KAPITOLA 5

Pokročilé funkce programu Wireshark

V předchozích kapitolách jsme se zabývali základy práce s programem Wireshark. Nyní můžeme přejít k jeho funkcím pro analýzu a tvorbu grafů. V této kapitole představíme některé z těchto užitečných funkcí, včetně oken Endpoints a Conversation, detailů překladu názvů, rozboru protokolů, sledování datových proudů a vstupně-výstupních grafů.

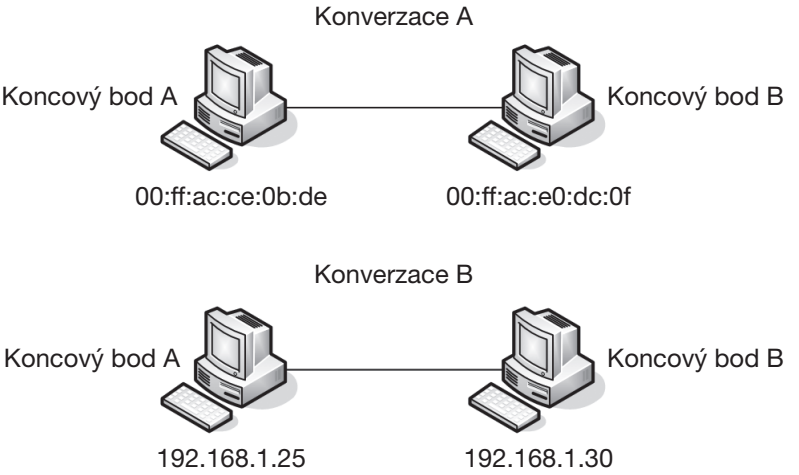
Koncové body sítě a konverzace

Komunikace v síti může probíhat jen za předpokladu, že lze přenášet data alespoň mezi dvěma zařízeními. *Koncový bod* (endpoint) je zařízení, které odesílá nebo přijímá data v síti. Komunikace protokolu TCP/IP například zahrnuje dva koncové body: IP adresy systémů, které odesílají a přijímají data, např. 192.168.1.25 a 192.168.1.30.

Na vrstvě 2 spolu komunikují dvě fyzické síťové karty označené MAC adresami. Pokud mají síťové karty, které odesílají a přijímají data, adresy 00:ff:ac:ce:0b:de a 00:ff:ac:e0:dc:0f, představují tyto adresy koncové body komunikace (viz obrázek 5.1).

Konverzace v síti je obdobou rozhovoru mezi dvěma lidmi a označuje komunikaci, která probíhá mezi dvěma hostiteli (koncovými body). Konverzace Honzy a Petry může vypadat třeba takto: „Ahoj, jak se máš?“ „Ále, mizerně. A ty?“ a „Škoda mluvit!“ Konverzace mezi adresami 192.168.1.5 a 192.168.0.8 by pak mohla mít tuto

podobu: „SYN“, „SYN/ACK“ a „ACK“. (Proces komunikace protokolu TCP/IP podrobněji popíšeme v kapitole 6.)



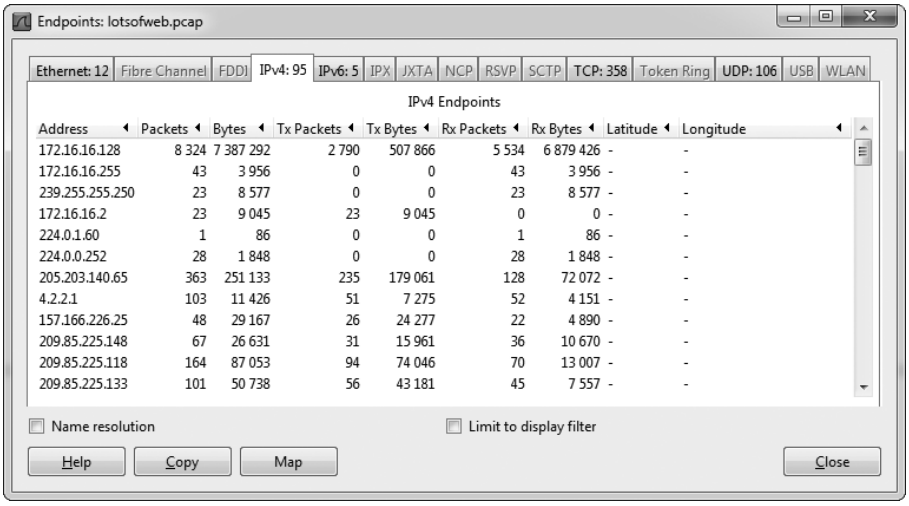
Obrázek 5.1: Koncové body v síti

Zobrazení koncových bodů

Při analýze síťového provozu se často ukazuje, že lze problém lokalizovat do určitého koncového bodu v síti. Okno Endpoints (Koncové body) programu Wireshark (**Statistics → Endpoints**) zobrazuje několik užitečných statistik o každém koncovém bodě (viz obrázek 5.2), včetně adres a počtu paketů a bajtů odeslaných a přijatých jednotlivými koncovými body.

Karty v horní části okna představují všechny podporované a rozpoznané koncové body v aktuálním zachyceném souboru. Chcete-li seznam koncových bodů zúžit na určité protokoly, klepněte na kartu. V okně Endpoints zaškrtněte políčko Name resolution (Překlad názvů).

V okně Endpoints lze odfiltrovat konkrétní pakety, které se zobrazují v podokně Packet List. Po klepnutí pravým tlačítkem myši na určitý koncový bod se zobrazí několik příkazů včetně možnosti vytvořit filtr, který zobrazí pouze provoz související s daným koncovým bodem nebo veškerý provoz mimo vybraný koncový bod. Koncový bod lze také exportovat přímo do pravidla barevného kódování (těmito pravidly jsme se zabývali v kapitole 3).

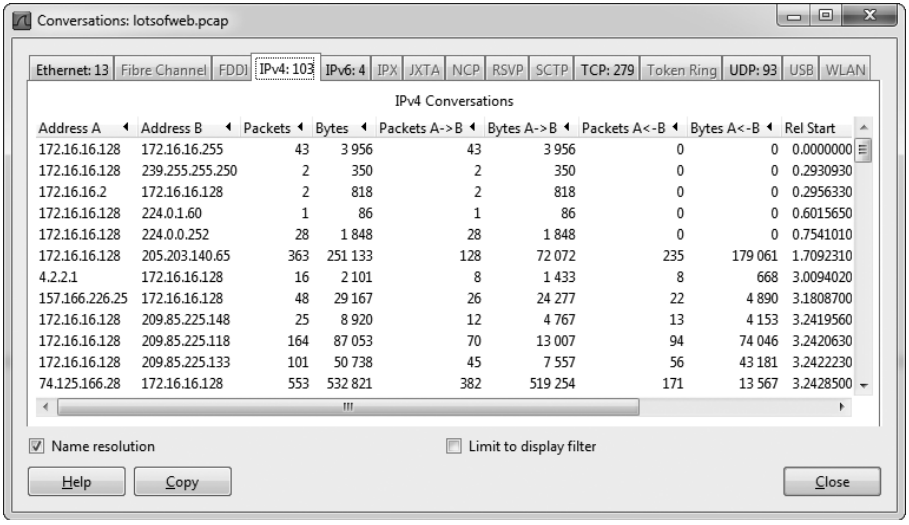


Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Latitude	Longitude
172.16.16.128	8 324	7 387 292	2 790	507 866	5 534	6 879 426	-	-
172.16.16.255	43	3 956	0	0	43	3 956	-	-
239.255.255.250	23	8 577	0	0	23	8 577	-	-
172.16.16.2	23	9 045	23	9 045	0	0	-	-
224.0.1.60	1	86	0	0	1	86	-	-
224.0.0.252	28	1 848	0	0	28	1 848	-	-
205.203.140.65	363	251 133	235	179 061	128	72 072	-	-
4.2.2.1	103	11 426	51	7 275	52	4 151	-	-
157.166.226.25	48	29 167	26	24 277	22	4 890	-	-
209.85.225.148	67	26 631	31	15 961	36	10 670	-	-
209.85.225.118	164	87 053	94	74 046	70	13 007	-	-
209.85.225.133	101	50 738	56	43 181	45	7 557	-	-

Obrázek 5.2: Okno Endpoints umožňuje zobrazit jednotlivé koncové body v zachyceném souboru

Zobrazení síťových konverzací

Okno Conversations (Konverzace) programu Wireshark (**Statistics → Conversations**), které je znázorněno na obrázku 5.3, zobrazuje adresy koncových bodů účastnících se na konverzaci. Tyto adresy jsou uvedeny ve sloupcích *Address A a Address B* vedle počtu paketů a bajtů přenesených jednotlivými zařízeními v obou směrech.



Address A	Address B	Packets	Bytes	Packets A->B	Bytes A->B	Packets A<-B	Bytes A<-B	Rel Start
172.16.16.128	172.16.16.255	43	3 956	43	3 956	0	0	0.0000000
172.16.16.128	239.255.255.250	2	350	2	350	0	0	0.2930930
172.16.16.2	172.16.16.128	2	818	2	818	0	0	0.2956330
172.16.16.128	224.0.1.60	1	86	1	86	0	0	0.6015650
172.16.16.128	224.0.0.252	28	1 848	28	1 848	0	0	0.7541010
172.16.16.128	205.203.140.65	363	251 133	128	72 072	235	179 061	1.7092310
4.2.2.1	172.16.16.128	16	2 101	8	1 433	8	668	3.0094020
157.166.226.25	172.16.16.128	48	29 167	26	24 277	22	4 890	3.1808700
172.16.16.128	209.85.225.148	25	8 920	12	4 767	13	4 153	3.2419560
172.16.16.128	209.85.225.118	164	87 053	70	13 007	94	74 046	3.2420630
172.16.16.128	209.85.225.133	101	50 738	45	7 557	56	43 181	3.2422230
74.125.166.28	172.16.16.128	553	532 821	382	519 254	171	13 567	3.2428500

Obrázek 5.3: Okno Conversations umožňuje zkoumat jednotlivé konverzace v zachyceném souboru

Konverzace uvedené v tomto okně se dělí podle použitého protokolu, který lze vybrat pomocí karet v horní části okna. Po klepnutí pravým tlačítkem myši na určitou konverzaci je možné vytvořit potenciálně užitečné filtry, například pro zobrazení veškerého provozu odeslaného ze zařízení A, veškerého provozu přijatého zařízením B nebo všech dat přenesených mezi zařízeními A a B.

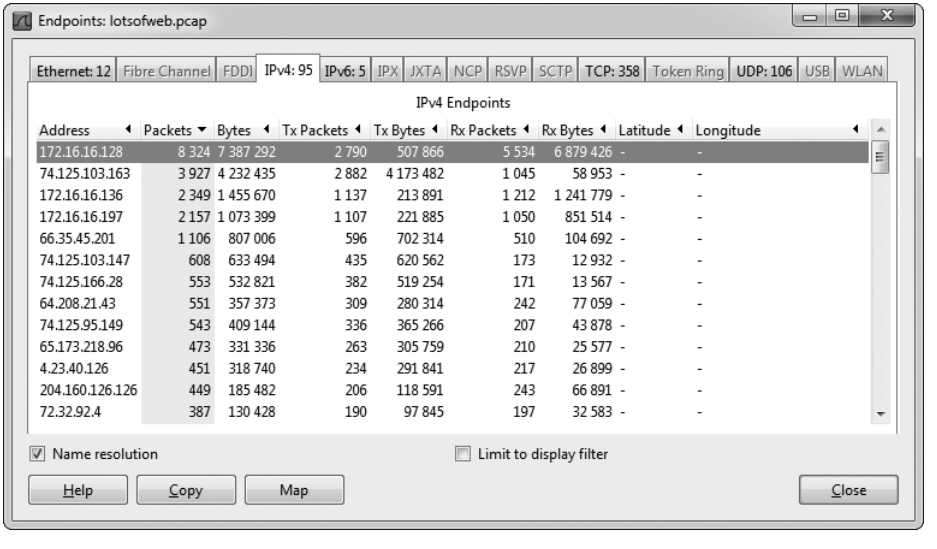
Řešení potíží pomocí oken Endpoints a Conversations

lotsofweb .pcap

Okna Endpoints a Conversations mají klíčovou roli při řešení potíží se sítěmi, zejména v situacích, kdy je potřeba najít zařízení zaplavující síť daty nebo určit, který ze serverů komunikuje nejvíce.

Otevřete-li například soubor *lotsofweb.pcap*, uvidíte velký objem dat protokolu HTTP, který generuje více klientů při procházení Internetu. Začnete-li tím, že otevřete okno Endpoints, můžete si o zobrazeném provozu okamžitě udělat určité závěry.

Při pohledu na kartu IPv4 (viz obrázek 5.4) je zřejmé, že první adresa seřazená podle počtu bajtů je lokální adresa 172.16.16.128. To znamená, že příslušné zařízení v lokální síti odpovídá za největší podíl komunikace v rámci datové sady (tzn. jedná se o hostitele, který nejvíce komunikuje). Druhá adresa 74.125.103.163 není místní. V této fázi tedy můžete předpokládat, že jistý klient v lokální síti s touto IP adresou komunikuje intenzivně, nebo se na komunikaci přiměřeně podílí více klientů. Rych-



Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Latitude	Longitude
172.16.16.128	8 324	7 387 292	2 790	507 866	5 534	6 879 426	-	-
74.125.103.163	3 927	4 232 435	2 882	4 173 482	1 045	58 953	-	-
172.16.16.136	2 349	1 455 670	1 137	213 891	1 212	1 241 779	-	-
172.16.16.197	2 157	1 073 399	1 107	221 885	1 050	851 514	-	-
66.35.45.201	1 106	807 006	596	702 314	510	104 692	-	-
74.125.103.147	608	633 494	435	620 562	173	12 932	-	-
74.125.166.28	553	532 821	382	519 254	171	13 567	-	-
64.208.21.43	551	357 373	309	280 314	242	77 059	-	-
74.125.95.149	543	409 144	336	365 266	207	43 878	-	-
65.173.218.96	473	331 336	263	305 759	210	25 577	-	-
4.23.40.126	451	318 740	234	291 841	217	26 899	-	-
204.160.126.126	449	185 482	206	118 591	243	66 891	-	-
72.32.92.4	387	130 428	190	97 845	197	32 583	-	-

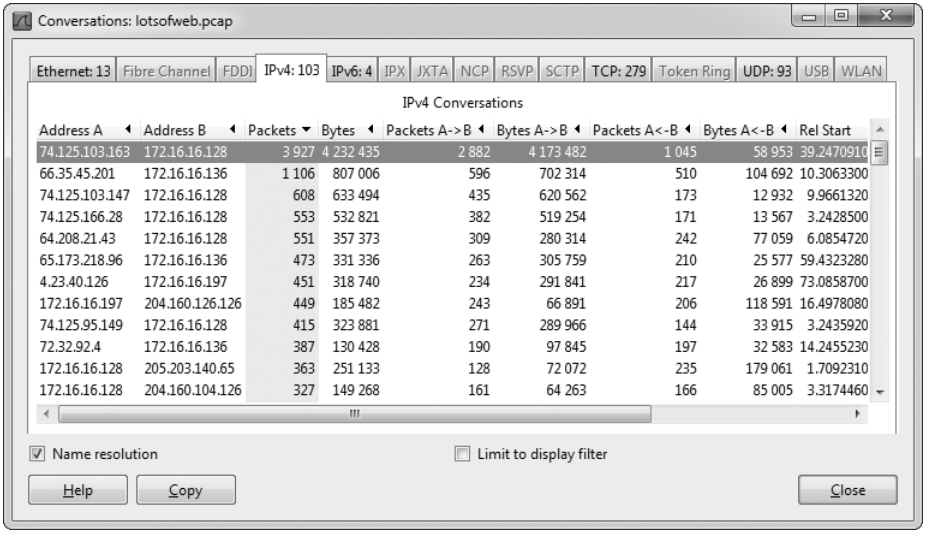
Obrázek 5.4: Okno Endpoints informuje o tom, kteří hostitelé komunikují nejvíce

lé hledání WHOIS (`http://whois.arin.net/ui/`) prozradí, že tato IP adresa patří společnosti Google a z analýzy paketů vyplývá, že jde o provoz služby YouTube.

POZNÁMKA

Za přiřazování IP adres odpovídají různé entity podle geografického umístění. V tomto příkladu používáme americký registr ARIN (American Registry for Internet Numbers), který zajišťuje přidělování IP adres v USA a některých okolních oblastech. Test WHOIS pro IP adresu se zpravidla provádí na webu organizace, která danou IP adresu přidělila. Jestliže geografickou oblast neznáte a zkusíte vyhledat v databázi nesprávného registru, budete přesměrováni do správného umístění. K dalším adresním registrům tohoto typu patří AfriNIC (Afrika), RIPE (Evropa) a APNIC (Asie a Tichomoří).

Dá se na základě těchto informací bezpečně tvrdit, že nejvíce komunikující koncové body současně generují nejobjemnější konverzaci? Když nyní otevřete okno Conversations a přejdete na kartu IPv4, můžete si to skutečně ověřit tím, že seznam seřadíte podle počtu bajtů. V tomto zobrazení je patrné, že provoz odpovídá stahování videa, protože počet dat přenesených z adresy A (74.125.103.163) je mnohem větší než počet bajtů odeslaných z adresy B (172.16.16.128) – viz obrázek 5.5.



Address A	Address B	Packets	Bytes	Packets A->B	Bytes A->B	Packets A<-B	Bytes A<-B	Rel Start
74.125.103.163	172.16.16.128	3 927	4 232 435	2 882	4 173 482	1 045	58 953	39.2470910
66.35.45.201	172.16.16.136	1 106	807 006	596	702 314	510	104 692	10.3063300
74.125.103.147	172.16.16.128	608	633 494	435	620 514	173	12 932	9.9661320
74.125.166.28	172.16.16.128	553	532 821	382	519 254	171	13 567	3.2428500
64.208.21.43	172.16.16.128	551	357 373	309	280 314	242	77 059	6.0854720
65.173.218.96	172.16.16.136	473	331 336	263	305 759	210	25 577	59.4323280
4.23.40.126	172.16.16.197	451	318 740	234	291 841	217	26 899	73.0858700
172.16.16.197	204.160.126.126	449	185 482	243	66 891	206	118 591	16.4978080
74.125.95.149	172.16.16.128	415	323 881	271	289 966	144	33 915	3.2435920
72.32.92.4	172.16.16.136	387	130 428	190	97 845	197	32 583	14.2455230
172.16.16.128	205.203.140.65	363	251 133	128	72 072	235	179 061	1.7092310
172.16.16.128	204.160.104.126	327	149 268	161	64 263	166	85 005	3.3174460

Obrázek 5.5: Okno Conversations potvrzuje, že dva nejaktivnější počítače komunikují spolu

Praktické scénáře použití oken Endpoints a Conversations naleznete v dalších částech této knihy.

Okno Protocol Hierarchy Statistics

lotsofweb
.pcap

Při zpracování mimořádně velkých zachycených souborů je někdy nutné zjistit rozložení protokolů v souboru – tzn. jaké procento zachycených dat tvoří protokoly TCP, IP, DHCP atd. Místo ručního počítání jednotlivých paketů a sčítání výsledků můžete v programu Wireshark otevřít okno Protocol Hierarchy Statistics (Statistiky hierarchie protokolů), které s výhodou umožňuje změřit parametry sítě. Pokud například víte, že protokol ARP obvykle odpovídá za 10 procent provozu ve vaší síti, a jednoho dne ze zachyceného souboru vyplývá, že provoz ARP tvoří 50 procent všech síťových přenosů, víte, že něco asi není v pořádku.

Ponechejte soubor *lotsofweb.pcap* načtený a otevřete okno Protocol Hierarchy Statistics (zobrazené na obrázku 5.6) příkazem **Statistics → Protocol Hierarchy** (Statistiky → Hierarchie protokolů). Všimněte si, že některé součty nedávají přesně 100 procent. Mnoho paketů totiž obsahuje více protokolů z různých vrstev, takže se statistiky protokolů v porovnání s jednotlivými pakety mohou lišit. Přesto je však k dispozici přesný pohled na distribuci protokolů v zachyceném souboru.

Wireshark: Protocol Hierarchy Statistics

Display filter: none

Protocol	% Packets	Packets	Bytes	Mbit/s	End Packets	End Bytes	End Mbit/s
Frame	100.00 %	12899	9931436	0.847	0	0	0.000
Ethernet	100.00 %	12899	9931436	0.847	0	0	0.000
Internet Protocol	99.71 %	12861	9926164	0.847	0	0	0.000
User Datagram Protocol	1.66 %	214	28932	0.002	0	0	0.000
NetBIOS Name Service	0.33 %	43	3956	0.000	43	3956	0.000
Hypertext Transfer Protocol	0.19 %	25	9395	0.001	25	9395	0.001
Service Location Protocol	0.01 %	1	86	0.000	1	86	0.000
Domain Name Service	1.08 %	139	14335	0.001	139	14335	0.001
Simple Network Management Protocol	0.03 %	4	476	0.000	4	476	0.000
Bootstrap Protocol	0.02 %	2	684	0.000	2	684	0.000
Transmission Control Protocol	98.03 %	12645	9897140	0.845	10867	8888570	0.759
Hypertext Transfer Protocol	13.78 %	1777	1008467	0.086	1406	749881	0.064
Line-based text data	1.17 %	151	107366	0.009	151	107366	0.009
Media Type	0.22 %	29	19543	0.002	29	19543	0.002
Compuserve GIF	0.75 %	97	55941	0.005	97	55941	0.005
JPEG File Interchange Format	0.52 %	67	58947	0.005	67	58947	0.005
Portable Network Graphics	0.12 %	16	10843	0.001	14	9511	0.001
Malformed Packet	0.02 %	2	1332	0.000	2	1332	0.000
eXtensible Markup Language	0.09 %	11	5946	0.001	11	5946	0.001
Secure Socket Layer	0.01 %	1	103	0.000	1	103	0.000
Internet Group Management Protocol	0.02 %	2	92	0.000	2	92	0.000
Internet Protocol Version 6	0.25 %	32	5020	0.000	0	0	0.000
User Datagram Protocol	0.25 %	32	5020	0.000	0	0	0.000
Domain Name Service	0.22 %	28	2408	0.000	28	2408	0.000
Data	0.02 %	2	2304	0.000	2	2304	0.000
DHCPv6	0.02 %	2	308	0.000	2	308	0.000
Address Resolution Protocol	0.05 %	6	252	0.000	6	252	0.000

Help Close

Obrázek 5.6: Okno Protocol Hierarchy Statistics zobrazuje distribuci různých protokolů

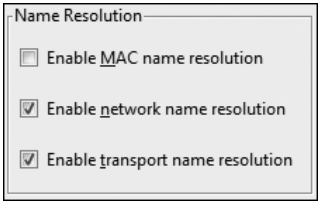
Okno Protocol Hierarchy Statistics často patří mezi první okna použitá při analýze provozu. Poskytuje velmi dobrou představu o typu aktivity, která se v síti objevuje. Když při analýze provozu shromáždíte více zkušeností, získáte schopnost zhodnotit uživatele a zařízení v síti pouhým pohledem na distribuci použitých protokolů. Často stačí podívat se na provoz ze síťového segmentu a ihned lze určit, že síťový segment patří oddělení IT – podle výskytu protokolů pro správu jako ICMP či SNMP. Provoz oddělení, které vyřizuje objednávky, se zase vyznačuje velkým podílem provozu poštovního protokolu SMTP. Drzého praktikanta pak odhalí provoz hry *World of Warcraft*.

Překlad názvů

Při přenosu síťových dat se uplatňuje mnoho alfanumerických adresních systémů, jejichž adresy jsou často příliš dlouhé nebo těžko zapamatovatelné. Platí to například o fyzických hardwarových adresách typu 00:16:CE:6E:8B:24. Pomocí procesu *překladu názvů* (name resolution), který se také označuje jako *vyhledávání názvů* (name lookup), protokol převádí jednu identifikující adresu na jinou. Počítač může mít například fyzickou MAC adresu 00:16:CE:6E:8B:24, ale protokoly DNS a ARP dovolují zjistit, že se nazývá *Marketing-2.domain.com*. Díky tomu, že lze původním kryptickým adresám přidělit snadno čitelné názvy, je možné počítače jednoduše identifikovat.

Povolení překladu názvů

Chcete-li zapnout překlad názvů, otevřete dialogové okno Capture Options (Možnosti zachytávání) příkazem **Capture → Options** (Zachytávat → Možnosti). Jak je patrné na obrázku 5.7, poskytuje program Wireshark tři typy překladu názvů:



Obrázek 5.7: Povolení překladu názvů v dialogovém okně Capture Options

- **MAC name resolution** (Překlad názvů MAC) – tento typ překladu názvů pomocí protokolu ARP převádí MAC adresy vrstvy 2, jako např. 00:09:5B:01:02:03, na adresy vrstvy 3, jako např. 10.100.12.1. Jestliže se pokus o převod nezdaří, program Wireshark se pokusí o převod pomocí souboru *ethers* ve svém programovém adresáři. Pokud neuspěje ani v tomto případě, program Wireshark nakonec konvertuje první 3 bajty MAC adresy do názvu výrobce zařízení podle specifikací IEEE, např. *Netgear_01:02:03*.
- **Network name resolution** (Překlad názvů síťové vrstvy) – tento typ překladu názvů převádí adresu vrstvy 3, jako např. IP adresu 192.168.1.50, na snadno čitelný název DNS, jako např. *MarketingPC1.domena.com*.

- **Transport name resolution** (Překlad názvů transportní vrstvy) – tento typ překladu názvů převádí číslo portu na přidružený název. Příkladem může být zobrazení portu 80 jako *http*.

Využitím různých nástrojů pro překlad názvů lze zlepšit čitelnost zachycených souborů a v některých situacích také ušetřit hodně času. Překlad názvů DNS například pomáhá snadno identifikovat název počítače, který je zdrojem určitého paketu.

Potenciální nevýhody překladu názvů

Překlad názvů nabízí mnoho výhod, a proto se zdá, že bychom jej mohli nasazovat automaticky. V praxi však přináší i některé potenciální nevýhody, například:

- Překlad názvů nemusí být úspěšný. Bývá to zpravidla způsobeno tím, že názvový server, kterému byl dotaz odeslán, příslušný název nezná.
- Překlad názvů musí proběhnout při každém otevření konkrétního zachyceného souboru, protože příslušné informace se do souboru neukládají. V případě nedostupnosti serverů, na nichž překlad názvů závisí, proto překlad názvů nebude úspěšný.
- Závislost na službě DNS může způsobit generování dalších paketů. Výsledný provoz překladu všech adres založených na systému DNS proto poznamená zachycený soubor. Platí obecné pravidlo, že při analýze určitého problému byste po lince neměli přenášet vlastní data.
- Překlad názvů vyžaduje dodatečnou režii zpracování. Pokud pracujete s velmi velkým zachyceným souborem a nemáte dostatek operační paměti, může být vhodné funkci překladu názvů vypnout, abyste šetřili systémové prostředky.

Rozbor protokolů

Disektor protokolu (protocol dissector) umožňuje programu Wireshark rozložit protokol na různé sekce, aby jej bylo možné analyzovat. Disektor protokolu ICMP v programu Wireshark například načítá nezpracovaná data linky a formátuje je do paketů ICMP.

Disektor lze přirovnat k překladateli, který převádí neformátovaná data přenášená po lince do podoby, kterou vyžaduje program Wireshark. Jestliže má program Wireshark podporovat určitý protokol, musí mít zabudovaný příslušný disektor (případně můžete napsat vlastní disektor v jazyku C nebo Python).

Program Wireshark používá při interpretaci každého paketu několik disektorů současně. Při výběru použitých disektorů používá naprogramovanou logiku a řídí se kvalifikovaným odhadem.

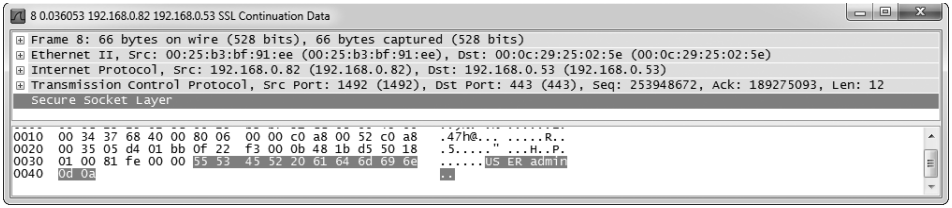
Změna disektoru

wrongdis-
sector
.pcap

Při výběru disektorů použitých pro určitý paket se program Wireshark bohužel občas dopouští chyb. Platí to zejména v případech, kdy se používá síťový protokol v nestandardní konfiguraci, například na jiném než výchozím portu (nastavení portů často mění správci sítí kvůli zvýšení bezpečnosti a zaměstnanci zase proto, aby obešli kontroly přístupu). Způsob implementace určitých disektorů v programu Wireshark lze naštěstí změnit.

Otevřete například zachycený soubor *wrongdissector.pcap*. Všimněte si, že tento soubor obsahuje komunikaci SSL mezi dvěma počítači. Protokol SSL (Secure Socket Layer) poskytuje zabezpečenou šifrovanou komunikaci mezi hostiteli. Za normálních okolností neposkytuje zobrazení provozu SSL v programu Wireshark příliš užitečných informací, což vyplývá z šifrované povahy protokolu. Zde však evidentně něco není v pořádku. Když klepnete na několik z těchto paketů a prozkoumáte jejich obsah v podokně Packet Bytes (Bajty paketů), najdete v nich nešifrovaná data. V paketu 4 je například uveden název serverové aplikace FTP FileZilla. Dalších několik paketů zjevně obsahuje požadavek a odpověď s uživatelským jménem i heslem.

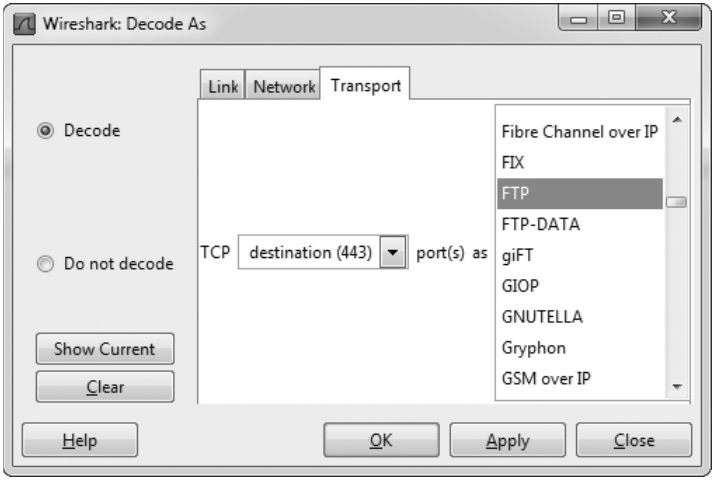
Pokud by se skutečně jednalo o provoz protokolu SSL, nemohli byste žádná data obsažená v paketech přečíst a rozhodně byste neviděli všechna nešifrovaná uživatelská jména a hesla (viz obrázek 5.8). Vzhledem k dostupným informacím lze s jistotou předpokládat, že jde o provoz protokolu FTP, a nikoli SSL. Nejspíše je to způsobeno tím, že provoz FTP využívá port 443, který je standardně určen pro protokol HTTPS (HTTP nad SSL).



Obrázek 5.8: Nešifrovaná uživatelská jména a hesla? To vypadá spíše na protokol FTP než SSL!

Chcete-li tento problém vyřešit, můžete program Wireshark přinutit, aby pro tyto pakety použil disektor protokolu FTP. Tento postup se označuje jako *nucené dekódování* (forced decode). Chcete-li to provést, postupujte takto:

1. Klepněte pravým tlačítkem myši na jeden z paketů SSL a vyberte příkaz **Decode As** (Dekódovat jako). Zobrazí se dialogové okno, kde můžete vybrat požadovaný disektor.
2. Pokud chcete programu Wireshark sdělit, aby veškerý provoz TCP se zdrojovým portem 443 dekódoval pomocí disektoru protokolu FTP, vyberte z rozevírací nabídky možnost **destination (443)** – cíl (443) – a ze seznamu na kartě Transport (Transportní vrstva) poté vyberte položku **FTP** (viz obrázek 5.9).



Obrázek 5.9: Dialogové okno Decode As umožňuje vytvořit nucená dekódování

3. Po výběru požadovaných možností klepněte na tlačítko **OK**. Změny se na zachycený soubor aplikují okamžitě.

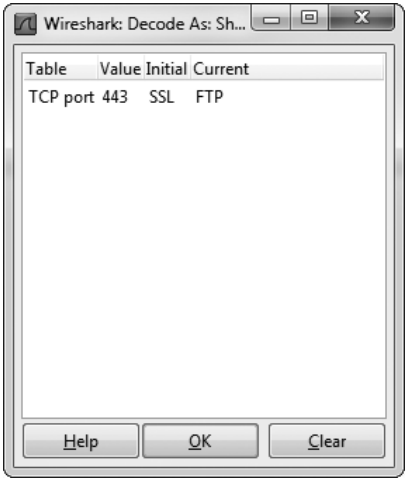
Data by měla být dekódována správně, aby je bylo možné analyzovat z podokna Packet List bez nutnosti procházení obsahu jednotlivých bajtů.

UPOZORNĚNÍ

Když vytváříte nucená dekódování, po uložení zachyceného souboru a ukončení programu Wireshark se provedené změny neuloží. Nucená dekódování je nutné znovu vytvořit při každém otevření zachyceného souboru.

Funkci nuceného dekódování lze v rámci stejného zachyceného souboru použít opakovaně. Vytvoříte-li v zachyceném souboru více nucených dekódování, můžete v nich snadno ztratit přehled. Program Wireshark však pomáhá i v tomto případě. V dialogovém okně Decode As můžete klepnutím na tlačítko Show Current (Zob-

razit aktuální) zkontrolovat všechna nucená dekódování, která jste dosud vytvořili (viz obrázek 5.10). Dekódování je také možné vymazat klepnutím na tlačítko Clear.



Obrázek 5.10: Po klepnutí na tlačítko Show Current se zobrazí všechna nucená dekódování vytvořená pro zachycený soubor

Zobrazení zdrojového kódu disektoru

Aplikace typu open source jsou atraktivní tím, že pokud některé jejich funkci nerozumíte, můžete se podívat na zdrojový kód a přesně zjistit, proč se chová daným způsobem. Tato možnost se hodí zejména v případech, kdy je potřeba zjistit, proč byl určitý protokol nesprávně interpretován.

Zdrojový kód disektorů protokolů můžete prozkoumat přímo z webu programu Wireshark, když umístíte ukazatel myši na nabídku Develop (Vývoj) a klepnete na odkaz Browse the Code (Procházení kódu). Tento odkaz vede do repozitáře nástroje Subversion, kde si můžete prohlédnout kód aktuálního vydání programu Wireshark i kód předchozích verzí. Po klepnutí na složku *releases* získáte přístup ke všem oficiálním vydáním programu Wireshark (dokonce i programu Ethereal). Nejnovější vydání jsou přitom na konci seznamu. Když zvolíte vydání, které chcete analyzovat, najdete disektory protokolů v příslušné složce *epan/dissectors*. Disektory mají standardní názvy ve tvaru *packet-název_protokolu.c*.

Tyto soubory mohou být poměrně složité, ale jak se můžete přesvědčit, vycházejí vesměs ze standardní šablony a bývají velmi dobře komentovány. Chcete-li pochopit základní funkce jednotlivých disektorů, nemusíte být zkušeným programátorem v jazyku C. Jestliže byste rádi rozuměli tomu, jak program Wireshark funguje, nahlédněte alespoň na disektory některých jednodušších protokolů.

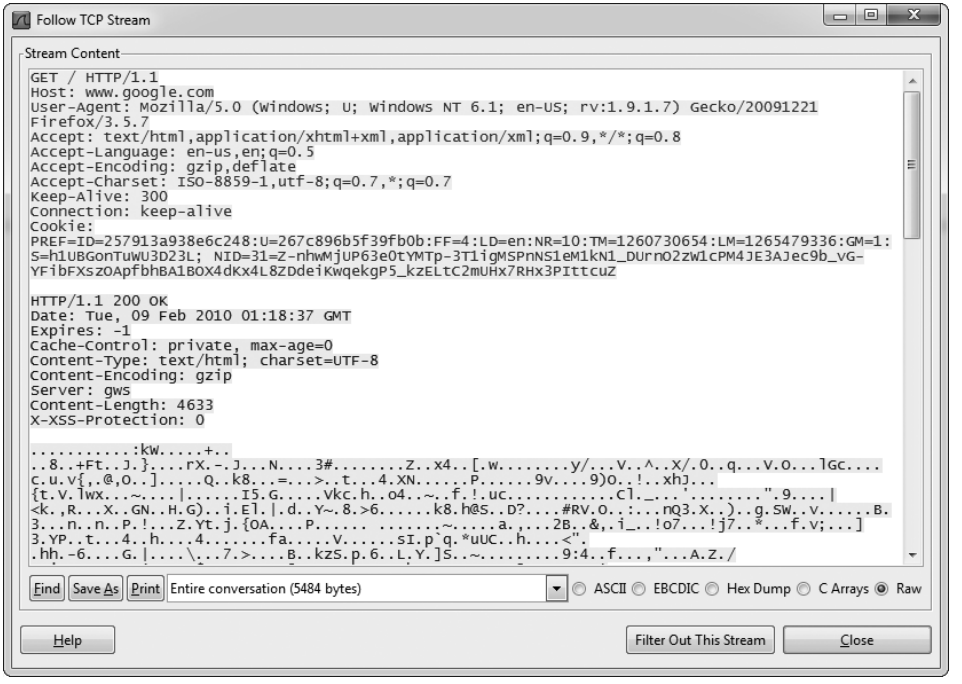
Sledování datových proudů TCP

http_google.pcap

K nejužitečnějším analytickým funkcím programu Wireshark patří možnost opakovaného sestavení datových proudů TCP do snadno čitelného formátu. Není nutné zobrazovat data odesílaná z klienta na server v mnoha malých dávkách, ale funkce Follow TCP Stream (Sledovat datový proud TCP) data seřadí a usnadní tak jejich zobrazení. Tato funkce se hodí při prohlížení textových protokolů aplikační

vrstvy, jako např. HTTP, FTP atd. (Principy těchto běžných protokolů se budeme podrobněji zabývat v další kapitole.)

Uvažme například jednoduchou transakci protokolu HTTP. Otevřete soubor *http_google.pcap*. Klepněte v souboru na libovolný paket TCP nebo HTTP, klepněte na soubor pravým tlačítkem myši a z místní nabídky zvolte příkaz **Follow TCP Stream**. Datový proud protokolu TCP se zobrazí v samostatném okně (viz obrázek 5.11).



Obrázek 5.11: Okno Follow TCP Stream znovu sestavuje komunikaci do snadno čitelného tvaru

Všimněte si, že text v tomto okně je zobrazen dvěma barvami. Červený text označuje provoz od zdroje k cíli a modrý text identifikuje provoz v opačném směru (od cíle ke zdroji). Barva odpovídá tomu, která strana inicializovala komunikaci. V tomto příkladu klient inicializoval připojení k webovému serveru, takže se má přidělenu červenou barvu.

Máte-li k dispozici tento datový proud TCP, můžete zřetelně sledovat většinu komunikace mezi oběma hostiteli. Komunikace začíná počátečním požadavkem `GET` na kořenový adresář webového serveru (`/`). Server odpoví, že požadavek byl úspěšný, pomocí zprávy `HTTP/1.1 200 OK`. Podobné schéma se v datovém prou-

du vícekrát opakuje, jak klient požaduje jednotlivé soubory a server reaguje jejich odesláním. Sledujete zde uživatele, který si prohlíží domovskou stránku Google. Vidíte přitom totéž co koncový uživatel, ale na jiné úrovni.

Kromě zobrazení nezpracovaných dat v tomto okně můžete také v textu vyhledávat, uložit jej do souboru, vytisknout jej nebo zvolit zobrazení dat v kódování ASCII či EBCDIC, hexadecimálním formátu nebo formátu pole C. Tyto možnosti jsou k dispozici v dolní části okna Follow TCP Stream.

Při analýze některých protokolů se bez sledování datových proudů TCP neobejdete.

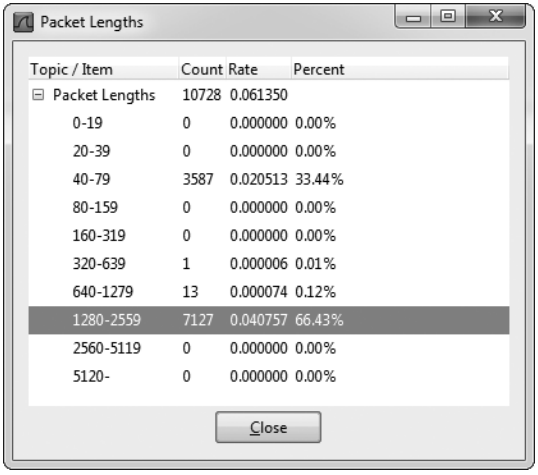
Okno Packet Lengths

download-slow.pcap

Velikost jednotlivého paketu nebo jejich skupiny může hodně prozrazovat o aktuální situaci. Rámec v síti Ethernet má za normálních okolností maximální velikost 1 518 bajtů. Když od tohoto čísla odečtete hlavičky protokolů Ethernet, IP a TCP, zůstane vám 1 460 bajtů, které lze využít k přenosu hlavičky protokolu vrstvy 7 nebo příslušných dat. Na základě této informace můžete začít zkoumat distribuci délek paketů v zachyceném souboru a kvalifikovaně odhadovat vlastnosti provozu.

Vhodný příklad naleznete po otevření souboru *download-slow.pcap*. Po otevření tohoto souboru vyberte příkaz **Statistics → Packet Lengths** (Statistiky → Délky paketů) a klepněte na možnost **Create Stat** (Vytvořit statistiku). Zobrazí se okno, které vidíte na obrázku 5.12.

V okně je zvýrazněn řádek se statistikou paketů s velikostí od 1 280 do 2 559 bajtů. Takové větší pakety obvykle svědčí o přenosu dat, zatímco menší pakety informují o výměně řídicích zpráv protokolů. V tomto případě si můžeme všimnout poměrně velkého podílu velkých paketů (66,43 procent). Na vlastní pakety v souboru se nemusíme ani podívat a můžeme usoudit, že zachycený soubor obsahuje jeden nebo více přenosů dat. Může se jednat o stahování protokolem HTTP, odesílání



Topic / Item	Count	Rate	Percent
Packet Lengths	10728	0.061350	
0-19	0	0.000000	0.00%
20-39	0	0.000000	0.00%
40-79	3587	0.020513	33.44%
80-159	0	0.000000	0.00%
160-319	0	0.000000	0.00%
320-639	1	0.000006	0.01%
640-1279	13	0.000074	0.12%
1280-2559	7127	0.040757	66.43%
2560-5119	0	0.000000	0.00%
5120-	0	0.000000	0.00%

Obrázek 5.12: Okno Packet Lengths pomáhá kvalifikovaně hodnotit provoz v zachyceném souboru

dat protokolem FTP nebo libovolný jiný typ síťové komunikace, kdy si hostitelé předávají data.

Většina zbývajících paketů (33,44 procent) patří do kategorie od 40 do 79 bajtů. Do této třídy obvykle spadají řídicí pakety TCP, které nenesou data. Vycházejme z typické velikosti hlaviček protokolů. Hlavička protokolu Ethernet má 14 bajtů (plus 4 bajty kontrolního součtu CRC), hlavička IP má alespoň 20 bajtů a paket TCP bez dat a dodatečných možností má rovněž 20 bajtů. To znamená, že standardní řídicí pakety TCP, např. pakety SYN, ACK, RST a FIN, budou mít velikost kolem 54 bajtů a budou náležet do této třídy. Doplňkové možnosti protokolů IP či TCP samozřejmě způsobí zvětšení paketů.

Informace o délkách paketů poskytuje ideální celkový pohled na zachycená data. Pokud zjistíme mnoho velkých paketů, můžeme bezpečně předpokládat, že probíhá přenos dat. Jestliže jsou pakety většinou malé, lze z toho odvodit, že zachycená data obsahují řídicí příkazy protokolů bez větších datových přenosů. Tato pravidla neplatí stoprocentně, ale před zahájením podrobnější analýzy lze někdy s těmito odhady vystačit.

Vytváření grafů

Grafy jsou základem analýzy a umožňují získat přehled o sadě dat jako máloco jiného. Program Wireshark poskytuje několik různých funkcí pro vytváření grafů, které pomáhají porozumět zachyceným datům. První z těchto funkcí jsou vstupně-výstupní grafy.

Zobrazení vstupně-výstupních grafů

download--fast.pcap

download--slow.pcap

Okno IO Graphs (Vstupně-výstupní grafy) programu Wireshark umožňuje zobrazit propustnost dat v síti. Pomocí těchto grafů lze hledat výchylky v datové propustnosti, zjišťovat poklesy výkonu jednotlivých protokolů a porovnávat souběžné datové proudy.

Chcete-li zobrazit příklad vstupně-výstupního grafu počítače, který stahuje soubor z Internetu, otevřete soubor *download-fast.pcap*. Klepnutím vyberte libovolný paket TCP a poté zvolte příkaz **Statistics → IO Graphs** (Statistiky → Vstupně--výstupní grafy).

Okno IO Graphs graficky znázorňuje tok dat v časovém intervalu zachyceného souboru. Ukázkový graf na obrázku 5.13 představuje stahování průměrnou rychlostí 500 paketů na jeden impulz. Rychlost zůstává v čase poměrně konstantní a teprve ke konci klesá.