtů dat a nestojí v sérii fragmentů poslední. Mnohé firewally identifikují jednotlivé toky pomocí zdrojové a cílové adresy IP, zdrojového a cílového portu a protokolu vyšší vrstvy. Tato informace by se měla nacházet v prvním fragmentu (čili s offsetem fragmentu rovným 0 a příznakem M rovným 1), lze ale záměrně vytvářet malé pakety, jejichž cílem je obejít firewall odsunutím identifikátorů z prvního fragmentu. Odsunutí z prvního fragmentu lze také docílit tvorbou dlouhých řetězců rozšiřujících hlaviček. Drobné fragmenty by měly být v tichosti odhozeny.

- **Atomické fragmenty** Jsou to pakety s offsetem fragmentu a příznakem M oběma rovnými 0. Vznikají reakcí hostitele na zprávu ICMPv6 „Packet Too Big“ a přestože vlastně nejsou fragmentovány na více částí, obsahují hlavičku Fragmentace (specifikace IPv6 toto dovoluje). Takový provoz pak útočník může napadnout útokem založeným na fragmentaci.

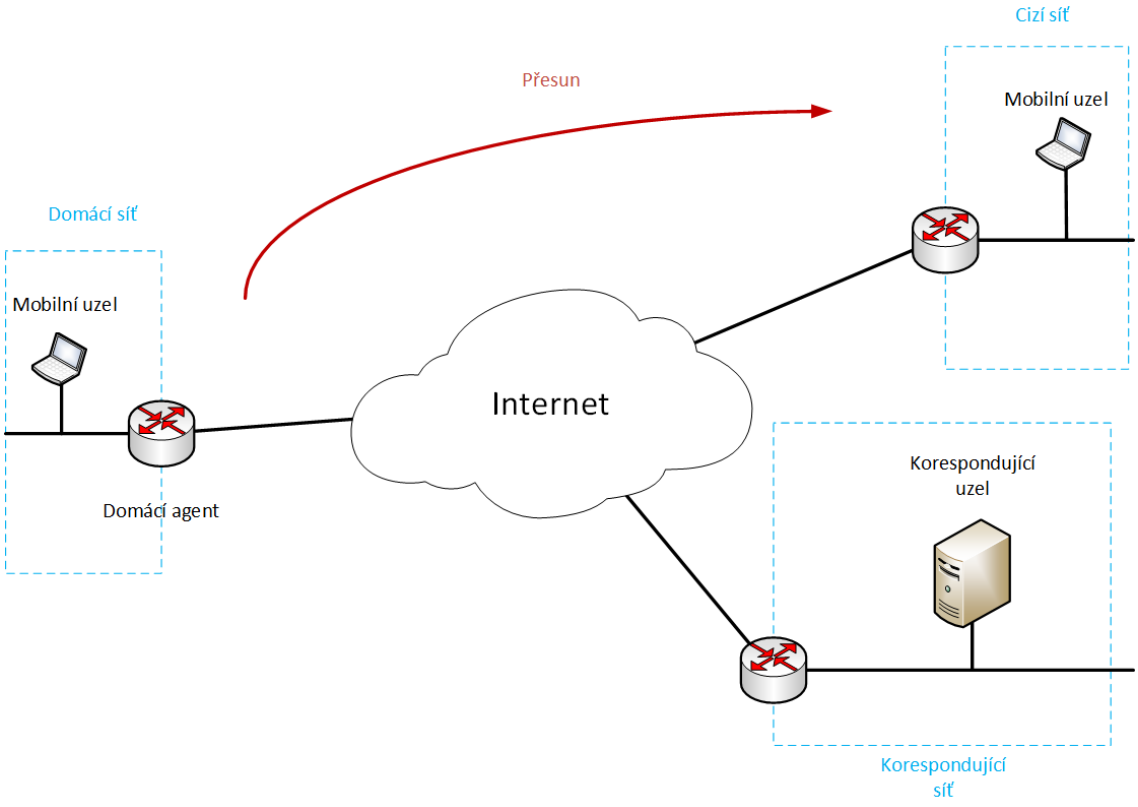
3.5 Mobile IPv6

Síťový provoz Mobile IPv6 probíhá mezi třemi sítěmi, těmi jsou:

- Domácí síť (Home Network, HN)
- Cizí síť (Foreign Network, FN)
- Korespondující síť (Corresponding Network, CN)

Když je mobilní uzel (Mobile Node, MN) v domácí síti, síťový provoz mezi domácí a korespondující sítí (kde leží např. webový server, s nímž si uzel přeje komunikovat) proudí běžným způsobem. Stručné schéma lze nalézt na Obrázku 11. Mobilní uzel má svoji domácí adresu (Home Address), která se nemění a je zaznamenána v DNS. Pokud ze sítě vycestuje, získává další, dočasné adresy (Care-of Address,

CoA). Všechny pakety adresované domácí adrese jsou transparentně směrovány dočasné adrese. Směrovač, který toto směrování provádí, nazýváme domácí agent (Home Agent). Pokud se mobilní uzel přesune na jiné stanoviště, nejprve se připojí k cizí síti. Poté zavolá domácí síti a s korespondujícím uzlem komunikuje skrze ni. Nakonec komunikuje s korespondující sítí (resp. uzlem) přímo. V každé z těchto sítí předpokládáme nasazení firewallu. Mobile IPv6 také přináší koncept optimalizace směru (Route Optimization), což je pro funkci protokolu nezbytná sada rozšíření, která umožňuje optimalizované směrování paketů mezi mobilním a korespondujícím uzlem.



Obrázek 11: Schéma komunikace Mobile IPv6 (*Zdroj: upraveno dle* [58])

Obecně jsou v Mobile IPv6 pro stavové firewally problémem zprávy, které souvisí s prací protokolu, ale jsou „nevyžádané“ – nejsou součástí stavu, který firewall vytvořil a udržuje na základě paketů odchozích zevnitř sítě, jíž firewall chrání (existuje mnoho potenciálních řešení tohoto problému, podrobný přehled nabízí práce [59]). Příkladem takové situace může být přesun mobilního uzlu do další přístupové sítě chráněné jiným firewallem – ten bude odhazovat všechny mobilnímu uzlu příchozí pakety, neboť pro ně nemá vytvořen odpovídající stav. Dále firewall nemusí rozumět

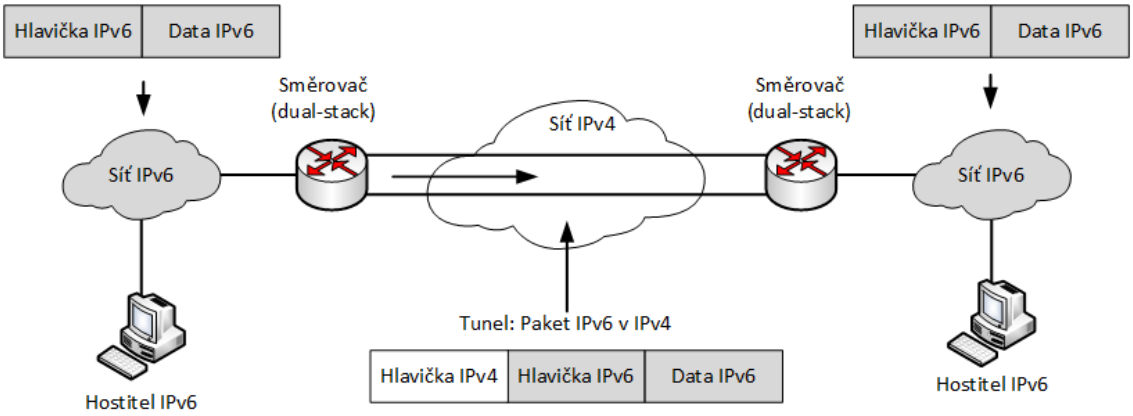
zprávám zapouzdřeným v Mobile IPv6 a zahazovat je, nemusí také rozumět hlavičce mobility (Mobility Header) Mobile IPv6. Problémem může být i odhazování síťového provozu IPSec ESP, který se používá pro registrační zprávy mezi mobilním uzlem a domácím agentem. Není pak možné provádět optimalizaci směrů [18].

Aktuálně je podpora mobility IPv6 specifikována v RFC 6275 [11], opravdové praktické nasazení je hudbou budoucnousti.

3.6 Tunely

Jelikož IPv6 není zpětně kompatibilní s IPv4, je v sítích třeba provozovat přechodové mechanismy, jakými jsou např. tunelování a konfigurace dual-stack. Dual-stack znamená, že hostitel je schopen komunikovat s pomocí obou protokolů. Je třeba zajistit, aby útočník nemohl získat výhodu volbou IPv4, nebo IPv6 – firewall musí zajistit konstantní úroveň zabezpečení pro obě verze protokolu IP.

Tunelováním obecně míníme zapouzdření jednoho protokolu (*tunelovaný protokol*) v jiném (*tunelujícím*) a bude v přechodovém období nejprve spojovat tzv. ostrovy IPv6 skrze infrastrukturu IPv4 (tato situace se bude s postupným šířením IPv6 obracet). Jakožto příklad jednoho možného způsobu tunelování znázorněme zapouzdřování IPv6 v IPv4 na Obrázku 12.



Obrázek 12: Tunelování 6in4 (*Zdroj: upraveno dle* [60])

Tunely lze rozdělit na manuálně konfigurované, kdy administrátor pevně zvolí koncové body tunelu, a automatické, kdy uzly dynamicky nakonfigurují tunel samy – automatické tunelování tak nicméně není v plné moci administrátora. Omezením automatických tunelů je, že umožňují komunikaci pouze mezi hostiteli IPv6, kteří

oba automatické tunelování podporují, a neposkytují opatření pro komunikaci s nativním Internetem IPv6 [6]. Tím je velmi omezeno jejich praktické použití, některé z těchto mechanismů navíc mají slabou podporu v operačních systémech.

Obecným problémem tunelů je, že při povolení provozu protokolu 41 (typicky používaný tunelovacími mechanismy) zevnitř sítě do sítí vnějších by firewall musel být schopen inspekce zapouzdřených paketů nebo administrátor zakázat patřičné protokoly a porty – jinak mohou hostitelé uvnitř sítě tunelovat směrem ven libovolný provoz [61]. Bezpečnostním rizikem je také podvržení adresy IPv4 jednoho konce tunelu a odesílání takovýchto paketů na druhý konec tunelu (automatické tunely musí přijímat pakety z libovolného zdroje, případný útočník tak nemusí ani podvrhovat adresu IPv4). Firewally by také měly být konfigurovatelné tak, aby odhodily jakýkoliv paket, jenž se z tunelu vynoří s hodnotou 255 v poli Max. skoků hlavičky IPv6. Tato hodnota zde není dle specifikace legální a lze ji zneužít pro vedení některých útoků [57].

Manuálně konfigurované tunely jednoduše zapouzdří paket IPv6 přidáním hlavičky IPv4 a posílají je přes explicitně definované spoje. V poli Protokol hlavičky IPv4 mají tyto pakety zmíněnou hodnotu 41. Manuálně konfigurované tunely se považují za nejbezpečnější vzhledem k vysoké úrovni administrátorovy kontroly.

Automatické tunelování může probíhat pomocí následujících mechanismů:

- **6to4** Pakety používající mechanismus lze snadno identifikovat dle jejich typického prefixu `2002::/16`. Nasazení spočívá v tom, že síť IPv4 vezme jednu ze svých adres IPv4, hexadecimálně ji převede a vytvoří tak `/48` síť IPv6. Filtrování lze provést pouhou blokací protokolu 41, nebo pokud firewall dokáže provádět hlubokou inspekci paketů IPv4 lze také filtrovat granulárněji dle protokolu 41 a IPv6 cílové, resp. zdrojové adresy patřící prefixu `2002::/16` [8]. IETF nyní diskutuje, zda používání tohoto mechanismu nezrušit (*draft-ietf-v6ops-6to4-to-historic-11*).
- **IPv6 Rapid Deployment** Tzv. 6rd specifikuje mechanismus pro nasazení IPv6 skrze síť IPv4 poskytovatele služby (ISP) [62]. Staví na principech 6to4 (a vyhýbá se přitom některým jeho problémům), lze jej opět filtrovat blokací protokolu 41.

- **Teredo** Specifikaci Teredo lze nalézt v RFC 4380 [63]. Teredo je schopno zajistit konektivitu IPv6 uzlům dual-stack, které se nachází za zařízením provozujícím NAT (ten je obecně překážkou pro tunelovací mechanismy, neboť mění adresy a čísla portů). Použít Teredo připadá v úvahu pouze pokud nelze použít jinou technologii, zejména kvůli nízké spolehlivosti navázání spojení [64]. Standardně Teredo využívá protokol UDP a cílový port 3544, firewall potom může takové pakety filtrovat. Nic ovšem nebrání útočníkovi provozovat svůj vlastní server Teredo ve veřejném Internetu na nestandardním portu. Pokud firewall zvládá hlubokou inspekci paketů, lze filtrovat odchozí/příchozí pakety IPv4 UDP zapouzdřující pakety IPv6 se zdrojovou/cílovou adresou patřící prefixu `2001::/32`. I při nastavení těchto politik může hostitel zevnitř sítě být schopen nekontrolován komunikovat s klienty Teredo. RFC 4380 [63] doporučuje každému hostiteli jenž chce používat Teredo provozovat vlastní firewall IPv6.

- **6over4** Tento protokol uvádíme pro úplnost, neboť 6over4 spoléhá na nepříliš podporovaný multicast IPv4 a moderní operační systémy 6over4 nepodporují [6]. 6over4 lze snadno firewallem blokovat filtrací všech IPv4 paketů, kde pole Protokol je rovno 41. Selektivněji lze filtrovat zároveň dle pole Protokol a cílové adresy, která patří prefixu `239.0.0.0/8` [8]. Nástupcem 6over4 má být Intra-Site Automatic Tunnel Addressing Protocol (ISATAP) [65].

- **ISATAP** Je určen pro tunelování v rámci jedné sítě, přičemž nespoléhá na multicast IPv4 jako 6over4. ISATAP má narozdíl od 6over4 dobrou podporu v operačních systémech, je např. součástí standardní instalace MS Windows. Nepočítá se s tím, že by překračoval perimetr sítě, nicméně jej lze filtrovat pomocí blokace paketů IPv4 s již zmíněným polem Protokol rovným 41 [8].

- **AYIYA** Uživatelé používající NAT, jenž brání použití protokolu 41 a tedy způsobům tunelování na něm závislým, mohou využít protokolu AYIYA („Anything in Anything“). Protokol umožní zapouzdřit „cokoliv v čemkoliv“ (jako příklad uveďme možnou sekvenci IPv4-UDP-AYIYA-IPv6 [8]) a efektivně překročit NAT. Filtrovat lze pomocí stanovení pravidel pro provoz, který pochází z vnitřní sítě a míří na port TCP 5072 nebo port UDP 5072.

- **Tunnel Setup Protocol** Tzv. TSP [66] má význam v modelu „Tunnel Broker“ [67], který dovolí dynamickou konfiguraci tunelu mezi tunelovacím serverem a klientem (je ustaven kontrolní kanál pro tvorbu, vymazání nebo aktualizaci tunelu a případná registrace uživatelovy adresy IPv6 a jména v DNS). TSP využívá buď TCP nebo UDP, číslo portu je v obou případech 3653 a na tomto základě lze stanovit pravidla filtrování [8].

Na závěr ještě uveďme, že je také dokonce dovoleno tunelovat IPv6 v IPv6. Není přitom jasné, jak má firewall takový provoz filtrovat. Zapouzdřené hlavičky IPv6 navíc mohou mít vlastní rozšiřující hlavičky a celek pak ještě být fragmentován [68].

Obecným doporučením je snažit se o pouze nezbytně nutné použití tunelů.

4 Praktické posouzení firewallů IPv6

Hlavním cílem této kapitoly je vytvořit jednoduché testovací prostředí, kde budou prakticky testovány firewally populárních operačních systémů ve vztahu ke zvoleným problémům prostředí IPv6 předestřeným v kapitole 3, a prezentovat nabyté výsledky. Popsány zde také budou použité nástroje a metodika testování.

4.1 Přehled použitých operačních systémů

Všechny operační systémy (a implementace firewallu, s nimiž jsou distribuovány) provozované v rámci experimentů na hostitelích jsou shrnuty v Tabulce 6. Systémy byly před testováním vždy plně aktualizovány. Vždy se jedná o 64-bitové varianty.

Tabulka 6: Operační systémy

Operační systém	**Firewall**	**Verze jádra**
MS Windows 7 SP1	Windows Firewall	6.1.7601
MS Windows 8.1	Windows Firewall	6.3.9600
MS Windows Server 2008 R2	Windows Firewall	6.1.7601
MS Windows Server 2012 R2	Windows Firewall	6.3.9600
Debian „Wheezy“	Netfilter/iptables	3.2.0-4-amd64
Ubuntu 14.04 LTS	Netfilter/iptables	3.13.0-24-generic
Fedora 21 Workstation	Netfilter/iptables[1]	3.17.4-301-fc21.x86_64
CentOS	Netfilter/iptables[1]	3.10.0-229.el7.x86_64
FreeBSD	IP Firewall[2]	10.0-RELEASE
OpenBSD	Packet Filter	5.6
NetBSD	IP Filter	6.1.5
Oracle Solaris	IP Filter	5.11

[1]Službu `iptables service` zde nahrazuje `firewalld`

[2]Standardně instalovány i Packet Filter a IP Filter

4.2 Popis testovaných firewallů

Hostitelské firewally jsou důležitou součástí vícevrstvé ochrany sítě. Bezpečnostní politika individuální instance hostitelského firewallu definuje, jaký provoz přijímá z Internetu nebo lokální sítě – to je velká výhoda takového firewallu, neboť firewall na perimetru není z principu schopen chránit hostitele před útoky zevnitř vlastní sítě nebo jejího segmentu. Nabízí navíc maximální flexibilitu díky možnosti nastavit specifická pravidla pro každého hostitele.

Na systémech typu Unix jsou pakety filtrovány v jádře a paketové filtry mají buď podobu modulů nahraných na požádaní, nebo jsou přímo vloženy do jádra. K těmto modulům přistupujeme pomocí odpovídajících programů uživatelského prostoru, které umožní vytvářet sady filtrovacích pravidel a další konfiguraci.

Windows Firewall využívá Windows Filtering Platform API, pomocí nějž lze interagovat se zpracováváním paketů v rámci síťového kódu operačního systému. V rámci experimentu budou použity následující softwarové implementace firewallu.

4.2.1 Windows Firewall

Windows Firewall with Advanced Security je stavový firewall schopný inspekce a filtrování paketů IPv4 a IPv6 (lze blokovat/povolit síťový provoz na základě administrátorem definovaných pravidel). V základním nastavení blokuje příchozí pakety pokud nejsou reakcí na předchozí požadavek hostitele nebo tento provoz není explicitně povolen. Explicitní povolení lze definovat pomocí čísla portu, jména aplikace, jména služby a dalších kritérií. Lze také stanovit specifická pravidla pro odchozí provoz hostitele.

4.2.2 Netfilter/iptables

Netfilter je framework pro filtrování vložený do jádra Linux (od verze jádra 2.4). Jádro má od verze 2.6 schopnosti plného stavového firewallu IPv6 [42]. Iptables je program CLI uživatelského prostoru pro konfiguraci množin filtrovacích pravidel. Pro konfiguraci pravidel paketového filtru IPv6 používáme specificky ip6tables. Pra-

vidla filtrování pro IPv4 a IPv6 musí být stanovena zvlášť a jsou na sobě nezávislá. Standardně existují tři sady pravidel (lze vytvářet vlastní):

- **INPUT** Pro příchozí pakety.
- **OUTPUT** Pro odchozí pakety.
- **FORWARD** Pro pakety směrované dále.

Pravidla jsou v těchto řetězcích čtena sekvenčně (záleží tak na jejich pořadí). Pokud paket odpovídá některému z definovaných pravidel, pravidlo nasměruje paket k cíli, který rozhodne o jeho dalším osudu, a prohledávání tabulky pravidel je ukončeno. Standardní cíle jsou tyto:

- **ACCEPT** Paketu je povolen průchod.
- **DROP** Paket je odhozen.
- **QUEUE** Paket je předán do uživatelského prostoru.
- **RETURN** Paket přestane procházet skrze řetězec, kde narazil na RETURN. Pokud procházel hlavním řetězcem (např. INPUT) je na něj aplikována výchozí politika řetězce. Pokud procházel podřízeným řetězcem, vrátí se do hlavního.
- **Uživatelem definovaný řetězec** Paket je předán do uživatelem definovaného řetězce pravidel.

Pokud k paketu není nalezeno odpovídající pravidla, je aplikována výchozí politika řetězce – *výchozí povolení*, nebo *výchozí zamítnutí* paketu. Pravidla lze vytvářet na základě zdrojové/cílové adresy nebo čísla portu, typu zprávy ICMPv6, rozšiřujících hlaviček atd. Jako příklad syntaxe uveďme explicitní povolení ICMPv6 (předpokládáme, že provoz není dále směrován):

```
ip6tables -A INPUT -p icmpv6 -j ACCEPT
ip6tables -A OUTPUT -p icmpv6 -j ACCEPT
```

4.2.3 IP Firewall

Také znám jako IPFW, jedná se o výchozí (stavový) firewall operačního systému FreeBSD. Podporuje IPv4 i IPv6. Na základě definování řetězce pravidel jsou poté

zpracovány pakety jádrem. Pravidla jsou číslovaná od 1 do 65535 a prochází se ve vzestupném pořadí. Při nalezení paketu odpovídajícího pravidla je provedena pravidlem zvolená akce a prohledávání je ukončeno. Pokud takové pravidlo není nalezeno, je provedena výchozí akce pravidla 65535, tj. tiché odhození paketu. Pravidla mohou být pro snazší správu sdružována do tzv. množin [69]. Základní akce které lze provádět s pakety jsou:

- **allow | accept | pass | permit** Tato klíčová slova jsou ekvivalentní a „povolí“ daný paket.
- **check-state** Ověří paket vůči dynamické stavové tabulce. Při nalezení shody je vykonaná akce spojená s daným dynamickým pravidlem, jinak se pokračuje dalším pravidlem.
- **count** Pro každý paket shodný s pravidlem zvýší čítače (ty se používají pro účtování). Poté se pokračuje porovnáním s dalším pravidlem.
- **deny | drop** Tiché odhození paketu.

Příklad syntaxe (explicitní povolení ICMPv6 v příchozím i odchozím směru):

```
ipfw add 100 allow ipv6-icmp from any to any in
ipfw add 110 allow ipv6-icmp from any to any out
```

4.2.4 Packet Filter

Zvaný také pf, Packet Filter je výchozím stavovým firewallem OpenBSD a OS X (od verze 10.7 „Lion“). Podporuje jak IPv4, tak IPv6. Pravidla jsou definována pomocí příkazu `pfctl` a nahrána do jaderné části pf. Pravidla jsou procházena odshora dolů a paket je vyhodnocen vůči všem pravidlům (pokud není použito klíčové slovo `quick`), čímž se liší od firewallů uvedených výše. Může tak existovat více platných pravidel pro paket, ale pouze akce definována posledním shodujícím se pravidlem bude provedena. Lze tak např. nejprve definovat pravidlo stylu `block all packets` a povolit pouze specifické výjimky [42]. Základní proveditelné akce s pakety jsou:

- **pass** Předá paket dále ke zpracování jádru.

- **block** Reaguje dle nastavení `block-policy`, standardně provede tiché odhození paketu.

Příklad syntaxe:

```
pass in inet6 proto ipv6-icmp from any to any
pass out inet6 proto ipv6-icmp from any to any
```

4.2.5 IP Filter

IP Filter je výchozím firewallem distribuovaným s NetBSD a Oracle Solaris. Je stavový a podporuje obě verze protokolu IP. Princip procházení je stejný jako u Packet Filter, tj. odshora dolů a vítězí poslední shoda (pokud v pravidle nefiguruje klíčové slovo `quick`). Stejně jako u Packet Filter se postupuje při stanovování pravidel, základem akcí nad pakety jsou znovu `pass` a `block`. Lze ale např. také použít klíčové slovo `log` pro zaznamenání shody s pravidlem aj. Příklad syntaxe (opět podoba s Packet Filter):

```
pass in quick proto ipv6-icmp from any to any
pass out quick proto ipv6-icmp from any to any
```

4.3 Použitý hardware

Jako zdroj v testech generovaných paketů figuroval fyzický osobní počítač hardwarové konfigurace, kterou shrnuje Tabulka 7.

Tabulka 7: Hardwarová konfigurace zdroje paketů

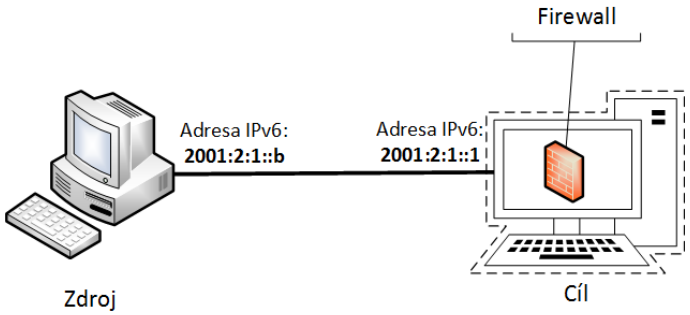
Procesor	AMD Phenom II X4 955 Deneb Quad-Core 3.2 GHz
Paměť	8 GB RAM
Operační systém	Xubuntu 14.04 LTS

Virtuální stroje (cíl, příp. prostředník) byly potom na témže PC virtualizovány s pomocí virtualizačního software Oracle VM VirtualBox verze 4.3.24. Každému virtuálnímu stroji byl vždy přidělen jeden virtuální procesor, 1024 MB operační paměti a zapnuta hardwarová podpora virtualizace (AMD-V). Jak na zdroji, tak cíli

paketů byl během testování vždy aktivní software pro podrobnou analýzu síťového provozu WireShark.

4.4 Testovací scénáře a nástroj firewall6

Prvním nástrojem, který bude použit pro otestování firewallů IPv6, je `firewall6` ze sady `The Hacker's Choice IPv6 Toolkit`. Jde o pravděpodobně nejsofistikovanější z podobných řešení. Spuštěn byl vždy na zdroji v rámci topologie, jíž zpodobňuje Obrázek 13.



Obrázek 13: Testovací topologie 1 (*Zdroj: autor*)

Záměrem `firewall6` je pokusit se o obejití pravidel řízení přístupu (ACL), definovaných na firewallu. Pro účely této práce generoval `firewall6` provoz protokolu TCP s cílovým portem 22, každým z testovaných firewallů byl tato kombinace protokolu a cílového portu blokována. `Firewall6` jednotlivě prověřuje následující scénáře [70], shrnuté v Tabulce 8:

Tabulka 8: Sada testů č. 1

Název	**Popis**
FW1	Odešle obyčejný paket SYN pro zjištění, zda je port blokován.
FW2	Stejně jako FW1, ale s přidanými daty.
FW3	Pole Typ na 2. vrstvě modelu ISO/OSI je chybně nastaveno na hodnotu IPv4. To může vést k obejití pravidla pro IPv6 pomocí síťového kódu IPv6, pokud je rozhodnutí o blokování založeno pouze na vrstvě 2.
FW4	Paketu SYN je přidána hlavička Hop-by-hop o délce 8 bajtů, přičemž její pole Voleb je vyplněno nulami. Paket by měl být zpracován jako paket bez jakékoliv rozšiřující hlavičky.

FW5	Stejným způsobem jako v `FW4` je použita hlavička Volby pro cíl, opět by neměla mít vliv na zpracování paketu.
FW6	Paket obsahuje hlavičku Hop-by-hop a v ní volbu Router Alert (ta indikuje, že by bylo vhodné paket zkoumat blíže, neboť může mít cenné informace pro směrovač). Dle [71] je takto možné v určité situaci obejít řízení přístupu.
FW7	Tento paket obsahuje 3 hlavičky Volby pro cíl. Zmínili jsme, že tato hlavička se může vyskytovat v paketu maximálně dvakrát, nicméně pole Voleb je opět vyplněno nulami a tak by tyto hlavičky měly být ignorovány. Neměly by tak podobně jako v `FW4` a `FW5` mít dopad na zpracování paketu.
FW8	Princip je stejný jako ve scénáři `FW 7` a `FW5`, pouze je použito 130 hlaviček Volby pro cíl. V takovémto množství mohou působit potíže některým zařízením.
FW9	Odešle atomický fragment (tj. paket s hlavičkou Fragmentace, ovšem není ve skutečnosti fragmentován).
FW10	Odešle 2 atomické fragmenty se stejným ID (hlavičky Fragmentace v paketu mají stejné 32-bitové identifikační číslo nutné pro znovusestavení). Atomické fragmenty nesmí být zpracovány stejným způsobem jako skutečně framentovaný síťový provoz (tj. pokusem o znovusestavení [13]).
FW11	2 atomické fragmenty s různým ID.
FW12	3 atomické fragmenty se stejným ID.
FW13	3 atomické fragmenty s různým ID.
FW14	130 atomických fragmentů se stejným ID.
FW15	130 atomických fragmentů s různým ID.
FW16	260 atomických fragmentů se stejným ID.
FW17	260 atomických fragmentů s různým ID.
FW18	Odešle paket s hlavičkou Volby pro cíl o velikosti 2KB. Ta bude muset být kvůli své velikosti fragmentována a může způsobit pád firewallu.

FW19	Stejný princip jako v `FW18`, přidá však ještě další hlavičku Volby pro cíl. Jako zajímavost uveďme, že (před vydaním příslušného patche) jeden takovýto paket dokázal způsobit kompletní pád PC s MS Windows a instalovaným firewallem Kaspersky [72].
FW20	Další varianta `FW18`. Počet hlaviček Volby pro cíl je zvýšen na 32.
FW21	Paket s kombinací 2 hlaviček Volby pro cíl a 2 hlaviček fragmentace.
FW22	Paket s kombinací 4 hlaviček Volby pro cíl a 3 hlaviček fragmentace.
FW23	Pouze první dva fragmenty ze tří mají hodnotu pole Další hlavička nastavenu na TCP.
FW24	Další hlavička prvního fragmentu je nastavena na ICMPv6, druhého na TCP.
FW25	První dva fragmenty se překrývají. Fragmenty jsou zaplněny hlavičkami Volby pro cíl.
FW26	Paket je odeslán ve třech fragmentech. Další hlavička prvního je nastavena na ICMPv6, zbylých dvou na TCP.
FW27	První ze tří fragmentů má Další hlavičku nastavenu na TCP, zbylé dva potom na ICMPv6.
FW28	Poslední dva fragmenty se překrývají.
FW29	Fragmentovaný paket ICMPv6 „ping“, kde je překryt první fragment druhým.
FW30	Na 3 fragmenty fragmentovaný paket ICMPv6 „ping“, kde je překryt druhý fragment třetím.
FW31	V řetězci fragmentů je po druhém fragmentu odeslán další fragment se stejným ID, jenž druhý fragment kompletně překryje, přičemž je shodný s překrytým až na pozměněný datový náklad na konci fragmentu.
FW32	Odeslání několika paketů s různými zdrojovými porty.

Kompletní sadu testů lze snadno spustit po instalaci `The Hacker's Choice IPv6 Toolkit` z příkazové řádky pomocí `./firewall6 <odchozí rozhraní> <cílová adresa> <cílový port>`.

4.5 Praktický test firewallů pomocí firewall6

V Tabulce 9 jsou představeny výsledky jednotlivých operačních systémů a jejich firewallů, testovaných nástrojem `firewall6`. Jak jsme již zmínili, všechen provoz generujeme jako provoz protokolu TCP a cílíme na port 22. Tuto kombinaci portu a protokolu vždy explicitně blokujeme. Úspěšné blokování paketu/paketů značíme pomocí ✓, obejití pravidel firewallu potom ✗.

Jelikož všechny systémy rodiny MS Windows podaly stejné výsledky, jsou v tabulce shrnuty v jedné kolonce.

Tabulka 9: Získané výsledky – Firewall6

	MS Windows	Debian	Ubuntu 14.04	Fedora 21	CentOS	FreeBSD	OpenBSD	NetBSD	Solaris
FW1	✓	✓	✓	✓	✓	✓	✓	✓	✓
FW2	✓	✓	✓	✓	✓	✓	✓	✓	✓
FW3	✓	✓	✓	✓	✓	✓	✓	✓	✓
FW4	✓	✓	✓	✓	✓	✓	✓	✓	✓
FW5	✓	✓	✓	✓	✓	✓	✓	✓	✓
FW6	✓	✓	✓	✓	✓	✓	✓	✓	✓
FW7	✓	✓	✓	✓	✓	✓	✓	✓	✓
FW8	✓	✓	✓	✓	✓	✓	✓	✓	✓
FW9	✓	✓	✓	✓	✓	✓	✓	✓	✓
FW10	✓	✓	✓	✓	✓	✓	✓	✓	✗
FW11	✓	✓	✓	✓	✓	✓	✓	✓	✗
FW12	✓	✓	✓	✓	✓	✓	✓	✓	✗
FW13	✓	✓	✓	✓	✓	✓	✓	✓	✗
FW14	✓	✓	✓	✓	✓	✓	✓	✓	✗
FW15	✓	✓	✓	✓	✓	✓	✓	✓	✗
FW16	✓	✓	✓	✓	✓	✓	✓	✓	✓
FW17	✓	✓	✓	✓	✓	✓	✓	✓	✓
FW18	✓	✓	✓	✓	✓	✓	✓	✗	✓
FW19	✓	✓	✓	✓	✓	✓	✓	✗	✓
FW20	✓	✓	✓	✓	✓	✓	✓	✗	✓
FW21	✓	✓	✓	✓	✓	✓	✓	✗	✓
FW22	✓	✓	✓	✓	✓	✓	✓	✗	✓
FW23	✓	✓	✓	✓	✓	✓	✓	✓	✓
FW24	✓	✓	✓	✓	✓	✓	✓	✓	✓
FW25	✓	✓	✓	✓	✓	✓	✓	✓	✓
FW26	✓	✓	✓	✓	✓	✓	✓	✓	✓
FW27	✓	✓	✓	✓	✓	✓	✓	✓	✓
FW28	✓	✓	✓	✓	✓	✓	✓	✓	✓
FW29	✓	✓	✓	✓	✓	✓	✓	✗	✓
FW30	✓	✓	✓	✓	✓	✓	✓	✗	✓
FW31	✓	✓	✓	✓	✓	✓	✓	✓	✓
FW32	✓	✓	✓	✓	✓	✓	✓	✗	✗

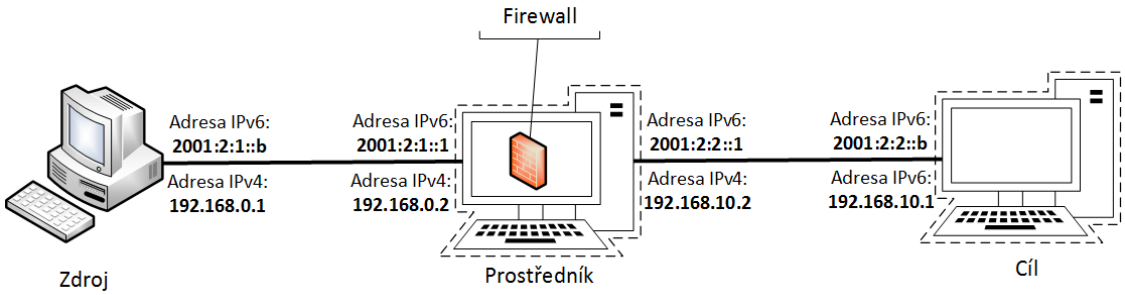
Většina firewallů vyšla z těchto testů s čistým štítem. Podrobnou analýzou logů jednotlivých firewallů lze nicméně zjistit, že některé druhy fragmentovaného provozu

vždy prochází, korektně však nejsou znovusestaveny a systém na ně neodpoví. To lze považovat za uspokojivý výsledek a tak jsou tyto případy v Tabulce 9 hodnoceny.

Problémy měl s některými testovacími scénaři firewall IP Filter. Ten využívají systémy NetBSD a Oracle Solaris, v rámci každého systému měl přitom problém s jinými scénáři. Nad Oracle Solaris IP Filter nezvládá správnou práci s atomickými fragmenty, v NetBSD zase práci s určitými kombinacemi rozšiřujících hlaviček a obě varianty překrývajících se ICMPv6 „ping“. Při odesílání paketů s různými zdrojovými porty v obou systémech některé zdrojové porty blokoval, některé ne – chování je navíc mezi oběma systémy nekonzistentní. Nad NetBSD blokoval pakety se zdrojovými porty 21, 179, 443 a 8080, nad Oracle Solaris se jednalo o porty 53, 80, 111, 123, 179, 443 a 8080. To nelze považovat za dobrý výsledek, lze si představit např. zneužití pro takzvaný „OS fingerprinting“ – vzdálenou identifikaci operačního systému běžícího na daném hostiteli, což představuje určité bezpečnostní riziko, neboť útočník tak získává podrobnější informace o potenciálním cíli svého útoku a může přesněji volit prostředky útoku, například využitím známých bezpečnostních děr daného systému. Jednou ze zodpovědností firewallu je přitom bránit nedovolenému odtoku informací.

4.6 Testovací scénáře a nástroj Scapy

V testování jsme se neomezili pouze na `firewall6`. `Scapy` je nástroj pro manipulaci paketů, jenž pomocí jednoduché syntaxe umožňuje vytvořit téměř libovolný paket. `Scapy` využívá pro svou práci tzv. *raw sockets*, díky čemuž bylo třeba poněkud upravit testovací topologii na topologii z Obrázku 14.



Obrázek 14: Testovací topologie 2 (*Zdroj: autor*)

V původní topologii přichází testovací pakety na cílového hostitele a tam jsou blo-

kovány pravidly pro příchozí pakety – to však pro pakety vytvořené pomocí *raw sockets* neplatí. I přes explicitní zákaz konkrétního paketu tento nebude odhozen. Aby mohla pravidla firewallů být vůbec uplatňována, bylo vždy na hostiteli provozujícím firewall třeba zapnout forwarding IPv6 a provoz předávat dále. Kromě `Scapy` bylo použito navíc nástrojů `randicmp6` a `toobig6` ze sady `The Hacker's Choice IPv6 Toolkit`. V této sadě testů se také místo pokusů obejít firewall zaměříme na to, zda firewall určité druhy paketů předává dále, přestože jsou např. zformovány v rozporu s RFC. Tentokrát budeme zasílat provoz protokolu UDP na port 69 a v kontrastu s minulým postupem vždy vytvoříme explicitní pravidlo pro jeho povolení.

Podrobně jsou jednotlivé testy představeny v Tabulce 10. Vždy očekáváme, že firewall by žádný (s výjimkou scénáře č. 1, kde nás zajímá pouze předání či nepředání úzké skupiny vybraných paketů ze všech ve scénáři vytvořených) z paketů vytvářených v rámci následujících scénařů neměl nechat projít.

Tabulka 10: Sada testů č. 2

#	**Popis**	**Příkaz**
1	Pošle cíli všechny kombinace Typu a Kódu ICMPv6. Pro nás však budou zajímavé jen některé Typy: 100, 101, 127, 130 – 143, 148, 149, 151 – 153, 200, 201, 255. Předání některého z nich je porušením RFC 4890 [16].	`./randicmp6 eth0 2001:2:2::b`
2	Pošle chybovou zprávu ICMPv6 Too Big s příliš malou hodnotou MTU.	`toobig6 eth0 2001:2:2::b 2001:2:1::b 30`
3	Pošle chybovou zprávu ICMPv6 Too Big s příliš velkou hodnotou MTU.	`toobig6 eth0 2001:2:2::b 2001:2:1::b 1000000`
4	Odešleme několik vnořených fragmentů (tj. fragmentovaných fragmentů). Pro vznik takových paketů v síti není důvod. Kompletní skript se nachází v Příloze A.	`python ./NestedFragments`

5	Nespecifikovaná adresa jako zdrojová [42].	`send(IPv6(src="::", dst=""2001:2:2::b"")/ UDP(sport=69, dport=69)/ "testAdresy1")`
6	Adresa zpětné smyčky jako zdrojová [42].	`send(IPv6(src="::1", dst=""2001:2:2::b"")/ UDP(sport=69, dport=69)/ "testAdresy2")`
7	Adresa „všechny uzly na lokálním úseku sítě“ jako zdrojová [42].	`send(IPv6(src="ff02::1", dst=""2001:2:2::b"")/ UDP(sport=69, dport=69)/ "testAdresy3")`
8	Paket se směrovací hlavičkou typu 0. Jeho předání je porušením RFC 5095 [10].	`send(IPv6(dst="2001:2:2::b") / IPv6ExtHdrRouting(type=0, addresses=["2001:2:1::1"]) / UDP(sport=69, dport=69) /"testSH1")`
9	Paket se směrovací hlavičkou typu 2 a polem hlavičky „Segments Left“ jiným od 1. To je porušení RFC 3775 [54].	`send(IPv6(dst="2001:2:2::b")/ IPv6ExtHdrRouting(type=2, addresses=["2001:2:1::b"], segleft=10)/ UDP(sport=69,dport=69)/ "testSH2")`
10	Paket s nealokovanou směrovací hlavičkou typu 100. Předání je porušením RFC 2460 [9].	`send(IPv6(dst="2001:2:2::b")/ IPv6ExtHdrRouting(type=100, segleft=100)/ UDP(sport=69, dport=69)/ "testSH3")`

11	Paket s neplatným řetězcem hlaviček: Hop-by-hop, Hop-by-hop. Porušení RFC 2460 [9].	`send(IPv6(dst="2001:2:2::b")/ IPv6ExtHdrHopByHop()/ IPv6ExtHdrHopByHop()/ UDP(sport=69, dport=69)/ "testRH1")`
12	Paket s neplatným řetězcem hlaviček: Volby pro cíl, Hop-by-hop. Porušení RFC 2460 [9].	`send(IPv6(dst="2001:2:2::b")/ IPv6ExtHdrDestOpt()/ IPv6ExtHdrHopByHop()/ UDP(sport=69, dport=69)/ "testRH2")`
13	Paket s neplatným řetězcem hlaviček: Hop-by-hop, Volby pro cíl, Směrovací hlavička typu 2, Hop-by-hop. Porušení RFC 2460 [9].	`send(IPv6(dst="2001:2:2::b")/ IPv6ExtHdrHopByHop()/ IPv6ExtHdrDestOpt()/ IPv6ExtHdrRouting(type=2, addresses=[2001:2:1::1])/ IPv6ExtHdrHopByHop()/ UDP(sport=69, dport=69)/ "testRH3")`
14	Pole PadN smí legálně obsahovat pouze nuly (RFC 2460 [9]), jinak se jedná o tzv. skrytý kanál.	`send(IPv6(dst="2001:2:2::b")/ IPv6ExtHdrDestOpt(options= [PadN(optdata= ("11111111"))]+[PadN(optdata= ("44444444"))])/ UDP(sport=69, dport=69)/ "testSK")`
15	Paket s nevalidním řetězcem voleb hlavičky Hop-by-hop (kromě Pad1 a PadN se vždy mohou vyskytnout pouze jednou), dvakrát se zde vyskytuje volba Jumbo Payload. Více RFC 4942 [5].	`send(IPv6(dst="2001:2:2::b")/ IPv6ExtHdrHopByHop(options= [Jumbo(), PadN(), Jumbo()])/ UDP(sport=69, dport=69)/ "testVolby1")`

16	Nevalidní řetězec voleb hlavičky Volby pro cíl, dvakrát výskyt Jumbo Payload. Více RFC 4942 [5].	`send(IPv6(dst="2001:2:2::b")/ IPv6ExtHdrDestOpt(options= [Jumbo(), PadN(), Jumbo()])/ UDP(sport=69, dport=69)/ "testVolby2")`
17	Nevalidní řetězec voleb hlavičky Hop-by-hop, dvakrát výskyt Router Alert. Více RFC 4942 [5].	`send(IPv6(dst="2001:2:2::b")/ IPv6ExtHdrHopByHop(options= [RouterAlert(), Pad1(), RouterAlert()])/ UDP(sport=69, dport=69)/ "testVolby3")`
18	Nevalidní řetězec voleb hlavičky Volby pro cíl, dvakrát výskyt Router Alert. Více RFC 4942 [5].	`send(IPv6(dst="2001:2:2::b")/ IPv6ExtHdrDestOpt(options= [RouterAlert(), Pad1(), RouterAlert()])/ UDP(sport=69, dport=69)/ "testVolby4")`
19	Nevalidní řetězec voleb hlavičky Hop-by-hop, dvakrát výskyt Tunnel Encapsulation Limit. Více RFC 4942 [5].	`send(IPv6(dst="2001:2:2::b")/ IPv6ExtHdrHopByHop(options= [HBHOptUnknown(otype=4, optlen=1), PadN(), HBHOptUnknown(otype=4, optlen=1)])/ UDP(sport=69, dport=69)/ "testVolby5")`
20	Nevalidní řetězec voleb hlavičky Volby pro cíl, dvakrát výskyt Tunnel Encapsulation Limit. Více RFC 4942 [5].	`send(IPv6(dst="2001:2:2::b")/ IPv6ExtHdrDestOpt(options= [HBHOptUnknown(otype=4, optlen=1), PadN(), HBHOptUnknown(otype=4, optlen=1)])/ UDP(sport=69, dport=69)/ "testVolby6")`

21	PadN by neměl vkládat více než 5 oktetů [42].	`send(IPv6(dst="2001:2:2::b")/ IPv6ExtHdrDestOpt(options= [PadN(optdata= "00000000000

4.7 Praktický test firewallů pomocí Scapy

V Tabulce 12 jsou představeny výsledky jednotlivých operačních systémů a jejich firewallů, testovaných nástroji `Scapy` plus `randicmp6` a `toobig6` ze sady `The Hacker's Choice IPv6 Toolkit` dle scénářů navržených výše. Tentokrát budeme generovat provoz UDP, cílený na port 69 a na firewallu jej explicitně povolíme. Výjimkou jsou scénáře 22 a 23, kdy zapouzdřujeme IPv6 v IPv6, resp. v IPv4. Pro tyto 2 scénáře vždy vytvoříme na firewallu explicitní blok kombinace protokolu UDP a cílového portu 69. Bernou mincí pro nás v těchto testech je, zda firewall předává/blokuje daný paket. Kromě zmíněných scénářů 22 a 23 (kde očekáváme blokaci paketů na základě explicitních pravidel řízení přístupu) vytváříme pakety v rozporu s RFC a tyto by neměly být předány dále. Správné chování firewallu opět značíme pomocí ✓, v opačném případě udělíme ✗.

Testovací scénáře jsou číslovány a organizovány stejně jako v Tabulkách 10 a 11.

Tabulka 12: Získané výsledky – Scapy

	Debian	Ubuntu 14.04	Fedora 21	CentOS	FreeBSD	OpenBSD	NetBSD	Solaris
1	✗	✗	✗	✗	✗	✗	✗	✗
2	✗	✗	✗	✗	✗	✗	✗	✗
3	✗	✗	✗	✗	✗	✗	✗	✗
4	✗	✓	✓	✗	✗	✓	✗	✓
5	✓	✓	✓	✓	✓	✓	✓	✓
6	✓	✓	✓	✓	✓	✓	✓	✓
7	✓	✓	✓	✓	✓	✓	✓	✓
8	✗	✗	✗	✗	✗	✓	✗	✗
9	✗	✗	✗	✗	✗	✗	✗	✗
10	✗	✗	✗	✗	✓	✗	✗	✗
11	✗	✗	✗	✗	✗	✗	✗	✗
12	✗	✗	✗	✗	✗	✗	✗	✗
13	✗	✗	✗	✗	✗	✗	✗	✗
14	✗	✗	✗	✗	✗	✗	✗	✗
15	✓	✓	✓	✓	✓	✓	✓	✓
16	✗	✗	✗	✗	✗	✗	✗	✗
17	✗	✗	✗	✗	✓	✓	✓	✓
18	✗	✗	✗	✗	✗	✗	✗	✗
19	✓	✓	✓	✓	✗	✓	✗	✓
20	✗	✗	✗	✗	✗	✗	✗	✗
21	✗	✗	✗	✗	✗	✗	✗	✗
22	✗	✗	✗	✗	✗	✗	✗	✗
23	✗	✗	✗	✗	✗	✗	✗	✗

Jak patrno, žádný z firewallů se implicitně neřídí doporučeními RFC vzhledem k filtrování ICMPv6 (scénář č.1). Všechny testované firewally nicméně umožňují tvořit granulární pravidla pro filtrování ICMPv6 na základě Typu a Kódu. Je tedy zodpovědností administrátora sestavit takový soubor pravidel řízení přístupu, jenž zajistí korektní chování firewallu v této oblasti.

Všechny firewally také nechají projít chybové zprávy ICMPv6 TooBig s příliš malými či velkými hodnotami MTU (scénáře č. 2, respektive č. 3). Administrátor firewallu nemá možnost se takovým paketům bránit specificky na základě hodnoty MTU. Zprávy ICMPv6 TooBig přitom patří mezi ty, jež jsou nezbytné pro komunikaci IPv6 a nesmí být obecně filtrovány.

Ve scénáři č. 4 jsme vytvářeli fragmentované fragmenty – reálně pro vznik takových paketů není racionální důvod. Distribuce GNU/Linux se starší verzí jádra (v našem testu Debian a CentOS) tyto pakety předávají, Ubuntu 14.04 LTS a Fedora 21 tyto pakety blokují. Je zde tedy patrný vývoj ke korektnímu chování firewallu. Zajímavý výsledek poskytly také systémy NetBSD a Oracle Solaris, jež oba používají firewall IP Filter. Zatímco verze IP Filter distribuovaná s Oracle Solaris tento provoz blokuje, verze v NetBSD nikoliv.

Všechny firewally si dobře poradily se scénáři č. 5–7, kde byly generovány pakety se zdrojovými adresami, jež nemají být předávány dále.

Ve scénářích č. 8–10 jsme se zaměřili na směrovací hlavičky. Netfilter/iptables s těmito hlavičkami ve výchozím nastavení evidentně vůbec nepracuje, je třeba využít modul `rt` a filtrovat ve speciálně vytvořeném řetězci pravidel. Po vytvoření specifických pravidel je možné dosáhnout chování zcela kompatibilního s RFC [73]. IP Firewall a Packet Filter dokázaly ve výchozí konfiguraci každý v jednom případě uspět. IP Firewall systému FreeBSD korektně nepředá nealokovaný typ směrovací hlavičky, nedodržuje ale RFC 5095 [10], tj. zastarání Směrovací hlavičky typu 0. Packet Filter systému OpenBSD zjevně provádí inspekci typu směrovací hlavičky, ale nezajímá se o pole „Segments Left“, jinak by odhodil paket ze scénáře č. 9. Pro IP Filter je třeba vytvořit explicitní zákaz/zákazy (s pomocí klíčového slova `v6hdrs`), jinak s těmito hlavičkami nepracuje.

Scénáře č. 11–13 generují pakety s neplatnými řetězci rozšiřujících hlaviček. Žádný

z firewallů zde ve výchozí konfiguraci neuspěl. Netfilter/iptables musí využít modulu `ipv6header` nebo `ip6headerorder` a vytvořit speciální řetězec pravidel. Administrátor přitom musí manuálně definovat všechny kombinace hlaviček, které má firewall zamítnout. Obdobná situace platí i pro ostatní firewally. IP Firewall a IP Filter podporují filtrování jednotlivých rozšiřujících hlaviček, nepodařilo se nicméně definovat pravidla pro filtrování řetězců těchto hlaviček. Manuálové stránky Packet Filter neuvádí, jak rozšiřující hlavičky explicitně filtrovat (je-li to vůbec možné).

Žádný z firewallů nekontroluje obsah PadN (scénář č. 14) a nelze tuto situaci ani řešit explicitní definicí pravidel firewallu.

Tvorba paketů s neplatnými řetězci voleb rozšiřujících hlaviček byla účelem scénářů číslo 15–20. Netfilter/iptables pakety ze scénářů č. 15 a 19 sám odhodí, pro explicitní práci s těmito volbami je zapotřebí modulů `hbh` a `dst` a vyjmenovat všechny kombinace určené k odhození. IP Firewall, Packet Filter ani IP Filter v manuálových stránkách neuvádí způsob definování pravidel pro tyto scénáře. Přitom však implicitně vždy některé pakety těchto scénářů blokují. Přesný důvod tohoto chování není jasný.

Žádný z firewallů nekontroluje délku PadN (scénář č. 21) a opět nelze tuto situaci řešit ani explicitní definicí pravidel firewallu.

Neúspěchem pro všechny zúčastněné skončily také scénáře č. 22 a 23, kde jsme pakety IPv6 zapouzdřili další hlavičkou IPv6, resp. IPv4. Odsunutím hlavičky UDP zapouzdřením paketu došlo k obejití pravidel řízení přístupu na základě protokolu a portu na všech firewallech. Administrátor musí být na tento provoz připraven a filtrovat na základě protokolu 41 (pro ip6tables takové pravidlo může mít podobu `ip6tables -A FORWARD -p 41 -j DROP`).

Za nejvyspělejší z testovaných firewallů lze nejspíše považovat netfilter/iptables, zejména díky systému modulů, které umožňují tvořit granulární pravidla.

Z výsledků je zřejmé, že reálné prostředí firewallů IPv6 se značně rozchází s předpoklady RFC. Kvalitním souborem pravidel řízení přístupu je nicméně možné dosáhnout chování RFC mnohem bližšího, ani takovým souborem se však nelze bránit proti skrytým tunelům. Jeho vytvoření může být navíc velmi pracné (nutnost explicitně

vyjmenovat kombinace rozšiřujích hlaviček k odhození) a tedy náchylné k chybám. S tím souvisí i vyšší nároky na znalosti a čas administrátora. Protokol IPv6 tedy život v oblasti firewallů nijak neusnadňuje, přibylo mnoho nových možností (rozšiřující hlavičky...) a s tím i potenciálních vektorů útoků. Pro rozšiřující hlavičky není v některých firewallech ani adekvátní podpora, to platí i pro obsah ICMPv6 zpráv (TooBig).

Doporučením je budovat vícevrstvou obranu a přinejmenším doplnit firewall systémy IDS/IPS pro detekci/prevenci průniku, jež detekují (nebo reagují na) nežádoucí aktivitu v síti a to v reálném čase.

Závěr

Cílem této práce bylo zmapovat problematiku firewallů v prostředí protokolu IPv6. Pro splnění tohoto cíle byl nejdříve nastudován současný stav problematiky, kdy tato část se opírá zejména o související RFC a poskytuje čtenáři aktuální přehled výzkumu v oblasti. Dále byly popsány obecné principy firewallů, jejich vztah k modelu ISO/OSI, postavení v síťové architektuře a také hrozby, kterým jsou navrženy čelit. Prozkoumány byly relevantní partie specifikace protokolu IPv6 spolu s definicí nezbytných pojmů. Zde vyzvedněme např. rozšiřující hlavičky IPv6, jež jsou citlivým místem firewallů IPv6.

Firewally IPv6 byly poté prakticky posouzeny, za tímto účelem bylo vybudováno laboratorní, částečně virtualizované prostředí. Testováno bylo několik hostitelských firewallů distribuovaných standardně s vybranými operačními systémy. Tyto firewally pravděpodobně získají s nástupem protokolu IPv6 ještě větší důležitost vzhledem k návratu myšlenky skutečné end-to-end konektivity. Vyšší podíl zodpovědnosti za bezpečnost se přesune z centralizované obrany perimetru sítě na každého hostitele.

Každý firewall byl podroben sérií testů pomocí vybraných softwarových nástrojů, kdy tyto testy byly zaměřeny na známá problematická místa specifikace IPv6. Chování jednotlivých firewallů bylo zaznamenáno a diskutováno. Závěrem je, že firewally se chovají nekonzistentně nejen s RFC, ale i mezi sebou. Dosáhnout korektnějšího chování, bližšího RFC je třeba vytvořením kvalitního souboru pravidel řízení přístupu. To vyžaduje znalosti a čas administrátora. Život v prostředí IPv6 zjevně nebude snazší než v prostředí IPv4, situaci ještě komplikuje nutnost provozovat přechodové metody. Mimo jiné je vhodné budovat vícevrstvou obranu, alespoň nasazením systémů IDP/IPS spolu s firewally.

Spolupráce firewallů a IDS/IPS by mohla být námětem pro další výzkum, stejně jako opravdu zevrubný průzkum práce firewallů s Mobile IPv6 a jeho rozšířením NEMO (Network Mobility). Další prostor se potom nabízí pro výzkum vztahu firewallů, některých přechodových metod a IPSec, v této práci použité softwarové nástroje pro ně mají jen minimální nebo žádnou podporu.

Použitá literatura

[1] ICANN. Available pool of unallocated IPv4 internet addresses now completely emptied [online]. <https://www.icann.org/en/system/files/press-materials/release-03feb11-en.pdf>, 2011-02-03 [cit. 2014-10-13].

[2] GOOGLE. IPv6 – Google [online]. <https://www.google.com/intl/en/ipv6/statistics.html>, 2014 [cit. 2014-10-13].

[3] HAIN, T. RFC 2993 – Architectural Implications of NAT [online]. <http://tools.ietf.org/html/rfc2993>, 2000 [cit. 2014-10-27].

[4] DE VELDE, V., et al. RFC 4864 – Local Network Protection for IPv6 [online]. <http://tools.ietf.org/html/rfc4864>, 2007 [cit. 2014-10-27].

[5] DAVIES, E., et al. RFC 4942 – IPv6 Transition/Coexistence Security Considerations [online]. <http://tools.ietf.org/html/rfc4942>, 2007 [cit. 2014-10-27].

[6] STEFFANN, S., et al. RFC 7059 – A Comparison of IPv6-over-IPv4 Tunnel Mechanisms [online]. <https://tools.ietf.org/html/rfc7059>, 2013 [cit. 2014-10-28].

[7] WOODYATT, J. RFC 6092 – Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service [online]. <https://tools.ietf.org/html/rfc6092>, 2011 [cit. 2014-10-28].

[8] GONT, F.; LIU, W. RFC 7123 – Security Implications of IPv6 on IPv4 Networks [online]. <https://tools.ietf.org/html/rfc7123>, 2014 [cit. 2014-10-28].

[9] DEERING, S.; HINDEN, R. RFC 2460 – Internet Protocol, Version 6 (IPv6) Specification [online]. <https://www.ietf.org/rfc/rfc2460.txt>, 1998 [cit. 2014-10-30].

[10] ABLEY, J., et al. RFC 5095 – Deprecation of Type 0 Routing Headers in IPv6 [online]. <http://www.ietf.org/rfc/rfc5095.txt>, 2007 [cit. 2014-10-30].

[11] PERKINS C., E., et al. RFC 6275 – Mobility Support in IPv6 [online]. <https://tools.ietf.org/html/rfc6275>, 2011 [cit. 2014-11-3].

[12] KRISHNAN, S. RFC 5722 – Handling of Overlapping IPv6 Fragments [online]. <https://tools.ietf.org/html/rfc5722>, 2009 [cit. 2014-10-30].

[13] GONT, F. RFC 6946 – Processing of IPv6 „Atomic“ Fragments [online]. <http://tools.ietf.org/html/rfc6946>, 2013 [cit. 2014-11-2].

[14] GONT, F. RFC 6980 – Security Implications of IPv6 Fragmentation with IPv6 Neighbor Discovery [online]. <https://tools.ietf.org/html/rfc6980>, 2013 [cit. 2014-10-30].

[15] CONTA, A., et al. RFC 4443 – Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification [online]. <https://tools.ietf.org/html/rfc4443>, 2006 [cit. 2014-10-28].

[16] DAVIES, E.; MOHACSI, J. RFC 4890 – Recommendations for Filtering ICMPv6 Messages in Firewalls [online]. <http://www.ietf.org/rfc/rfc4890.txt>, 2007 [cit. 2014-10-28].

[17] AMANTE, S., et al. RFC 6437 – IPv6 Flow Label Specification [online]. <https://tools.ietf.org/html/rfc6437>, 2011 [cit. 2014-10-30].

[18] LE, F., et al. RFC 4487 – Mobile IPv6 and Firewalls: Problem Statement [online]. <http://www.ietf.org/rfc/rfc4487.txt>, 2006 [cit. 2014-11-2].

[19] NORDMARK, E.; BAGNULO, M. RFC 5533 – Shim6: Level 3 Multihoming Shim Protocol for IPv6 [online]. <http://tools.ietf.org/html/rfc5533>, 2009 [cit. 2014-10-28].

[20] ABLEY, J., et al. RFC 6629 – Considerations on the Application of the Level 3 Multihoming Shim Protocol for IPv6 (Shim6) [online]. <https://tools.ietf.org/html/rfc6629>, 2012 [cit. 2014-10-28].

[21] ALANGAR, V.; SWAMINATHAN, A. Ipv6 Security: Issue Of Anonymity. *International Journal Of Engineering And Computer Science*, 2:2486–2493, 2013.

[22] GONT, F., et al. RFC 7112 – Implications of Oversized IPv6 Header Chains [online]. <https://tools.ietf.org/html/rfc7112>, 2014 [cit. 2014-11-2].

[23] KIM, J.; CHO, H.; MUN, G.; et al. Experiments and Countermeasures of Security Vulnerabilities on Next Generation Network. *Future Generation Communication and Networking (FGCN 2007)*, 2:559–564, 2007.

[24] LIM, J.; KIM, Y.; KIM, K.; YANG, Z. Packet Filter Algorithm to prevent the security hole of routing header in IPv6. *2006 SICE-ICASE International Joint Conference*, pages 3924–3927, 2006.

[25] ATLASIS, A. Attacking IPv6 Implementation Using Fragmentation [online]. <https://media.blackhat.com/bh-eu-12/Atlasis/bh-eu-12-Atlasis-Attacking_IPv6-WP.pdf>, 2012 [cit. 2014-11-1].

[26] ATLASIS, A. Fragmentation (Overlapping) Attacks One Year Later... [online]. <https://www.troopers.de/wp-content/uploads/2013/01/TROOPERS13-Fragmentation_Overlapping_Attacks_Against_IPv6_One_Year_Later-Antonios_Atlasis.pdf>, 2013 [cit. 2014-11-1].

[27] VAN DEN BROEK, G.; VAN RIJSWIJK-DEIJ, R.; SPEROTTO, A.; PRAS, A. DNSSEC meets real world: dealing with unreachability caused by fragmentation. *IEEE Communications Magazine*, 52:154–160, 2014.

[28] GONT, F.; LINKOVA, J. IPv6 Extension Headers in the Real World v2.0 [online]. <http://www.iepg.org/2014-07-20-ietf90/iepg-ietf90-ipv6-ehs-in-the-real-world-v2.0.pdf>, 2014-07-20 [cit. 2015-2-17].

[29] LAI, Y.; JIANG, G.; LI, J.; YANG, Z. Design and Implementation of Distributed Firewall System for IPv6. *2009 International Conference on Communication Software and Networks*, pages 428–432, 2009.

[30] BELLOVIN, S. M. Distributed firewalls. *IEEE Communications Magazine*, 32:50–57, 1999.

[31] SCARFONE, K.; HOFFMAN, P. Guidelines on Firewalls and Firewall Policy. <http://csrc.nist.gov/publications/nistpubs/800-41-Rev1/sp800-41-rev1.pdf>, 2009 [cit. 2014-11-1].

[32] NEWMAN, D. RFC 2647 – Benchmarking Terminology for Firewall Performance [online]. <https://tools.ietf.org/html/rfc2647>, 1999 [cit.

2014-12-22].

[33] FRAHIM, J.; SANTOS, O. *Cisco ASA: all-in-one firewall, IPS, and VPN adaptive security appliance*. Indianapolis: Cisco Press, 3rd edition, 2013. ISBN 15-871-4307-0.

[34] SHINDER, T. W. *Best damn firewall book period*. Oxford: Elsevier Science [distributor], 2nd edition, 2007. ISBN 15-974-9218-3.

[35] MORAES, A. M. da S. P. de. *Cisco firewalls*. Indianapolis: Cisco Press, 2011. ISBN 978-1-58714-109-6.

[36] GYEONGOU, Y.; BYUNGGYU, N.; WAN, S. Y., et al . Firewall protection profile v2.0 [online]. <http://www.commoncriteriaportal.org/files/ppfiles/FW%20PP-93-EN.pdf>, 2008 [cit. 2014-12-25].

[37] ZWICKY, E. *Building Internet firewalls*. Cambridge: O'Reilly, 2nd edition, 2000. ISBN 15-659-2871-7.

[38] LOWDER, J. Deploying Host-Based Firewalls Across the Enterprise: A Case Study [online]. <http://www.giac.org/paper/gsec/1771/deploying-host-based-firewalls-enterprise-case-study/103172>, 2002 [cit. 2015-1-1].

[39] INFORMATION SCIENCES INSTITUTE, UNIVERSITY OF SOUTHERN CALIFORNIA. RFC 791 – Internet Protocol [online]. <https://tools.ietf.org/html/rfc791>, 1981 [cit. 2014-12-26].

[40] BAKER, F., et al. RFC 4192 – Procedures for Renumbering an IPv6 Network without a Flag Day [online]. <https://tools.ietf.org/html/rfc4192>, 2005 [cit. 2014-12-27].

[41] GORALSKI, W. IPv4 and IPv6 Addressing - Part 4: CIDR, VLSM & IPv6 addressing [online]. <http://www.embedded.com/print/4200905>, 2010 [cit. 2015-2-22].

[42] HOGG, S., VYNCKE, E. *IPv6 security*. Indianapolis: Cisco Press, 2009. ISBN 978-1-58705-594-2.

[43] THOMSON, S., et al. RFC4862 – IPv6 Stateless Address Autoconfiguration [online]. <https://tools.ietf.org/html/rfc4862>, 2007 [cit. 2015-1-1].

[44] PAASCH, Ch. Supporting IPv6 host-based multihoming (shim6) in Linux firewalls. <http://inl.info.ucl.ac.be/system/files/memoire_0.pdf>, 2009 [cit. 2014-12-28].

[45] BARRÉ, S. Implementation and evaluation of the Shim6 protocol in the Linux kernel [online]. <http://inl.info.ucl.ac.be/system/files/main_1.pdf>, 2011 [cit. 2014-12-28].

[46] BARRÉ, S. LinShim6 [online]. <http://inl.info.ucl.ac.be/softwares/linshim6>, 2014 [cit. 2014-12-28].

[47] BARRÉ, S. Implementation and Assessment of Modern Host-based Multipath Solutions. <http://inl.info.ucl.ac.be/system/files/phd-thesis.pdf>, 2011 [cit. 2014-12-28].

[48] FORD, A. et al. RFC6824 – TCP Extensions for Multipath Operation with Multiple Addresses [online]. <https://tools.ietf.org/html/rfc6824>, 2013 [cit. 2015-2-21].

[49] TROAN, O., et al. RFC 7157 – IPv6 Multihoming without Network Address Translation [online]. <https://tools.ietf.org/html/rfc7157>, 2014 [cit. 2014-12-28].

[50] NARTEN, T., et al. RFC 4861 – Neighbor Discovery for IP version 6 (IPv6) [online]. <https://tools.ietf.org/html/rfc4861>, 2007 [cit. 2014-10-30].

[51] McCANN, J. RFC 1981 – Path MTU Discovery for IP version 6 [online]. <https://tools.ietf.org/html/rfc1981>, 1996 [cit. 2014-10-30].

[52] FRANKEL, S.; GRAVEMAN, R., PEARCE, J. Guidelines for the Secure Deployment of IPv6. <http://csrc.nist.gov/publications/nistpubs/800-119/sp800-119.pdf>, 2010 [cit. 2014-12-30].

[53] Tech Stuff - IPv6 [online]. <http://www.zytrax.com/tech/protocols/ipv6.html>, 2015 [cit. 2015-2-21].

[54] JOHNSON, D., et al. RFC 3775 – Mobility Support in IPv6 [online]. <https://tools.ietf.org/html/rfc3775>, 2004 [cit. 2015-1-1].

[55] BORMAN, D., et al. RFC 2675 – IPv6 Jumbograms [online]. <https://tools.ietf.org/html/rfc2675>, 1999 [cit. 2015-2-16].

[56] KENT, S.; SEO, K. RFC 4301 – Security Architecture for the Internet Protocol [online]. <https://tools.ietf.org/html/rfc4301>, 2005 [cit. 2015-2-17].

[57] POTYRAJ, C. A. Firewall Design Considerations for IPv6. [online]. <http://nsabackups.com/ia/_files/ipv6/I733-041R-2007.pdf>, 2007 [cit. 2014-11-2].

[58] TEUFEL, M. Mobile IPv6 [online]. <http://www.ibk.tuwien.ac.at/~ipv6/mipl.htm>, 2002 [cit. 2015-2-21].

[59] STEINLEITNER, N. Firewall Traversal in Mobile IPv6 Networks. <http://ediss.uni-goettingen.de/bitstream/handle/11858/00-1735-0000-0006-B3B0-2/steinleitner.pdf?sequence=1>, 2008 [cit. 2014-12-29].

[60] DESMEULES, R. *Cisco Self-Study: Implementing Cisco IPv6 Networks (IPV6)* . Indianapolis: Cisco Press, 1st edition, 2003. ISBN 1587142864.

[61] WEBER, J. IPv6 Security Test Laboratory [online]. <http://www.hpc.mil/images/hpcdocs/ipv6/masterthesis_johannes_weber_ipv6securitytestlaboratory.pdf>, 2013 [cit. 2015-2-19].

[62] TOWNSLEY, W.; TROAN, O. RFC5969 – IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) – Protocol Specification [online]. <https://tools.ietf.org/html/rfc5969>, 2010 [cit. 2015-2-20].

[63] HUITEMA, C. RFC4380 – Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs) [online]. <https://tools.ietf.org/html/rfc4380>, 2006 [cit. 2015-2-21].

[64] HUSTON, G. Testing Teredo [online]. <http://www.potaroo.net/ispcol/2011-04/teredo.html>, 2011 [cit. 2015-2-21].

[65] TEMPLIN, F., et al. RFC5214 – Intra-Site Automatic Tunnel Addressing Protocol (ISATAP) [online]. <https://tools.ietf.org/html/rfc5214>, 2008 [cit. 2015-2-20].

[66] BLANCHET, M.; PARENT, F. RFC5572 – IPv6 Tunnel Broker with the Tunnel Setup Protocol (TSP) [online]. <https://tools.ietf.org/html/rfc5572>, 2010 [cit. 2015-2-21].

[67] DURAND, A., et al. RFC3053 – IPv6 Tunnel Broker [online]. <https://tools.ietf.org/html/rfc3053>, 2001 [cit. 2015-2-21].

[68] ATLASIS, A. Advanced Attack Techniques against IPv6 Networks [online]. <https://www.ernw.de/download/Advanced%20Attack%20Techniques%20against%20IPv6%20Networks-final.pdf>, 2013 [cit. 2015-2-19].

[69] IPFW [online]. <http://www.freebsd.cz/doc/en/books/handbook/firewalls-ipfw.html>, [cit. 2015-2-27].

[70] FOJTŮ, P. Vulnerabilities and Threats in IPv6 Environment [online]. <https://otik.uk.zcu.cz/bitstream/handle/11025/7583/fojtu_a10n0030p.pdf?sequence=1>, 2013 [cit. 2015-4-19].

[71] HEUSE, M. IPv6 Insecurity Revolutions [online]. <https://conference.hitb.org/hitbsecconf2012kul/materials/D1T2%20-%20Marc%20Heuse%20-%20IPv6%20Insecurity%20Revolutions.pdf>, 2012 [cit. 2015-4-20].

[72] LEYDEN, J. Single IPv6 packet KILLS Kaspersky-protected PCs, fix emerges [online]. <http://www.theregister.co.uk/2013/03/08/kaspersky_ipv6_lock_up_bug/>, 2013 [cit. 2015-4-20].

[73] EGGERT, O. Testing IPv6 Firewalls with ft6 [online]. <https://www.troopers.de/wp-content/uploads/2013/11/TROOPERS14-Testing_IPv6_Firewalls_with_ft6-Oliver_Eggert.pdf>, 2014 [cit. 2015-4-27].

[74] IPv6 Extension Headers - New Features, and New Attack Vectors.py [online]. <https://www.ernw.de/download/IPv6%20Extension%20Headers%20-%20New%20Features,%20and%20New%20Attack%20Vectors.py>, [cit. 2015-4-26].

A Příloha I – Skript

V rámci práce je použit převzatý skript pro tvorbu vnořených fragmentů [74], vytvořený v programu Scapy. Do skriptu byla přidána zdrojová a cílová adresa IP, proměnná `length` a odkomentován řádek `no_of_fragments`.

A.1 NestedFragments

```
#!/usr/bin/python
from scapy.all import *

sip="2001:2:1::b"
ip="2001:2:2::b"
length=1

myid=random.randrange(1,4294967296,1)
myid2=random.randrange(1,4294967296,1)
icmpid=random.randrange(0,65535,1)
no_of_fragments = 3
#no_of_fragments = 65537
payload1=Raw("AAAAAAAA"*(length))
icmpv6=ICMPv6EchoRequest(data=payload1, id=icmpid)
ipv6_1=IPv6(src=sip, dst=ip, plen=(length+1)*8)
csum=in6_chksum(58, ipv6_1/icmpv6, str(icmpv6))
icmpv6=ICMPv6EchoRequest(cksum=csum, data=payload1, id=icmpid)
ipv6_1=IPv6(src=sip, dst=ip, plen=8*2)
frag2=IPv6ExtHdrFragment(offset=0, m=0, id=myid2, nh=44)
for i in range(0, no_of_fragments):
        frag1=IPv6ExtHdrFragment(offset=i, m=1, id=myid, nh=44)
        packet=ipv6_1/frag1/frag2
        send(packet)
frag1=IPv6ExtHdrFragment(offset=no_of_fragments, m=1, id=myid, nh=44)
frag2=IPv6ExtHdrFragment(offset=0, m=0, id=myid2, nh=58)
packet=ipv6_1/frag1/frag2
send(packet)
ipv6_1=IPv6(src=sip, dst=ip, plen=8*(length+2))
frag1=IPv6ExtHdrFragment(offset=no_of_fragments+1, m=0, id=myid, nh=44)
packet=ipv6_1/frag1/icmpv6
send(packet)
```