KAPITOLA 2

Představujeme Wireshark: analyzátor síťových protokolů

Řešení v této kapitole:

- Co je Wireshark?
- Podpůrné programy
- Používání Wiresharku ve vaší síťové architektuře
- Používání Wiresharku pro řešení problémů se sítí
- Používání Wiresharku pro administraci systému
- Používání Wiresharku pro administraci zabezpečení
- Zabezpečení Wiresharku
- Optimalizace Wiresharku
- Pokročilé techniky zachytávání dat
- Ochrana sítě proti zachytávání dat
- Shrnutí
- Rychlá řešení
- Časté dotazy

Úvod

Možná jste si vybrali tuto knihu, protože jste už dříve slyšeli o aplikaci Wireshark (nebo jeho předchůdci, programu Ethereal) a jejím grafickém rozhraní nabitém funkcemi. Nebo jste o Wiresharku četli na Internetu, zaslechli kolegy hovořící o tomto programu nebo byl zmíněn na některé konferenci o počítačové bezpečnosti. Ať už byly vaše pohnutky jakékoliv, pokud hledáte srozumitelného průvodce, který vám pomůže odhalit potenciál Wiresharku, vybrali jste si správně.

Wireshark je tím nejlepším dostupným síťovým analyzátorem šířeným jako open-source program. Obsahuje funkce srovnatelné s těmi, které jsou součástí komerčních síťových analyzátorů, a díky velkému a různorodému kolektivu autorů je neustále vyvíjen.

Wireshark je stálou a užitečnou součástí všech síťových sad nástrojů a neustále jsou do něj přidávány nové funkce a opravovány případné chyby. Od počátku vývoje Wiresharku (od dob, kdy se ještě jmenoval Ethereal) byl učiněn velký pokrok; program nyní pracuje srovnatelně a v některých ohledech i lépe než komerční produkty pro zachytávání dat.

V této kapitole porozumíte programu Wireshark, naučíte se, jaké jsou jeho funkce a jak je použít pro řešení problémů v architektuře vaší sítě.

Navíc se dozvíte i něco z minulosti tohoto programu, zejména proč se stal tak oblíbeným síťovým analyzátorem a proč stále zůstává jasnou volbou administrátorů starajících se o zabezpečení sítě. Také se podíváme na některé tipy, jak provozovat Wireshark bezpečně a jak jej optimalizovat pro plynulý chod pokročilých technik.

Co je Wireshark?

Wireshark je síťový analyzátor. Umí číst pakety ze sítě, dekódovat je a zobrazit ve srozumitelném formátu. Jednou z důležitých vlastností Wiresharku je skutečnost, že je distribuován jako open-source, a tedy zdarma. Následuje výpis některých dalších vlastností tohoto programu:

- Wireshark je distribuován zdarma pod open-source licencí Gnu's not UNIX (GNU) General Public License (GPL).
- Pracuje v promiskuitním i nepromiskuitním módu.
- Může zachytávat data ze sítě nebo je číst ze souboru.
- Má srozumitelné a konfigurovatelné rozhraní.
- Má bohaté možnosti nastavení zobrazovacích filtrů.
- Podporuje formát souborů dat zachycených programem tcpdump. Také má nástroje pro rekonstrukci relace protokolu TCP a dokáže je zobrazit v kódu American Standard Code for Information Interchange (ASCII), Extended Binary Coded Decimal Interchange Code (EBCDIC), dále pak v hexadecimálním tvaru nebo ve formátu pole jazyka C.
- Je dostupný v podobě předkompilovaných binárních souborů i v podobě zdrojového kódu.
- Pracuje na více než 20 platformách, včetně operačních systémů založených na Uniplexed Information and Computing Systém (UNIX), Windows a od jiných dodavatelů jsou dostupné i instalační balíky pro operační systém Mac OS X.
- Podporuje více než 750 protokolů, a protože je open-source, nové protokoly jsou komunitou přidávány poměrně často.

- Dokáže číst data zachycená více než 25 odlišnými programy.
- Může ukládat zachycená data v různých formátech (jako libpcap, Network Associated Sniffer, Microsoft Network Monitor (NetMon) a snoop z operačního systému Sun).
- Umí zachytávat data z různých přenosových médií (jako Ethernet, Token-Ring, bezdrátové protokoly 802.11 a dalších).
- Obsahuje verzi programu ovládanou z příkazové řádky, nazvanou *tshark*.
- Obsahuje další podpůrné programy jako *editcap*, *mergecap* a *text2pcap*.
- Výstup může být uložen nebo vytištěn jako prostý text nebo PostScript.

Historie Wiresharku

V roce 1977 Gerald Combs vyvinul program Ethereal, aby byl schopen rozšířit své znalosti o počítačových sítích. K tomu potřeboval vhodný nástroj pro řešení problémů v síti. První verze (v0.2.0) byla vydána v červenci roku 1998. V krátké době se sešel tým vývojářů včetně Gilberta Ramireze, Guye Harrise a Richarda Sharpa, který pracoval na vývoji záplat, rozšíření stávajících funkcí a dalších disektorech. Disektory jsou tím, co umožňuje Wiresharku dekódovat jednotlivé protokoly a zobrazovat je v čitelné formě. Od počátku vývoje se na vývoji disektorů a dalších rozšíření podílelo velké množství jednotlivců. Seznam všech autorů můžete najít na *www.wireshark.org/about.html#authors*. Díky ohromné podpoře vývoje a velké uživatelské základně rostou schopnosti Wiresharku i jeho popularita každým dnem.

Další informace

Licence GNU GPL

Projekt GNU byl původně vyvinut v roce 1984, aby poskytl zdarma distribuovaný operační systém založený na systému UNIX. Někteří zastávají názor, že Linux jako operační systém by měl být nazýván „GNU/Linux“, protože používá nástroje GNU spolu s jádrem Linuxu.

Projekt GNU je provozován a sponzorován nadací Free Software Foundation (FSF). Richard Stallman vytvořil GNU GPL v roce 1989, aby bylo možné šířit programy vyvinuté v rámci projektu GNU. Jedná se o tzv. copyleft (tzn. Copyleft – všechna práva vyhrazena), zdarma poskytovaná licence k softwaru, která je založena na podobných licencích, které byly použity v dřívějších verzích GNU programu Emacs (MACroS).

Copyleft představuje použití autorského práva takovým způsobem, aby byla zajištěna veřejná svoboda pro změny, vylepšení a redistribuci práce autorské a jejich deriváty. To znamená, že držitel autorského práva zaručuje nezpochybnitelnou licenci všem příjemcům kopie tohoto programu, přičemž jim zaručuje souhlas s redistribucí i prodejem případně více modifikovaných kopií, avšak jen pod tou podmínkou, že všechny takto vzniklé kopie budou šířeny v souladu s platností licenčních ujednání GNU GPL a budou podporovat další modifikaci díla. V případě, že dojde k porušení licenčních ujednání, existují zde právní důsledky, se kterými se osoba porušující licenci bude muset vypořádat. Pokud někdo šíří kopie díla, je povinen šířit i zdrojový kód a úpravy, které provedl.

Znění licence GPL nesmí být modifikováno. Můžete ji kopírovat a šířit, ale je zakázáno měnit její obsah. Můžete použít znění licence GPL a upravit je podle vašich potřeb, ale potom už

se tato licence nesmí nazývat GPL. Další licence vytvořené v rámci projektu GNU zahrnují také GNU Lesser GPL a licenci GNU Free Documentation License.

Ve světě neustále probíhají spory o licenci GPL a o tom, zda softwarový produkt, který není šířen pod licencí GPL, může používat knihovny spadající pod licenci GPL. Ačkoliv programy odvozené od programů vyvinutých pod licencí GPL musí dodržovat pravidla GPL, není jasné, zda spustitelné soubory linkované s knihovnami GPL mohou být považovány za odvozené programy. Nadace Free Software Foundation tvrdí, že takovéto spustitelné soubory jsou deriváty děl spadajících pod licenci GPL, ale část vývojářské komunity s tím nesouhlasí. V době vydání této knihy ještě nebylo vydáno žádné soudní rozhodnutí, které by tento problém vyřešilo.

Kompatibilita

Jak už bylo uvedeno, Wireshark umí číst a zpracovávat soubory s daty zachycenými jinými programy, jako sniffery, směrovači a síťovými nástroji. Protože Wireshark využívá formát dat populární knihovny Promiscuous Capture Library (libpcap), je schopen spolupráce s dalšími aplikacemi, které rovněž pracují s daty knihovny libpcap. Wireshark umí číst také data mnoha různých jiných formátů. Je schopen automaticky rozeznat typ čteného souboru a může také rozbalit data komprimovaná programem GNU Zip (gzip). Následující seznam obsahuje výčet souborů, jejichž data je Wireshark schopen přečíst:

- Tcpdump
- Sun snoop a atmsnoop
- Microsoft NetMon
- Network Associates Sniffer (komprimovaná i nekomprimovaná data) a Sniffer Pro
- Shomiti/Finisar Surveyor
- Novell LANalyzer
- Cinco Networks NetXRay
- AG Group/WildPackets EtherPeek/TokenPeek/AiroPeek
- RADCOM's wide area network (WAN)/local area network (LAN) analyzer
- Visual Network's Visual UpTime
- Lucent/Ascent router debug output
- Výstup směrovače Toshiba's Integrated Services Digital Network (ISDN)
- Záznamy systému pro detekci průniku Cisco Secure
- Iptrace systému AIX
- HP-UX nettl
- Výstup i4btrace projektu ISDN4BSD
- Záznamy ve formátu pppdump programu Point-To-Point Protocol Daemon (PPPD)
- Nástroj TCPIPtrace systému VMS
- Nástroj DBS Etherwatch Virtual Memory Systém (VMS)
- CoSine L2 debug
- Výstup Accellent's 5Views LAN agenta

- Formát dat zachycených programem Endace Measurement System's Electronice Remote Fill (ERF)
- Záznamy z Linux Bluez Bluetooth stack "hcidump -w"
- Catapult DCT2000
- Network Instruments Observer verze 9
- Záznamy EyeSDN Universal Serial Bus (USB) S0

Podporované protokoly

Když síťový analyzátor čte data ze sítě, potřebuje vědět, jak tato data interpretovat a zobrazit ve srozumitelném formátu. Této interpretaci říkáme *dekódování protokolu*. Celková kvalita síťového analyzátoru je často dána počtem protokolů, které sniffer dokáže přečíst a zobrazit, takže většina komerčních snifferů dokáže pracovat až se stovkami různých protokolů. Wireshark je v této oblasti při své podpoře 750 protokolů velkou konkurencí. Kromě toho jsou další protokoly neustále přidávány množstvím lidí, kteří na projektu Wireshark spolupracují. Dekodéry protokolů, také známé jako disektory, mohou být přidávány přímo do kódu aplikace nebo formou zásuvných modulů (tzv. plug-inů). Následující seznam obsahuje výčet protokolů podporovaných v době psaní této knihy:

> 3COMXNS, 3GPP2 A11, 802.11 MGT, 802.11 Radiotap, 802.3 Slow protocols, 9P, AAL1, AAL3/4, AARP, ACAP, ACN, ACSE, ACtrace, ADP, AFP, AFS (RX), AgentX, AH, AIM, AIM Administration, AIM Advertisements, AIM BOS, AIM Buddylist, AIM Chat, AIM ChatNav, AIM Directory, AIM E-mail, AIM Generic, AIM ICQ, AIM Invitation, AIM Location, AIM Messaging, AIM OFT, AIM Popup, AIM Signon, AIM SSI, AIM SST, AIM Stats, AIM Translate, AIM User Lookup, AJP13, ALC, ALCAP, AMR, ANS, ANSI BSMAP, ANSI DTAP, ANSI IS-637-A Teleservice, ANSI IS-637-A Transport, ANSI IS-683-A (OTA (Mobile)), ANSI IS--801 (Location Services (PLD)), ANSI MAP, AODV, AOE, ARCNET, Armagetronad, ARP/RARP, ARTNET, ASAP, ASF, ASN1, ASP, ATM, ATM LANE, ATP, ATSVC, Auto-RP, AVS WLANCAP, AX4000, BACapp, BACnet, Basic Format XID, BEEP, BER, BFD Control, BGP, BICC, BitTorrent, Boardwalk, BOFL, BOOTP/DHCP, BOOTPARAMS, BOSSVR, BROWSER, BSSAP, BSSGP, BUDB, BUTC, BVLC, CAMEL, CAST, CBAPDev, CCSDS, CCSRL, CDP, CDS_CLERK, cds_solicit, CDT, CFLOW, CGMP, CHDLC, CIGI, CIMD, CIP, CISCOWLL2, CLDAP, CLEARCASE, CLNP, CLTP, CMIP, CMP, CMS, CONV, COPS, COSEVENTCOMM, CoSine, COSNAMING, COTP, CPFI, CPHA, cprpc_server, CRMF, CSM_ENCAPS, CUPS, DAAP, DAP, Data, dc, DCCP, DCE_DFS, dce_update, DCERPC, DCOM, DCP, DDP, DDTP, DEC_DNA, DEC_STP, DFS, DHCPFO, DHCPv6, DIAMETER, dicom, DIS, DISP, DISTCC, DLSw, DLT User A, DLT User B, DLT User C, DLT User D, DNP 3.0, DNS, DNSSERVER, DOCSIS, DOCSIS BPKM-ATTR, DOCSIS BPKM-REQ, DOCSIS BPKM-RSP, DOCSIS DSA-ACK, DOCSIS DSAREQ, DOCSIS DSA-RSP, DOCSIS DSC-ACK, DOCSIS DSC--REQ, DOCSIS DSC-RSP, DOCSIS DSD-REQ, DOCSIS DSD-RSP, DOCSIS INT-RNG-REQ, DOCSIS MAC MGMT, DOCSIS MAP, DOCSIS REGACK, DOCSIS REG-REQ, DOCSIS REG-RSP, DOCSIS RNG-REQ, DOCSIS RNG-RSP, DOCSIS TLVs, DOCSIS type29ucd, DOCSIS UCCREQ, DOCSIS UCC-RSP, DOCSIS UCD, DOCSIS VSIF, DOP, DRSUAPI, DSI, DSP, DSSETUP, DTP, DTSPROVIDER, DTSSTIME_REQ, DUA, DVMRP, E.164, EAP, EAPOL, ECHO, EDONKEY, EDP, EFS, EIGRP, ENC, ENIP, ENRP, ENTTEC, EPM, EPMv4, ESIS, ESP, ESS, ETHERIC, ETHERIP, Ethernet, EVENTLOG, FC, FC ELS, FC FZS, FC-dNS, FC-FCS, FC-SB3, FC-SP, FC-SWILS, FC_CT, FCIP, FCP, FDDI, FIX, FLDB, FR, Frame, FRSAPI,

FRSRPC, FTAM, FTBP, FTP, FTP-DATA, FTSERVER, FW-1, G.723, GIF image, giFT, GIOP, GMRP, GNM, GNUTELLA, GPRS NS, GPRS-LLC, GRE, Gryphon, GSM BSSMAP, GSM DTAP, GSM RP, GSM SMS, GSM SMS UD, GSM_MAP, GSM_SS, GSS-API, GTP, GVRP, H.223, H.225.0, H.235, H.245, H.261, H.263, H.263 data, H1, h221nonstd, H248, h450, HCL-NFSD, HPEXT, HPSW, HSRP, HTTP, HyperSCSI, IAP, IAPP, IAX2, IB, ICAP, ICBAAccoCB, ICBAAccoCB2, ICBAAccoMgt, ICBAAccoMgt2, ICBAAccoServ, ICBAAccoServ2, ICBAAccoServSRT, ICBAAccoSync, ICBABrowse, ICBABrowse2, ICBAGErr, ICBAGErrEvent, ICBALDev, ICBALDev2, ICBAPDev, ICBAPDev2, ICBAPDevPC, ICBAPDevPCEvent, ICBAPersist, ICBAPersist2, ICBARTAuto, ICBARTAuto2, ICBAState, ICBAStateEvent, ICBASysProp, ICBATime, ICEP, ICL_RPC, ICMP, ICMPv6, ICP, ICQ, IDispatch, IDP, IEEE 802.11, IEEE802a, iFCP, IGAP, IGMP, IGRP, ILMI, IMAP, INAP, INITSHUTDOWN, IOXIDResolver, IP, IP/IEEE1394, IPComp, IPDC, IPFC, IPMI, IPP, IPv6, IPVS, IPX, IPX MSG, IPX RIP, IPX SAP, IPX WAN, IRC, IrCOMM, IRemUnknown, IRemUnknown2, IrLAP, IrLMP, ISAKMP, iSCSI, ISDN, ISIS, ISL, ISMP, iSNS, ISUP, isup_thin, ISystemActivator, itunes, IUA, IuUP, Jabber, JFIF (JPEG) image, Juniper, JXTA, JXTA Framing, JXTA Message, JXTA UDP, JXTA Welcome, K12xx, KADM5, KINK, KLM, Kpasswd, KRB4, KRB5, KRB5RPC, L2TP, LANMAN, LAPB, LAPBETHER, LAPD, Laplink, LDAP, LDP, Line-based text data, LLAP, llb, LLC, LLDP, LMI, LMP, Log, LogotypeCertExtn, LOOP, LPD, LSA, Lucent/Ascend, LWAPP, LWAPP-CNTL, LWAPP-L3, LWRES, M2PA, M2TP, M2UA, M3UA, MACC, Malformed packet, Manolito, MAP_DialoguePDU, MAPI, MDS Header, Media, MEGACO, message/http, Messenger, MGCP, MGMT, MIME multipart, MIPv6, MMS, MMSE, Mobile IP, Modbus/TCP, MOUNT, MPEG1, MPLS, MPLS Echo, MQ, MQ PCF, MRDISC, MS NLB, MS Proxy, MSDP, MSMMS, MSNIP, MSNMS, MSRP, MTP2, MTP3, MTP3MG, MySQL, NBAP, NBDS, NBIPX, NBNS, NBP, NBSS, NCP, NCS, NDMP, NDPS, NetBIOS, Netsync, nettl, NFS, NFSACL, NFSAUTH, NHRP, NIS+, NIS+ CB, NJACK, NLM, NLSP, NMAS, NMPI, NNTP, NORM, NS_CERT_EXTS, NSIP, NSPI, NTLMSSP, NTP, Null, NW_SERIAL, OAM AAL, OCSP, OLSR, OPSI, OSPF, P_MUL, PAGP, PAP, PARLAY, PCLI, PCNFSD, PER, PFLOG, PFLOG-OLD, PGM, PGSQL, PIM, PKCS-1, PKInit, PKIX Certificate, PKIX1EXPLICIT, PKIX1IMPLICIT, PKIXPROXY, PKIXQUALIFIED, PKIXTSP, PKTC, PNDCP, PN-RT, PNIO, PNP, POP, Portmap, PPP, PPP BACP, PPP BAP, PPP CBCP, PPP CCP, PPP CDPCP, PPP CHAP, PPP Comp, PPP IPCP, PPP IPV6CP, PPP LCP, PPP MP, PPP MPLSCP, PPP OSICP, PPP PAP, PPP PPPMux, PPP PPPMuxCP, PPP VJ, PPP-HDLC, PPPoED, PPPoES, PPTP, PRES, Prism, PTP, PVFS, Q.2931, Q.931, Q.933, QLLC, QUAKE, QUAKE2, QUAKE3, QUAKEWORLD, R-STP, RADIUS, RANAP, Raw, Raw_SigComp, Raw_SIP, rdaclif, RDM, RDT, Redback, REMACT, REP_PROC, RIP, RIPng, RLM, Rlogin, RMCP, RMI, RMP, RNSAP, ROS, roverride, RPC, RPC_BROWSER, RPC_NETLOGON, RPL, rpriv, RQUOTA, RRAS, RS_ACCT, RS_ATTR, rs_attr_schema, RS_BIND, rs_misc, RS_PGO, RS_PLCY, rs_prop_acct, rs_prop_acl, rs_prop_attr, rs_prop_pgo, rs_prop_plcy, rs_pwd_mgmt, RS_REPADM, RS_REPLIST, rs_repmgr, RS_UNIX, rsec_login, RSH, rss, RSTAT, RSVP, RSYNC, RTcfg, RTCP, RTmac, RTMP, RTP, RTP Event, RTPS, RTSE, RTSP, RUDP, RWALL, RX, SADMIND, SAMR, SAP, SCCP, SCCPMG, SCSI, SCTP, SDLC, SDP, SEBEK, SECIDMAP, Serialization, SES, sFlow, SGI MOUNT, Short frame, SIGCOMP, SIP, SIPFRAG, SIR, SKINNY, SLARP, SliMP3, SLL, SM, SMB, SMB Mailslot, SMB Pipe, SMB2, SMB_NETLOGON, smil, SMPP, SMRSE, SMTP, SMUX, SNA, SNA XID, SNAETH, SNDCP, SNMP, Socks, SONMP, SoulSeek, SPNEGO, SPNEGO-KRB5, SPOOLSS, SPP, SPRAY, SPX, SRP, SRVLOC, SRVSVC, SSCF-NNI, SSCOP, SSH, SSL, SSS, STANAG 4406, STANAG 5066,

STAT, STAT-CB, STP, STUN, SUA, SVCCTL, Symantec, Synergy, Syslog, T.38, TACACS, TACACS+, TALI, TANGO, TAPI, TCAP, TCP, TDMA, TDS, TEI_MANAGEMENT, TELNET, Teredo, TFTP, TIME, TIPC, TKN4Int, TNS, Token-Ring, TPCP, TPKT, TR MAC, TRKSVR, TSP, TTP, TUXEDO, TZSP, UBIKDISK, UBIKVOTE, UCP, UDP, UDPENCAP, UDPlite, UMA, Unreassembled fragmented packet, V.120, V5UA, Vines ARP, Vines Echo, Vines FRP, Vines ICP, Vines IP, Vines IPC, Vines LLC, Vines RTP, Vines SPP, VLAN, VNC, VRRP, VTP, WAP SIR, WBXML, WCCP, WCP, WHDLC, WHO, WINREG, WINS-Replication, WKSSVC, WLANCERTEXTN, WSP, WTLS, WTP, X.25, X.29, X11, X411, X420, X509AF, X509CE, X509IF, X509SAT, XDMCP, XML, XOT, XYPLEX, YHOO, YMSG, YPBIND, YPPASSWD, YPSERV, YPXFRZEBRA, ZIP

Uživatelské rozhraní programu Wireshark

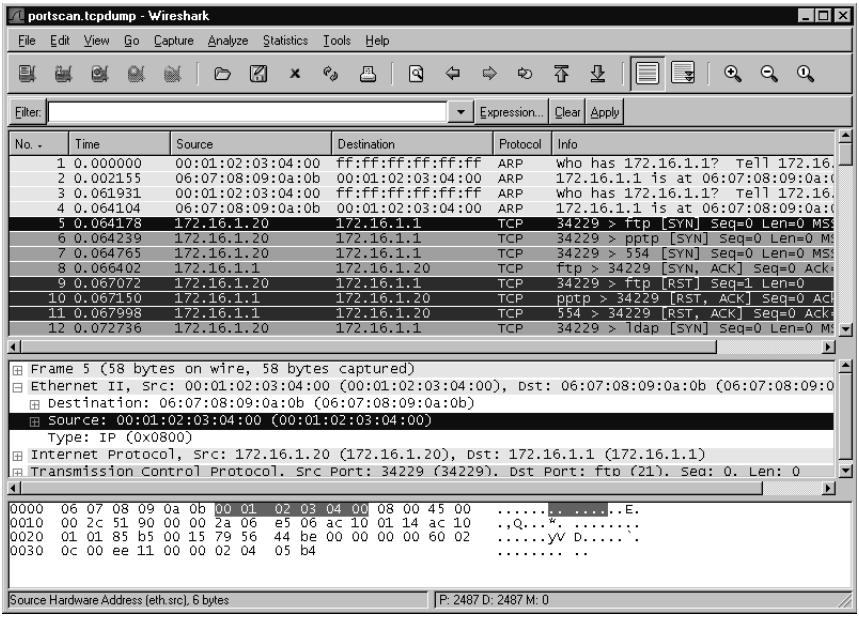
Grafické uživatelské rozhraní programu Wireshark je konfigurovatelné a jeho použití je snadné. A stejně jako jiné síťové analyzátory, i Wireshark zobrazuje zachycená data ve 3 hlavních panelech. Obrázek 2.1 ukazuje, jak vypadají typická data zachycená Wiresharkem. Každé okno je nastavitelné klepnutím na řadu teček mezi okny panelů a jejich tažením nahoru či dolů. Nejvýše je umístěn panel *summary (souhrn)*, který zobrazuje jednořádkový souhrn zachycených dat. Výchozí pole Wiresharku obsahují:

- Číslo paketu
- Čas
- Zdrojovou adresu
- Cílovou adresu
- Název a informace o protokolu vyšší vrstvy

Tyto sloupce mohou být snadno nastaveny a další můžete přidat v nabídce Preferences. Sloupce můžete také sestupně nebo vzestupně setřídit podle pole a také můžete změnit rozvržení panelů.

Poznámka: Všimněte si, že grafické uživatelské rozhraní Wiresharku je podobné Unixovým aplikacím více než programům, které známe z operačního systému Windows. Je to dáno tím, že Wireshark používá sadu nástrojů GNU Image Manipulation Program (GIMP) – knihovny GTK pro tvorbu uživatelského rozhraní. A tak Wireshark vypadá téměř identicky na všech operačních systémech.

Střední panel je panel *protokolu* a formou stromové struktury poskytuje detaily o každé vrstvě obsažené v zachyceném paketu. Klepnutím na různé části hierarchie protokolů zvýrazníte odpovídající výstup v hexadecimálním a ASCII tvaru ve spodním panelu, přičemž spodní panel zobrazuje surová zachycená data jak v hexadecimálním, tak i ASCII formátu. Klepnutím na různé části těchto dat zvýrazníte odpovídající pole v hierarchii protokolů a v panelu pro zobrazení protokolu. Na obrázku 2.1 můžete vidět rozhraní programu Wireshark a příklad zachycené síťové synchronizace. Všimněte si, že označením adresy MAC ve středním panelu je automaticky zvýrazněna i část obsahující hexadecimální výpis v dolním panelu.

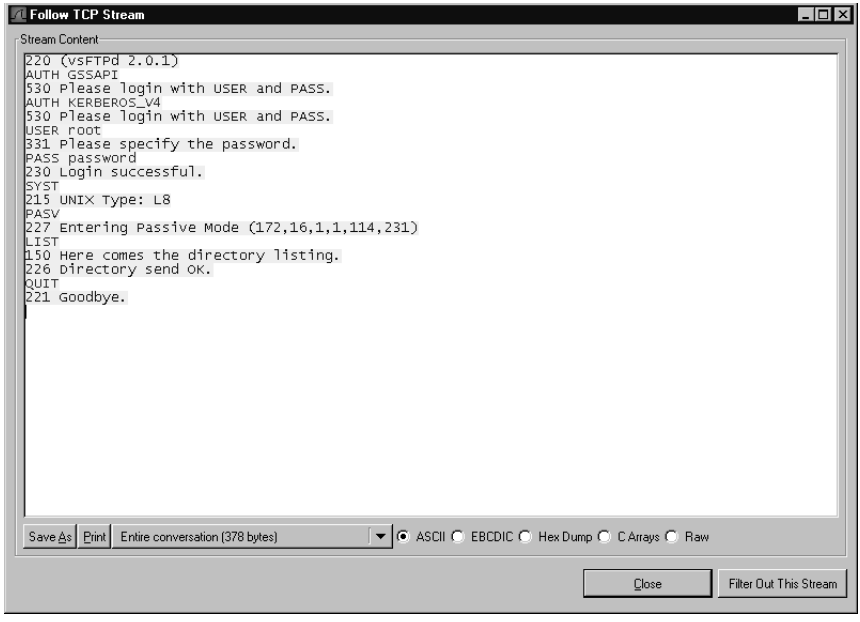


Obrázek 2.1 Grafické uživatelské rozhraní programu Wireshark

Jednou z nejlepších vlastností programu Wireshark je jeho schopnost rekonstrukce všech paketů komunikace TCP a jejich zobrazení ve srozumitelném formátu ASCII (mohou být také zobrazena ve formátu EBCDIC, hexadecimálním formátu nebo v podobě polí jazyka C). Tato data mohou být poté uložena či vytištěna a později použita pro opětovné sestavení webové stránky, relace protokolu SMTP nebo Telnet. V případě, že potřebujete rekonstruovat webovou stránku ze zachycených dat, hledejte sled relace protokolu HTTP a výsledek uložte do souboru. Poté byste měli být schopni takto sestavenou webovou stránku opět lokálně zobrazit (i když bez grafiky) ve svém webovém prohlížeči. Obrázek 2.2 ukazuje výstup sekvence TCP relace protokolu FTP.

Filtry

Filtrování paketů umožňuje najít hledaný paket bez toho, že byste museli prohledávat všechna zachycená data. Wireshark umí používat jak *filtry pro zachytávání dat (tzv. capture filter)*, tak i *filtry pro zobrazení (tzv. display filter)*. Syntaxe filtrů pro zachytávání dat se drží konvencí zavedených programem tcpdump s knihovnou libpcap. Tento filtr se nastavuje pro zachytávání určitých typů dat, a to buď z příkazové řádky nebo jej lze nastavit v dialogovém okně „Capture Filter“. Filtry pro zobrazení poskytují mocný nástroj v podobě třídění zachycených dat. Stejně jako narůstá počet protokolů, narůstá i počet polí pro protokoly ve filtrech pro zobrazování. Avšak ne všechny v současnosti Wiresharkem podporované protokoly mají také filtry pro zobrazování. Kromě toho některé protokoly sice poskytují názvy polí filtrů pro zobrazování, ale jen některých, nikoliv všech. Doufejme však, že s tím, jak se Wireshark stává stále vyzrálejším produktem, a s tím, jak se zvětšuje počet uživatelů přispívajících k jeho vývoji, budou tyto nedostatky odstraněny.



Obrázek 2.2 Sled sekvence TCP

V tabulce 2.1 můžete vidět příklad podporovaných protokolů a filtrů pro zobrazení:

Tabulka 2.1 Filtry pro zobrazení protokolu IP

Pole protokolu IP	Název	Typ
ip.addr	Source or Destination Address	Adresa IPv4
ip.checksum	Header checksum	Unsigned 16-bit Integer
ip.checksum_bad	Bad Header checksum	Boolean
ip.dsfield	Differentiated Services field	Unsigned 8-bit Integer
ip.dsfield.ce	ECN-CE, Explicit Congestion Notification: Congestion Experienced	Unsigned 8-bit Integer
ip.dsfield.dscp	Differentiated Services Codepoint	Unsigned 8-bit Integer
ip.dsfield.ect	ECN-Capable Transport (ECT)	Unsigned 8-bit Integer
ip.dst	Destination	Adresa IPv4
ip.flags	Flags	Unsigned 8-bit Integer
ip.flags.df	Don't fragment	Boolean
ip.flags.mf	More fragments	Boolean
ip.frag_offset	Fragment Offset	Unsigned 16-bit Integer
ip.fragment	IP Fragment	Číslo rámce

Pole protokolu IP	Název	Typ
ip.fragment.error	Defragmentation error	Číslo rámce
ip.fragment.multipletails	Multiple tail fragments found	Boolean
ip.fragment.overlap	Fragment overlap	Boolean
ip.fragment.overlap.conflict	Conflicting data in fragment overlap	Boolean
ip.fragment.toolongfragment	Fragment too long	Boolean
ip.fragments	IP fragments	Žádná hodnota
ip.hdr_len	Header length	Unsigned 8-bit Integer
ip.id	Identification	Unsigned 16-bit Integer
ip.len	Total length	Unsigned 16-bit Integer
ip.proto	Protocol	Unsigned 8-bit Integer
ip.reassembled_in	Reassembled IP in frame	Číslo rámce
ip.src	Source	Adresa IPv4
ip.tos	Type of service	Unsigned 8-bit Integer
ip.tos.cost	Cost	Boolean
ip.tos.delay	Delay	Boolean
ip.tos.precedence	Precedence	Unsigned 8-bit Integer
ip.tos.reliability	Reliability	Boolean
ip.tos.throughput	Throughput	Boolean
ip.ttl	Time-to-live	Unsigned 8-bit Integer
ip.version	Version	Unsigned 8-bit Integer

Jakmile je filtr pro zobrazování nastaven, všechny pakety splňující daná kritéria jsou zobrazeny ve výpisu paketů v panelu *summary*. Tyto filtry můžou být aplikovány pro porovnání polí s hodnotami uvnitř protokolu, jako např. *ip.src == 192.168.1.1*, pro porovnání polí, jako v tomto případě: *ip.src == ip.dst*, nebo prostě jen pro zkontrolování, zda protokol dané pole obsahuje. Filtry se také používají pro statistické funkce a pro barevné zvýraznění paketů. Chcete-li vytvořit jednoduchý filtr pro hledání dat určitého protokolu nebo jeho polí (dejme tomu, že chcete vidět pakety protokolu http), napište **http**. Abyste zobrazili pouze pakety obsahující nějakou žádost (např. GET, POST, HEAD atd.), zadejte **http.request**. Jednotlivá pole filtru mohou být také porovnávána s hodnotami, například takto: **http.request.method=="GET"**. V tomto případě budou ve výpisu zobrazeny jen žádosti protokolu HTTP. Porovnávací operátory mohou být zapsány s použitím následujících zkratek a symbolů:

- **Rovnost:** eq, ==
- **Nerovnost:** ne, !=
- **Větší než:** gt, >
- **Menší než:** lt, <
- **Větší nebo rovno:** ge, >=
- **Menší nebo rovno:** le, <=

Tři operátory mohou být vyjádřeny formou názvu. Operátor *Is Present* umožňuje ověřit, zda dané pole existuje (např. v protokolu ARP [Address Resolution Protocol] je přítomno pole pro adresu MAC, ale není zde číslo portu TCP). Operátor *Contains* se používá pro vyhledávání fráze nebo řetězce v datech paketu. Operátor *Matches* používá řetězec regulérního výrazu (*regex*) pro výkonnější vyhledávání vzorků dat.

Jak vidíte, filtry umožňují značnou pružnost při řešení problémů na síti.

Poznámka: Program Wireshark podporuje mnoho různých typů přenosových médií (jako Ethernet, Token Ring, bezdrátové sítě a sítě ATM [Asynchronous transfer mode]).

Abyste se ujistili, že používáte kompatibilní operační systém, podívejte se na tabulku „Supported Capture Media" na stránce *http://wireshark.org/CaptureSetup/NetworkMedia*. Jak sami uvidíte, operační systém Linux podporuje téměř všechny typy přenosových médií a Ethernet je podporován všemi operačními systémy.

Další informace

Systém subverzí

Subverzování (SVN) je systém pro údržbu verzí vyvíjené aplikace, který umožňuje současnou práci více vývojářů na jednom projektu, přičemž se uchovává informace o tom, jaké změny byly provedeny, kým byly provedeny a jaké verze byly vytvořeny. Je běžné, že jeden projekt obsahuje mnoho verzí ve stromu SVN.

SVN a jeho předchůdce Concurrent Verions Systems (CVS) se používají pro většinu projektů open-source (např. web Sourceforge [www.sourceforge.net] má datová úložiště pro všechny projekty CVS i SVN, které jsou zde uloženy). Některé projekty umožňují přístup ke stromům SVN prostřednictvím webového rozhraní a většina z nich zpřístupňuje svůj obsah pomocí klientských aplikací SVN.

Několik užitečných odkazů o SVN:

- **SVN Command-line Client** – klient založený na práci z příkazové řádky je dostupný na adrese *www.SVN.tigris.org* nebo ve formě instalačních balíčků na mnoha distribucích operačního systému Berkley Software Distribution (BSD) a UNIX.
- **TortoiseSVN** – TortoiseSVN je rozšířením uživatelského prostředí pro operační systém Windows, ve kterém je již vestavěn průzkumník souborů. TortoiseSVN můžete stáhnout na adrese *www.tortoisesvn.tigris.org*.
- **RapidSVN** – RapidSVN je grafickým uživatelským rozhraním SVN pro operační systém Windows.
- **Verze pro Mac OS X a Linux** jsou dostupné ke stažení na *www.rapidsvn.tigris.org*.
- **Visual Studio .NET** – vývojáři, kteří používají Microsoft Visual Studio .NET, mohou pro integraci do SVN použít nástroj třetí strany AnkhSVN (*www.ankhsvn.tigris.org*).

Seznam subverzí projektu Wireshark je dostupný na adrese *www.wireshark.org/develop.html*. Zdrojové kódy Wiresharku je možné získat s použitím SVN několika způsoby:

- **SVN Command Line** slouží k anonymnímu stažení zdrojového kódu.

- **Nightly Snapshots** – používá se ke stažení komprimovaných (.gz.tar) obrazů zdrojového kódu, které se vytvářejí periodicky na 24hodinové bázi.
- **Webové rozhraní SVN** – můžete si stáhnout celou strukturu zdrojových kódů prostřednictvím webového rozhraní SVN na adrese *http://anonsvn.wireshark.org/wireshark/trunk/* a následně zobrazit každý soubor či změny, které byly provedeny mezi jednotlivými verzemi.

Vždy když používáte některou z SVN verzí Wiresharku nebo jiného open-source projektu, mějte na paměti, že se jedná o verze beta, které mohou obsahovat chyby.

Doporučené zdroje

Mezi nejlepší zdroje informací o Wiresharku patří bezesporu i e-mailové distribuční seznamy (formuláře najdete na *www.wireshark.org/list*).

Poznámka: Když vyplňujete registrační formulář, může vám být heslo někdy zasláno formou prostého textu. Ujistěte se, že nepoužíváte toto heslo pro žádné jiné účty, abyste se vyhnuli zneužití zaslaného hesla v případě, že by někdo na vaší síti zachytával data.

- **Wireshark-announce** obsahuje oznámení o vydání nových verzí, oprav chyb a obecné informace týkající se Wiresharku. Do tohoto distribučního seznamu by se měli zaregistrovat všichni uživatelé programu Wireshark, aby byli neustále informováni o důležitých tématech. Členství v tomto distribučním seznamu vám schránku rozhodně nezahltí. Pokud byste chtěli oznámit nějakou zprávu členům tohoto seznamu, zašlete ji na wiresharkannounce@wireshark.org.
- **Wireshark-users** obsahuje obecné informace a nápovědu k programu Wireshark. Uživatelé Wiresharku by se měli do tohoto distribučního seznamu zaregistrovat zejména za účelem sdílení nápadů a doporučení. Komunikace zde je spíše střídmá.
- **Wireshark-dev** se zabývá informacemi o vnitřním dění kolem Wiresharku určenými spíše pro vývojáře a je určen pro uživatele, kteří nějakým způsobem přispívají k jeho vývoji. Objem komunikace v této konferenci je značný. Zprávu můžete odeslat na adresu *wireshark-dev@wireshark.org*.
- **Wireshark-commits** slouží k výměně informací o vývoji a změnách ve struktuře zdrojových kódů. Informuje vývojáře o tom, kdy a k jakým změnám došlo. Datové úložiště SVN automaticky odesílá e-mailovou zprávu vždy, když jsou do úložiště nahrány nějaké změny, a tak tato konference generuje velké množství zpráv. Uživatelé zde neposílají zprávy přímo do tohoto distribučního seznamu. Odpovědi na zprávy byste měli zasílat na adresu *www.wireshark-dev@wireshark.org*.

Při registraci do distribučního seznamu si můžete zvolit, zda si přejete, aby e-mailové zprávy byly „baleny" do jednoho e-mailu, tzv. *daily diggest (denní výběr)*, což je vynikající způsob, jak přespříliš nezatěžovat vaši internetovou linku v případě velkého objemu zpráv. V tomto případě vám však nebudou doručovány přílohy těchto zpráv. Všechny zprávy z výše uvedených e-mailových konferencí jsou archivovány na webových stránkách Wiresharku a dalších webech s duplicitním obsahem. Zprávy jsou roztříděny po jednotlivých měsících, a to zpětně až do roku 1998. V případě potřeby rady či pomoci s konkrétním problémem je vhodné tyto archivy nejprve prohledat a teprve potom vznášet dotaz či žádost do konference. Dalším skvělým zdrojem informací je Uživatelská příručka programu

Wireshark (autorem je Richard Sharpe), kterou najdete na adrese *www.wireshark.org/docs/wsug.html/*. Tato příručka je dostupná také v mnoha dalších formátech včetně PDF na *www.wireshark.org/docs*. A jako vždy, i samotná stránka *www.wireshark.org* obsahuje mnoho užitečných informací. Stránka s příklady zachycených dat (*http://wiki.wireshark.org/SampleCaptures*) obsahuje záznamy zachycených paketů datového provozu, které mohou být staženy a zobrazeny ve Wiresharku.

Podpůrné programy

Většina lidí pracujících s programem Wireshark používá jeho grafické uživatelské rozhraní. Avšak instalační balík Wiresharku nakopíruje do programové složky i některé podpůrné programy. Varianta programu Wireshark pro příkazovou řádku (zvaná *tshark*) obsahuje tři aplikace usnadňující manipulaci se soubory zachycených dat.

Tshark

Tshark je verzí Wiresharku, která pracuje v prostředí příkazové řádky a může být použita pro zachytávání dat ze sítě nebo pro čtení souboru se zachycenými daty. Ve výchozím nastavení zobrazuje tshark řádek se souhrnnou informací. Tedy totéž, co můžete vidět v horním panelu grafického rozhraní Wiresharku. Následující výstup je příkladem výpisu programu tshark:

```
1.199008 192.168.100.132 -> 192.168.100.122 TCP 1320 > telnet [SYN]
Seq=1102938967 Ack=0 Win=16384 Len=0
1.199246 192.168.100.132 -> 192.168.100.122 TCP 1320 > telnet [SYN]
Seq=1102938967 Ack=0 Win=16384 Len=0
1.202244 192.168.100.122 -> 192.168.100.132 TCP telnet > 1320 [SYN
ACK] Seq=3275138168 Ack=1102938968 Win=49640 Len=0
1.202268 192.168.100.132 -> 192.168.100.122 TCP 1320 > telnet [ACK]
Seq=1102938968 Ack=3275138169 Win=17520 Len=0
1.202349 192.168.100.132 -> 192.168.100.122 TCP 1320 > telnet [ACK]
Seq=1102938968 Ack=3275138169 Win=17520 Len=0
```

Přepínač **-V** způsobí, že tshark vypíše strukturu protokolů podobně jako ve středním panelu grafického rozhraní Wiresharku. Takto zobrazí všechny protokoly obsažené v paketu a zahrnuje také datovou část na konci seznamu. Níže můžete vidět detailnější strukturu protokolů z výstupu programu tshark:

```
Frame 5 (74 bytes on wire
74 bytes captured)
Arrival Time: Nov 2
2003 15:22:33.469934000
Time delta from previous packet: 0.000216000 seconds
Time relative to first packet: 1.349439000 seconds
Frame Number: 5
Packet Length: 74 bytes
Capture Length: 74 bytes
Ethernet II
Src: 00:05:5d:ee:7e:53
Dst: 08:00:20:cf:5b:39
Destination: 08:00:20:cf:5b:39 (SunMicro_cf:5b:39)
Source: 00:05:5d:ee:7e:53 (D-Link_ee:7e:53)
```

```
Type: IP (0x0800)
Internet Protocol
Src Addr: 192.168.100.132 (192.168.100.132)
Dst Addr: 192.168.100.122 (192.168.100.122)
Version: 4
Header length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
0000 00.. = Differentiated Services Codepoint: Default (0x00)
.... ..0. = ECN-Capable Transport (ECT): 0
.... ...0 = ECN-CE: 0
Total Length: 60
Identification: 0x160c (5644)
Flags: 0x00
.0.. = Don't fragment: Not set
..0. = More fragments: Not set
Fragment offset: 0
Time to live: 128
Protocol: ICMP (0x01)
Header checksum: 0xda65 (correct)
Source: 192.168.100.132 (192.168.100.132)
Destination: 192.168.100.122 (192.168.100.122)
Internet Control Message Protocol
Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0x3c5c (correct)
Identifier: 0x0500
Sequence number: 0c:00
Data (32 bytes)
0000 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 abcdefghijklmnop
0010 71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69 qrstuvwabcdefghi
```

Nakonec přepínač **-x** způsobí, že tshark vytiskne na obrazovku obsah paketu v hexadecimálním a ASCII formátu buď se souhrnným řádkem nebo strukturou protokolů. Následující příklad zobrazuje hexadecimální a ASCII výstup spolu se souhrnným řádkem:

```
9.463261 192.168.100.122 -> 192.168.100.132 TELNET Telnet Data ...
0000 00 05 5d ee 7e 53 08 00 20 cf 5b 39 08 00 45 00 ..].~S.. .[9..E.
0010 00 9a c3 8a 40 00 3c 06 30 84 c0 a8 64 7a c0 a8 ....@.<.0...dz..
0020 64 84 00 17 05 29 cd 5d 7d 12 4c 1d ea 76 50 18 d....).]}.L..vP.
0030 c1 e8 47 ca 00 00 4c 61 73 74 20 6c 6f 67 69 6e ..G...Last login
0040 3a 20 53 75 6e 20 4e 6f 76 20 20 32 20 31 35 3a : Sun Nov 2 15:
0050 34 34 3a 34 35 20 66 72 6f 6d 20 31 39 32 2e 31 44:45 from 192.1
0060 36 38 2e 31 30 30 2e 31 33 32 0d 0a 53 75 6e 20 68.100.132..Sun
0070 4d 69 63 72 6f 73 79 73 74 65 6d 73 20 49 6e 63 Microsystems Inc
0080 2e 20 20 20 53 75 6e 4f 53 20 35 2e 39 20 20 20 . SunOS 5.9
0090 20 20 20 20 47 65 6e 65 72 69 63 20 4d 61 79 20 Generic May
00a0 32 30 30 32 0d 0a 23 20 2002..#
```

Když jsou ukládána data paketů zachycených tsharkem do souboru, výchozím výstupem je formát knihovny libpcap. Tshark je schopen tato data opět načíst a aplikovat na ně filtry pro zobrazování (také známé jako *filtry pro čtení*) a filtry pro zachytávání dat stejně jako Wireshark. Tshark umí rovněž dekódovat tytéž protokoly jako Wireshark. V podstatě má tytéž schopnosti jako Wireshark

(kromě těch, které přímo souvisejí s grafickým uživatelským rozhraním) v podobě snadno ovladatelného rozhraní příkazové řádky.

Editcap

Účelem Editcapu je odstraňování paketů ze souboru a překlad formátů zachycených dat. V podstatě funguje podobně jako funkce „Save As", ale lépe. Editcap umí číst zcela totožné formáty dat jako Wireshark a standardně je zapisuje ve formátu libpcap. Editcap ale umí také zapisovat zachycená dat do standardních i upravených verzí formátů těchto programů:

- Novel LANalyzer
- Sniffer Network Access Identifier (NAI)
- Microsoft NetMon
- Zachycená data programu Visual Network
- Zachycená data programu Accellent 5Views
- Network Instruments Observer ve verzi 9

Editcap má schopnost upřesnit, zda mají být překládány všechny pakety, nebo jen některé. Na následujícím příkladu vidíme použití editcapu pro překlad prvních pěti paketů z dat zachycených tsharkem za použití knihovny libpcap do formátu Sun snoop (říká se mu *capture_snoop*):

```
C:\Program Files\Wireshark>editcap -r -v -F snoop capture capture_snoop 1-5
File capture is a libpcap (tcpdump Wireshark etc.) capture file.
Add_Selected: 1-5
Inclusive ... 1
5
Record: 1
Record: 2
Record: 3
Record: 4
Record: 5
```

Mergecap

Účelem Mergecapu je spojování několika souborů se zachycenými daty do jednoho výstupního souboru. Mergecap umí přečíst tytéž datové formáty jako Wireshark a standardně zapisuje do formátu libpcap. Mergecap umí také zapisovat do výstupních souborů zachycených dat pro standardní a upravené verze:

- Libpcap
- Sun snoop
- Novel LANalyzer
- NAI Sniffer
- Microsoft NetMon
- Záznam zachycených dat programu Visual Network
- Záznam zachycených dat programu Accellent 5Views
- Network Instruments Observer

Standardně jsou pakety ze vstupních souborů spojovány v chronologickém pořadí daném časovým razítkem každého paketu. Pokud je uveden přepínač **-a**, pakety jsou kopírovány do výstupního souboru přímo, bez ohledu na časová razítka. Následující příklad zobrazuje použití mergecapu pro spojení čtyř souborů se zachycenými daty (*capture1*, *capture2*, *capture3 a capture4*) do jediného výstupního souboru ve formátu Sun snoop nazvaného *merge_snoop,* který bude číst pakety, až dokud nenarazí na konec posledního souboru:

```
C:\Program Files\Wireshark>mergecap -v -F snoop -w merge_snoop capture1
capture2 capture3 capture4
mergecap: capture1 is type libpcap (tcpdump Wireshark etc.).
mergecap: capture2 is type libpcap (tcpdump Wireshark etc.).
mergecap: capture3 is type libpcap (tcpdump Wireshark etc.).
mergecap: capture4 is type libpcap (tcpdump Wireshark etc.).
mergecap: opened 4 of 4 input files
mergecap: selected frame_type Ethernet (ether)
Record: 1
Record: 2
Record: 3
Record: 4
Record: 5
Record: 6
Record: 7
Record: 8
Record: 9
Record: 10
output removed
```

Text2pcap

Text2pcap čte data uložená v ASCII nebo hexadecimálním formátu a zapisuje je do výstupního souboru libpcap. Umí číst hexadecimální výpisy obsahující více paketů a vytvářet soubory zachycených dat z více paketů. Text2pcap může také číst data hexadecimálních výpisů z aplikační úrovně tím, že vkládá uměle vytvořené hlavičky protokolů Ethernet, IP a UDP nebo TCP. Jaký typ hlavičky bude vložen, určuje uživatel. Tímto způsobem umí tedy Wireshark a jiné sniffery číst kompletní data. Následující výpis je příkladem typu hexadecimálního výpisu, který text2pcap umí rozeznat:

```
0000 00 05 5d ee 7e 53 08 00 20 cf 5b 39 08 00 45 00 ..].~S.. .[9..E.
0010 00 9a 13 9e 40 00 3c 06 e0 70 c0 a8 64 7a c0 a8 ....@.<..p..dz..
0020 64 84 00 17 05 49 0e a9 91 43 8e d8 e3 6a 50 18 d....I...C...jP.
0030 c1 e8 ba 7b 00 00 4c 61 73 74 20 6c 6f 67 69 6e ...{..Last login
0040 3a 20 53 75 6e 20 4e 6f 76 20 20 32 20 31 37 3a : Sun Nov 2 17:
0050 30 36 3a 35 33 20 66 72 6f 6d 20 31 39 32 2e 31 06:53 from 192.1
0060 36 38 2e 31 30 30 2e 31 33 32 0d 0a 53 75 6e 20 68.100.132..Sun
0070 4d 69 63 72 6f 73 79 73 74 65 6d 73 20 49 6e 63 Microsystems Inc
0080 2e 20 20 20 53 75 6e 4f 53 20 35 2e 39 20 20 20 . SunOS 5.9
0090 20 20 20 20 47 65 6e 65 72 69 63 20 4d 61 79 20 Generic May
00a0 32 30 30 32 0d 0a 23 20 2002..#
```

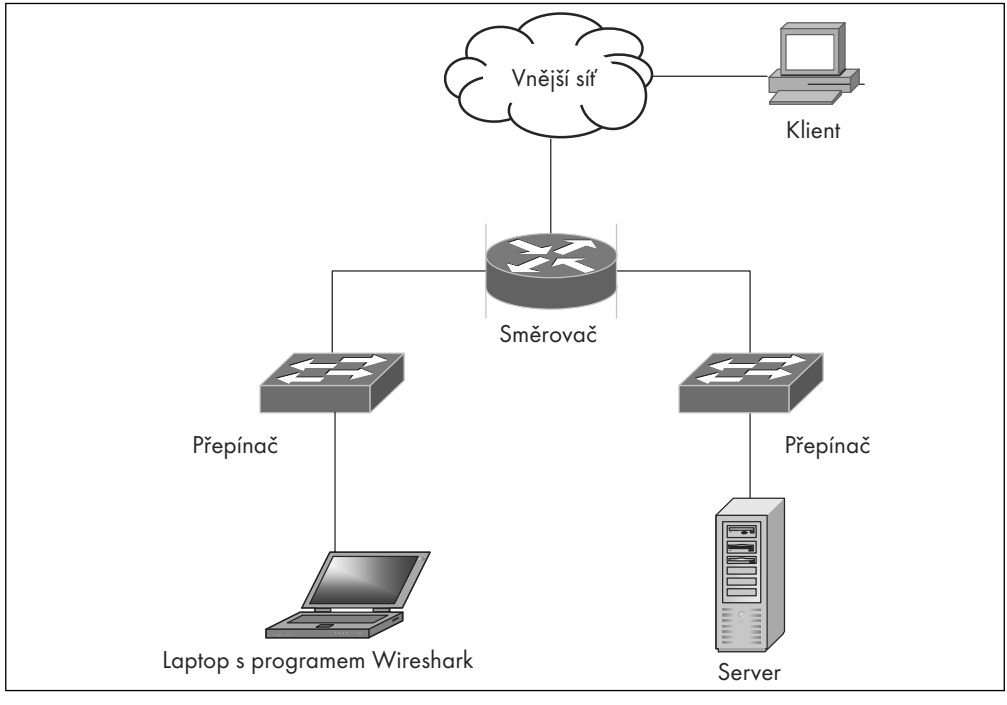
A tento výpis pro změnu ukazuje, jak si text2pcap poradil s přečtením výše uvedených dat (*hex_sample.txt*) a jejich uložením do souboru *libpcap_output*:

```
C:\Program Files\Wireshark>text2pcap hex_sample.txt libpcap_output
```

```
Input from: hex_sample.txt
Output to: libpcap_output
Wrote packet of 168 bytes at 0
Read 1 potential packets
wrote 1 packets
```

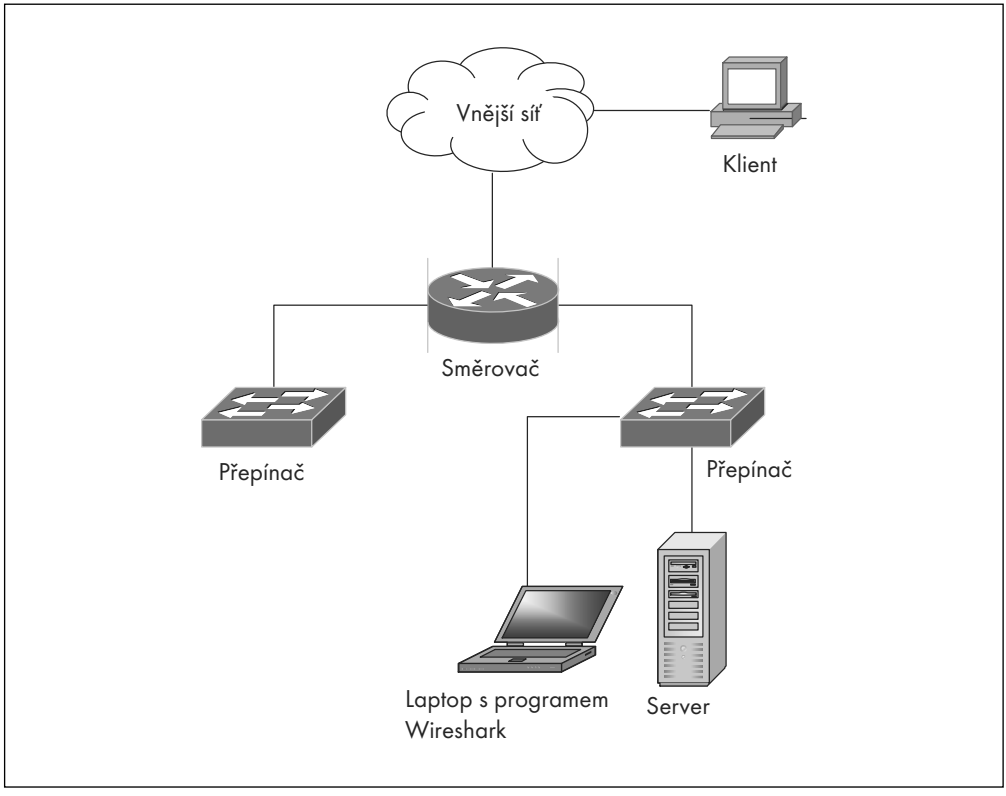
Používání Wiresharku ve vaší síťové architektuře

V první kapitole jsme se dozvěděli něco o různých typech kabelových odboček, rozbočovačích, přepínačích a také o tom, jak mohou být tato zařízení používána pro zachytávání dat síťového provozu. V této části se blíže podíváme na síťovou architekturu a některé kritické body týkající se Wiresharku. Umístění snifferu v síti je pro správnou analýzu a řešení problémů podstatné. Ještě podstatnější je však skutečnost, zda pracujete na správném segmentu sítě. Při řešení problémů v síti se můžete často přesouvat mezi více kabelovými skříněmi nebo i mezi budovami. Už z tohoto prostého důvodu je zcela zřejmé, že Wireshark by měl být provozován na laptopu. Také není od věci mít k tomuto laptopu alespoň jednoduchý rozbočovač a několik kabelů (křížených i přímých) jako součást sady nástrojů pro řešení problémů. Na obrázku 2.3 můžete vidět nesprávné umístění Wiresharku v situaci, kdy chcete zachytávat komunikaci mezi externím klientským počítačem a serverem. Laptop s Wiresharkem by v tomto případě požadovaná data nezachytil, neboť je zcela mimo komunikační cestu serveru, který je připojen k jinému přepínači.



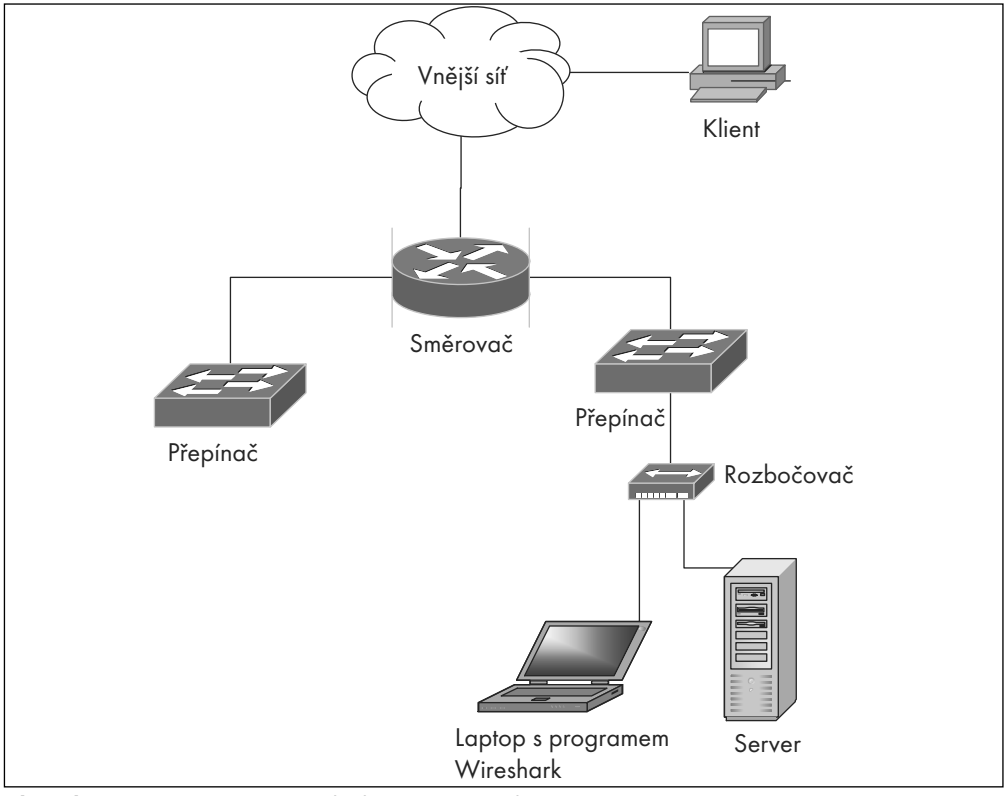
Obrázek 2.3 Nesprávné umístění Wiresharku

Obrázek 2.4 ukazuje, jak zachytávat data cestující po síti od klienta vnější sítě směrem k serveru, a to pomocí *větvení portu (port spanning)*. Laptop s Wiresharkem musí být připojen ke stejnému přepínači jako server. V dalším kroku je třeba aktivovat větvení portu na tomto přepínači pro zrcadlení datového provozu, který putuje k serverovému portu i z něj. Použití této metody nezpůsobí žádné výpadky v komunikaci serveru se sítí.



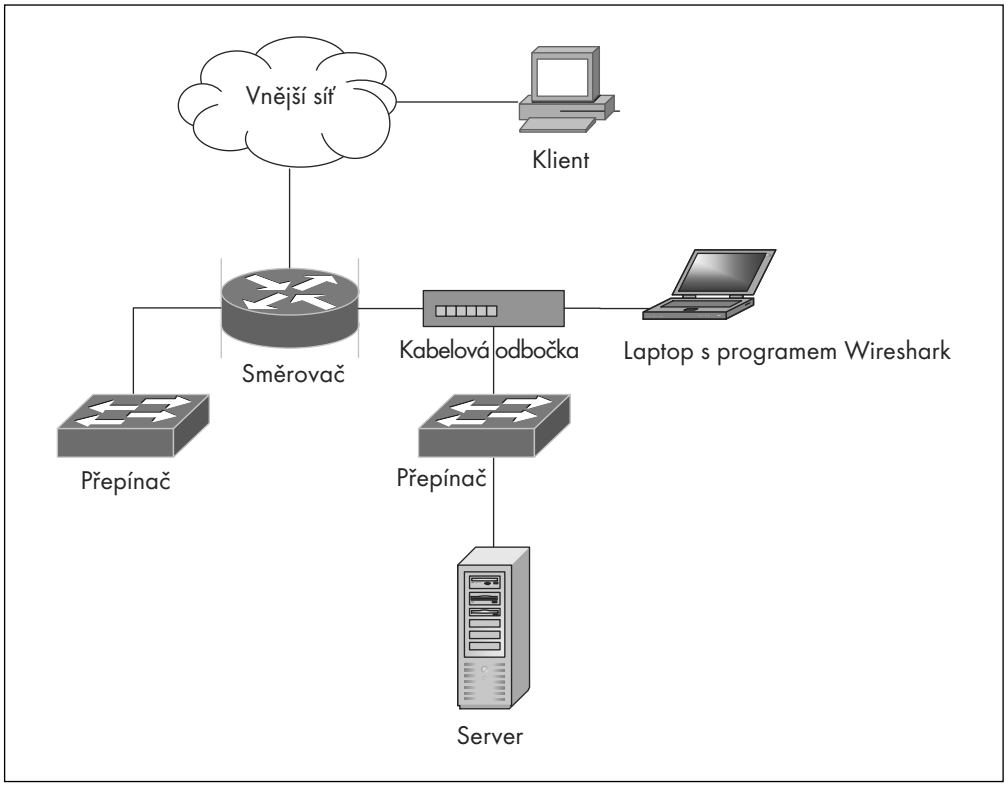
Obrázek 2.4 Správné umístění Wiresharku pro použití větvení portu

Obrázek 2.5 ukazuje, jak zachytávat data přicházející k serveru od klienta z vnější sítě s použitím rozbočovače. Nainstalujte rozbočovač mezi server a přepínač a připojte k němu laptop s programem Wireshark. Wireshark takto uvidí všechen datový provoz přicházející na server i odcházející z něj. Tato metoda naruší síťovou komunikaci na dobu potřebnou pro instalaci rozbočovače a přepojení kabeláže.



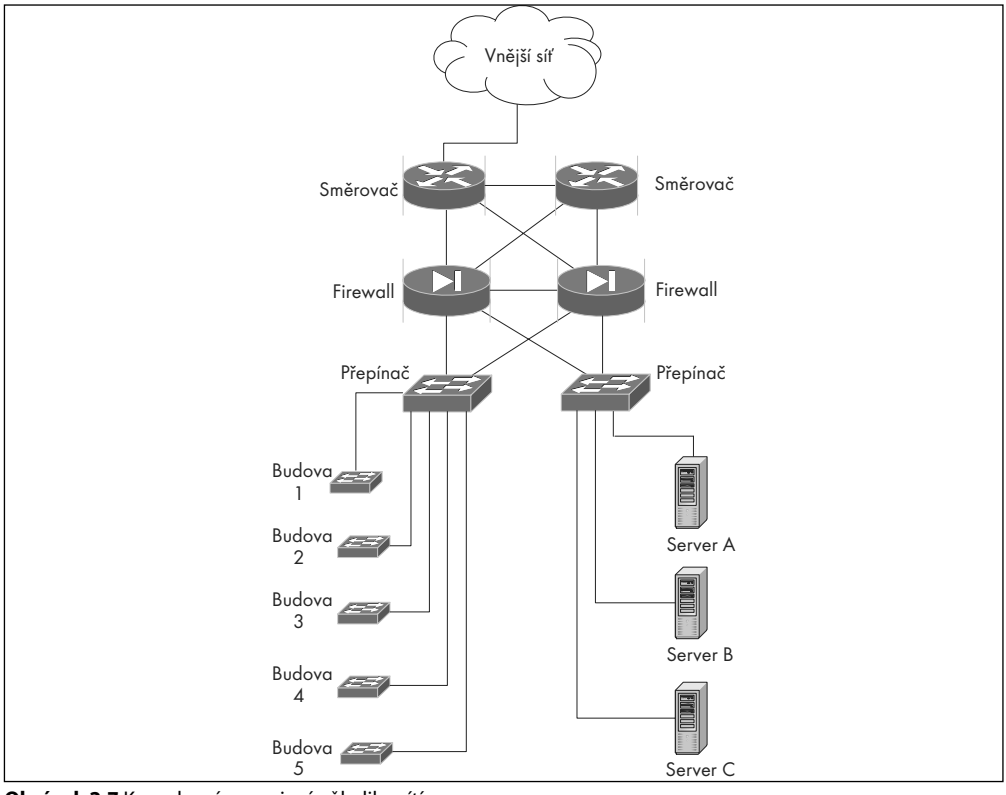
Obrázek 2.5 Správné umístění Wiresharku s použitím rozbočovače

Obrázek 2.6 znázorňuje síťovou architekturu s permanentní kabelovou odbočkou připojenou ke směrovači. Někteří administrátoři používají tuto metodu pro stálé připojení ke kritickým bodům sítě. Laptop s Wiresharkem v tomto případě vidí veškerý datový provoz směrem do i od serveru, a navíc i veškerý provoz na celém síťovém segmentu. Pokud je kabelová odbočka již nainstalována, nezpůsobí použití této metody žádný výpadek na síti. Kabelové odbočky mohou být také přenosné a mohou být použity podobným způsobem jako rozbočovač na obrázku 2.5.



Obrázek 2.6 Umístění Wiresharku s použitím kabelové odbočky

Většina síťových architektur není tak jednoduchá jako ty, které byly uvedeny v této sekci. Avšak tyto vzorové situace vám mohou poskytnout dobrý základ pro použití Wiresharku na různých místech sítě. Některé typy architektury jsou nesrovnatelně složitější, mohou být různě propojeny a zahrnovat i redundantní okruhy (viz obrázek 2.7). Samotné síťové segmenty se mohou dále rozvětvovat až do formy několika úrovní, aby odrážely strukturu firmy sídlící ve více budovách. Abyste byli schopni provádět efektivní a rozumná rozhodnutí pro umístění snifferu ve vaší síti, měli byste ji napřed velmi dobře poznat.



Obrázek 2.7 Komplexní propojení několika sítí

Používání Wiresharku pro řešení problémů se sítí

Snad každý síťový administrátor má za sebou nepříjemnou zkušenost, kdy je probuzen uprostřed noci naléhavým telefonátem žádajícím rychlou nápravu problému na síti. Takový telefonát může vyvolat řadu různých emocí (např. paniku, naléhavost a možná také určitý typ hrdinství). Klíčem ke schopnosti úspěšně vyřešit problémy na síti je znalost práce sítě za normálních okolností. To vám později umožní rychlé odhalení různých anomálií a neobvyklých činností, ke kterým běžně nedochází. Jednou z možností, jak poznat vaši síť, je použití snifferu na různých bodech sítě. To vám umožní získat představu o protokolech, které cestují vaší sítí, o zařízeních na každém segmentu a o tzv. *top talkers* (počítače nebo zařízení, která generují nejvíce síťového provozu).

Jakmile máte představu o tom, jak vaše síť pracuje, můžete začít budovat strategii pro řešení problémů. Díky této strategii budete schopni postupovat metodicky a vyřešíte problém s minimálním výpadkem služeb vašim zákazníkům. Můžete ušetřit drahocenné hodiny ztracené nesprávnou diagnostikou, když strávíte alespoň několik minut vyhodnocením symptomů daného problému při jeho řešení. Správný přístup při řešení problémů se sítí zahrnuje následujících sedm kroků:

1. Identifikace příznaků
2. Definice problému

3. Analýza problému
4. Izolace problému
5. Identifikace a ověření příčiny problému
6. Vyřešení problému
7. Ověření, že problém byl skutečně vyřešen

Prvním krokem při řešení problémů se sítí je tedy *identifikace symptomů*. Někdy se o problémech můžete dozvědět také z jiných sítí, kdy vám některý systém pro správu sítě zašle zprávu o vzniku problému (jako např. problémy s výkonem, konektivitou nebo jiné podivné chování sítě). Porovnejte tyto změny s normálním stavem sítě: Provedli jste předtím, než se projevil problém, nějakou změnu na síti nebo na serveru? Nebyla třeba spuštěna nějaká naplánovaná úloha jako zálohování? Byl pro tuto úlohu předem vytvořen přiměřený časový rámec? Až zodpovíte tyto a další podobné otázky, bude vhodná chvíle pro sepsání srozumitelné *definice* problému. Jakmile jsou popsány symptomy a problém je identifikován, měli bychom jej podrobit důkladné analýze. Měli byste shromáždit data potřebná k analýze a pokusit se zúžit výběr možných příčin problému. Je tato příčina v jádru sítě, v jediné budově nebo jen na pobočce vaší firmy? Má problém souvislost s celým síťovým segmentem, nebo se projevuje pouze na jediném počítači? Může být tento problém navozen i jinde na síti? Možná budete muset prověřit více částí vaší sítě, abyste jádro problému úspěšně lokalizovali.

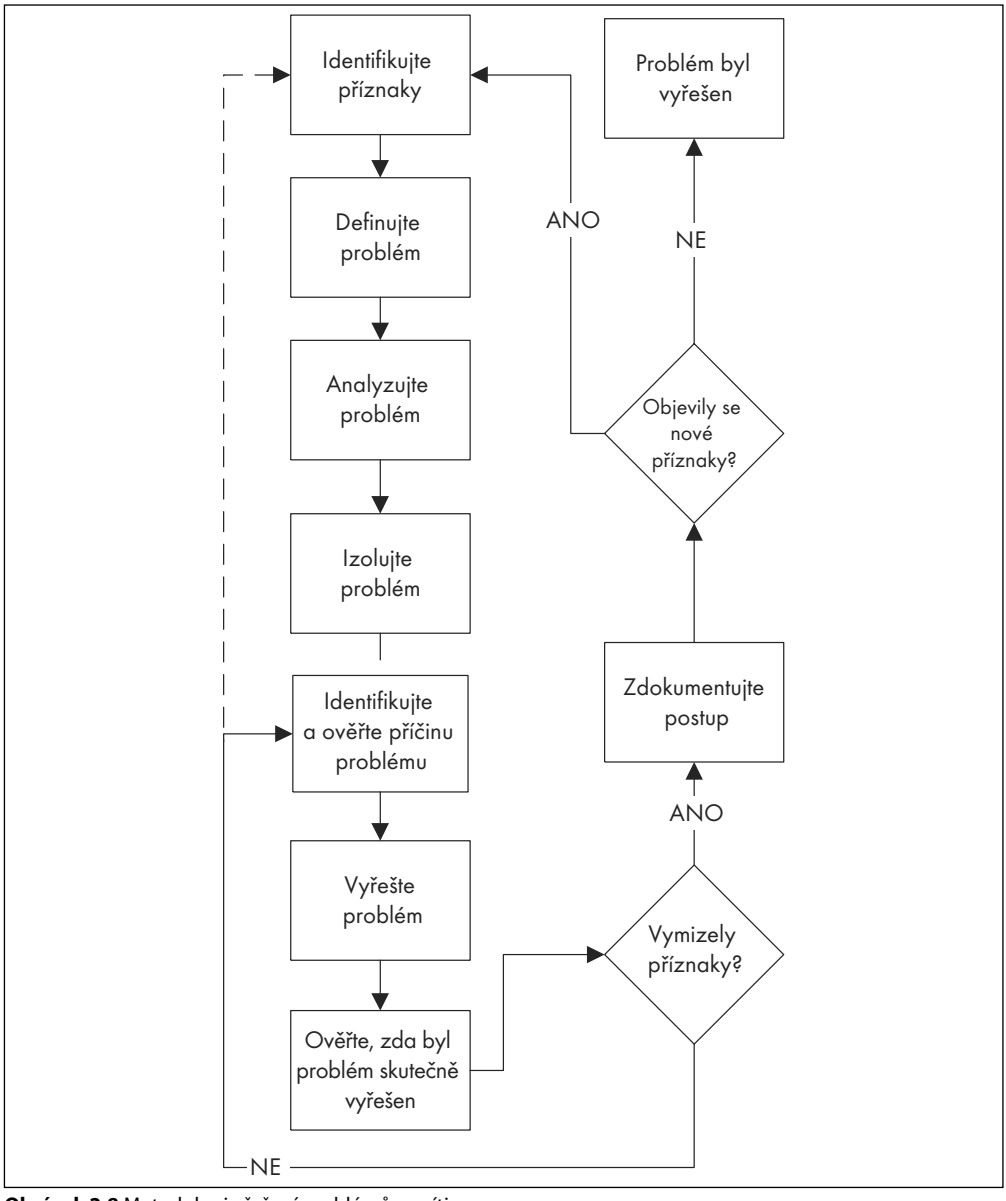
Nyní, když jste příčinu problému nalezli, by měly vaše další kroky vést k *izolaci* problému. Existuje mnoho způsobů, jak toho dosáhnout, například odpojením počítače, který problematické chování způsobuje, restartováním serveru, zastavením podezřelé datové komunikace aktivací některého pravidla na firewallu nebo přepnutím na záložní internetové připojení.

Další na řadě je *identifikace a ověření* příčiny problému. V okamžiku, kdy máte nějakou teorii o tom, co může problém působit, měli byste tuto teorii ověřit či vyvrátit. Váš síťový analyzátor vám umožňuje vidět, co se děje na pozadí sítě. V této fázi můžete hledat informace o problému na Internetu, obrátit se na výrobce hardwarového vybavení nebo vývojáře aplikace nebo také kontaktovat vašeho poskytovatele internetového připojení. Na webových stránkách *www.cert.org* a *www.incidents.org* se navíc můžete přesvědčit, zda váš problém nepostihuje také další uživatele po celém světě.

Až přijdete na to, jaký postup vede k vyřešení problému, nadejde ta pravá chvíle k realizaci tohoto postupu, potažmo *vyřešení* problému. To může proběhnout kupříkladu upgradem hardwaru nebo programového vybavení, zavedením nového pravidla na firewallu, reinstalací napadeného systému, výměnou vadné hardwarové komponenty nebo přehodnocením návrhu topologie segmentu sítě.

Posledním krokem by mělo být vždy *ověření* správnosti řešení, tedy ujištění, že problém byl skutečně vyřešen. Zjistěte, zda řešení jednoho problému nezpůsobilo problémy jiné nebo že konkrétní problém, který jste vyřešili, není pouze symptomem jiných, možná mnohem závažnějších potíží. Součástí této fáze je i sepsání dokumentace, kde budou obsaženy informace o krocích, které vedly k odstranění problému. Tyto mohou být užitečnou pomůckou pro řešení problémů v budoucnosti. Pokud se vám problém vyřešit nepodařilo, měli byste začít s celým postupem od úplného začátku. Vývojový diagram na obrázku 2.8 znázorňuje proces řešení problémů.

Poznámka: Abyste mohli být kvalitním odborníkem na řešení problémů se sítí, potřebujete velmi dobře rozumět síťovým protokolům. Porozumění různým protokolům a jejich vlastnostem vám pomůže rozeznat různé odchylky od normálního chodu sítě.



Obrázek 2.8 Metodologie řešení problémů na síti

Používání Wiresharku pro administraci systému

Systémoví administrátoři jsou notoricky známí otázkou, zda je něco špatně na síti, zatímco síťoví administrátoři s oblibou kontrují, že chyba je někde v systému. Někde v propasti mezi těmito dvěma názory leží skutečnost, kterou můžete pomocí Wiresharku odhalit.

Kontrola síťové konektivity

Srdcem moderních sítí postavených na sadě protokolů TCP/IP je Ethernet. Ethernet je protokol, který pracuje bez zbytečného povyku; avšak existuje nesčetně mnoho problémů s ovladači, operačními systémy, nastavením systému, aplikacemi, síťovými přepínači atd. Systémový administrátor potřebuje ke své práci nástroj, který dokáže zjistit, zda síť pracuje správně z hlediska druhé vrstvy síťového modelu OSI nebo z hlediska protokolu Ethernet.

Když dojde k problému s administrací systému, mělo by následovat ověření, zda postižený systém dostává datové pakety ze sítě. Nejzákladnějším paketem je paket protokolu ARP.

Podstata paketu ARP je následující: když počítač potřebuje komunikovat s jiným počítačem na stejné podsíti a zná jeho adresu IP, avšak nikoliv adresu MAC, tak tento počítač rozešle formou všesměrového vysílání žádost ARP na celý ethernetový segment (např. síť s počítači s adresami IP 192.168.1.1 a 192.168.1.2, které mají adresu MAC 00:01:02:03:04:05 a 06:07:08:09:0a:0b) a odešle následující sled příkazů prostřednictvím protokolu ARP:

```
00:01:02:03:04:05 to ff:ff:ff:ff:ff:ff Who has 192.168.1.2? Tell 192.168.1.1
06:07:08:09:0a:0b to 00:01:02:03:04:05 192.168.1.2 is at 06:07:08:09:0a:0b
```

Když víme, že datový provoz protokolu ARP je nezbytným předchůdcem běžného datového provozu, můžeme použít program Ethereal, abychom zjistili, že se na síti vůbec nějaká komunikace odehrává. Existuje několik stavů protokolu ARP, které signalizují specifický problém. Pokud na síti není žádný provoz protokolu ARP, tak buď nezachytáváte síťový provoz správně anebo existuje důvod domnívat se, že jsou zde problémy s ovladači nebo operačním systémem, které brání síťové komunikaci. Pokud počítač odesílá žádosti protokolu ARP, ale nedostává se mu žádné odpovědi od cílového počítače, může to znamenat, že tento počítač není v síti přítomen. Ujistěte se, že tento systém je na správné síti LAN; toto už není tak jednoduché jako prosté zasunutí správného síťového konektoru. Když systém dostává žádosti protokolu ARP a odesílá data protokolu IP na síť, ale nedostává se mu žádné odpovědi, kterou jste možná ověřili pomocí snifferu, problémem může být nastavení firewallu nebo problém s ovladačem v operačním systému.

Kontrola dostupnosti síťových aplikací

Poté co jste ověřili dostupnost sítě jako takové, je dalším krokem v řešení problému kontrola konektivity mezi počítači a aplikacemi. Protože většina síťových aplikací je založených na protokolu TCP, testování je omezeno na aplikace používající tento protokol. Pro ilustraci uveďme webový server operující na portu 80.

Jak už bylo dříve uvedeno, protokol TCP spoléhá na tzv. třícestný handshake (navázání komunikace před samotným zahájením přenosu dat – pozn. překl.). Tento handshake může sám o sobě indikovat určitý problém s komunikací aplikace. Protože Wireshark má schopnosti k tomu, aby dokázal doslova rozebrat pakety protokolu TCP, můžeme jej použít pro lokalizaci problému. S následujícími scénáři se můžete setkat v průběhu řešení některých problémů komunikace aplikací po síti:

Scénář 1: Zpráva SYN bez zprávy SYN+ACK

Pokud data zachycená Wiresharkem ukazují, že klient odesílá paket SYN, ale nedostává žádnou odpověď od serveru, pravděpodobně to znamená, že server tento paket nezpracoval. Důvodem může

být firewall mezi těmito dvěma systémy, který blokuje potřebné pakety, nebo skutečnost, že firewall běží přímo na serveru.

Scénář 2: Zpráva SYN s okamžitou odezvou RST

Když data zachycená Wiresharkem ukazují, že server odpovídá paketem s nastaveným příznakem RST, znamená to, že pakety jsou cílovému serveru doručovány, ale na daném portu nenaslouchá žádná aplikace. Ujistěte se, že vaše aplikace naslouchá na správném portu na správné adrese IP.

Scénář 3: SYN SYN+ACK ACK

Spojení ukončeno

Pokud vám síťový provoz zachycený Wiresharkem ukazuje, že relace TCP je zahájena a bezprostředně poté ukončena, pravděpodobně cílový server tuto komunikaci odmítá z důvodu bezpečnostních opatření. Na operačních systémech UNIX zkontrolujte soubor *tcpwrappers* v adresáři */etc/hosts.allow* a */etc/hosts.deny* a ověřte, zda jste komunikaci nedopatřením nezablokovali.

Používání Wiresharku pro administraci zabezpečení

„Je tento protokol bezpečný?" Jedním z obvyklých úkolů, kterými se administrátoři zabezpečení zabývají, je ověřování bezpečnosti libovolných protokolů. Wireshark je ideálním nástrojem pro tuto práci.

Jednou z nejoblíbenějších a nejužitečnějších funkcí Wiresharku je jeho schopnost *rekonstrukce paketů*, která umožňuje vidět obsah komunikovaných dat. Pro protokoly jako Telnet a FTP Wireshark jednoduše zobrazuje uživatelská jména a hesla daného připojení bez nutnosti opětovného sestavování paketů. Avšak pro protokoly neznámé, proprietární či jiným způsobem nesrozumitelné můžeme použít rekonstrukci paketů. Abyste mohli provést rekonstrukci, zachyťte data pomocí Wiresharku nebo jiného nástroje a poté soubor se zachycenými daty otevřete ve Wiresharku. Klepněte pravým tlačítkem myši na jakýkoliv paket z dané relace. Vyberte možnost **Follow TCP Stream**; otevře se okno obsahující veškerá komunikační data, která proběhla v rámci této relace. Pro lepší čitelnost textu vám může pomoci zapnutí volby ASCII, a pokud je protokol příliš rušený, zkuste vybrat různé typy zobrazení pomocí položek *sender, receiver* nebo *Entire conversation*.

Odhalení aktivity IRC (Internet Relay Chat)

Kromě toho, že používání chatovacích místností může mít svá omezení firemními směrnicemi, je s programem IRC spojena také skutečnost, že tato aplikace je velmi často používána hackery jako kontrolní a ovládací mechanismus. Protokol IRC používá pro komunikaci port 6667. Když nastavíte Wireshark tak, aby sledoval provoz určený pro provoz na portu 6667, uvidíte datovou komunikaci protokolu IRC, která vypadá podobně jako na následujícím příkladu:

```
Local client to IRC server
port 6667:
USER username localsystem.example.com irc.example.net :gaim
Remote IRC server to local client:
NOTICE AUTH :*** Looking up your hostname...
Local client to IRC server
```

```
port 6667:
NICK clever-nick-name
Remote IRC server to local client:
NOTICE AUTH :*** Checking identNOTICE AUTH :*** Found your hostname
```

Pokud vidíte podobný typ dat, můžete si být jisti, že na vaší síti někdo používá klienta protokolu IRC. Raději zjistěte, který uživatel to je.

Wireshark jako síťový systém pro detekci průniku (NIDS)

Přestože existují specializované open-source nástroje pro síťové systémy detekce průniku (NIDS) jako Snort od Sourceire (*www.snort.org*), pokud nemáte po ruce nic jiného než Wireshark, může vám pomoci i jako NIDS, protože jeho funkce mimo jiné zahrnují i schopnost vyvolat nějakou akci na základě různých kritérií. Podívejme se na některá pravidla, která bychom mohli ve Wiresharku případně nastavit:

- Připojení k vnitřní databázi iniciovaná jinými systémy než vaše vlastní webové servery,
- Pokusy o odeslání e-mailu na externí e-mailové servery prostřednictvím portu 25 z jiných serverů než vašich vlastních.
- Pokusy o použití Vzdáleného připojení k ploše (RDC) zvnějšku vaší sítě nebo použití Wiresharku jako honeypotu, naslouchání pokusům o připojení z nepoužívané adresy IP.

Wireshark jako detektor přenosu firemních informací

Pokud vaše firma označuje důvěrné či firemní informace nějakým ustáleným způsobem, není důvod, proč byste nemohli použít Wireshark pro odhalení přenosu těchto informací. Můžete použít Wireshark tak, aby zachytával veškerou odchozí komunikaci pomocí větvení portu, a poté použít funkci *Find packet.* Avšak tato možnost představuje obrovské množství dat, která se budou muset prohledávat. Abyste toto množství zredukovali na přijatelnou míru, použijte filtry pro zachytávání. Tyto by měly být nastaveny tak, aby vyloučily z prohledávání datový provoz, u kterého nepředpokládáte, že by obsahoval důvěrná data (např. dotazy DNS).

Zabezpečení Wiresharku

Ačkoliv je Wireshark považován za nástroj pro zabezpečení, ani on sám není zcela bez občasných bezpečnostních chyb. Podle stránek www.Securityfocus.com bylo mezi lety 2002 a 2006 uveřejněno 44 článků o jednotlivých chybách Etherealu a Wiresharku. Většina z těchto chyb byla obsažena v dekodérech velmi specifických nebo zřídka používaných protokolů. I tak byste však měli vědět několik věcí o tom, jak minimalizovat rizika způsobená případnými chybami v aplikaci Wireshark.

Prvním předpokladem pro bezpečnější běh Wiresharku je instalace aktualizací programového vybavení počítače a operačního systému. Aktualizace Wiresharku běžně vycházejí po několika měsících a mnoho lidí používá binární instalační balíky, které se snadno aktualizují. Dalším krokem na cestě k bezpečnějšímu Wiresharku je oddělení procesu zachytávání od procesu analýzy dat, přičemž oba z nich budou provozovány s množinou nejnižších oprávnění na operačním systému, která pro daný úkon postačují. Většinou knihovny aplikací pro zachytávání dat pro svůj běh vyžadují práva místního administrátora v operačním systému Windows nebo uživatele root v Unixových systémech. Pro-

tože většina problémů, které se v minulosti ve Wiresharku vyskytly, pramenila z různých dekodérů, pokud budete spouštět konzolu pro analýzu dat pod ne-administrativním účtem, můžete takto snížit riziko výskytu bezpečnostních problémů, jež by degradovaly vaše úsilí investované do analýzy dat. Také můžete použít tshark nebo dumpcap (oba jsou součástí balíku Wireshark) k zachytávání datové komunikace do souboru a později použít Wireshark pod neprivilegovaným účtem k analýze výsledků.

Optimalizace Wiresharku

Optimalizace systému, na kterém provozujete váš sniffer, bude dlouhou cestou k urychlení běhu Wiresharku nebo jakékoliv jiné aplikace pro zachytávání dat. Ovladače síťové karty i Wireshark odvedou výborně svou práci pro zachycení veškerých dat, která jimi projdou, ale abyste měli jistotu, že vidíte komplexní obraz vaší sítě, měli byste také zvážit některé případné problémy vztahující se k systému.

Rychlost síťového připojení

Wireshark nemůže zachytávat pakety rychleji, než dovoluje rychlost nejpomalejšího bodu mezi vámi a systémem, jehož data zachytáváte. Zvláštní pozornost by měla být věnována tomu, aby Wireshark nebyl umístěn na nejpomalejším síťovém připojení, pokud je síť více vytížená (např. když zachytáváte data ze spojení mezi dvěma počítači, které si vyměňují data rychlostí 75mbps, a vy použijete větvení portu rychlostí 10mbps, může to způsobit, že nebudete schopní zachytit všechna síťová data).

Minimalizace příslušenství Wiresharku

Wireshark je efektivní aplikací pro zachytávání paketů, a to nejen díky své ceně. Má dobrou schopnost efektivně zachytávat data, některé z jeho pokročilejších voleb však mohou jeho činnost zpomalit.

V dialogovém okně **Capture Options** (Možnosti zachytávání) mohou zpomalení činnosti způsobit zejména následující volby: **Update list packets in real-time** (Obnovovat seznam paketů v reálném čase), **Automatic scrolling in live capture** (Automatický posun při zachytávání) a jakákoliv z voleb v sekci **Name resolution** (Překlad názvů). Volba **Enable network name resolution** (Povolit překlad názvů v síti) může znatelně zpomalit proces zachytávání dat na vytížené síti, protože pro každý zdroj i cíl bude proveden překlad pomocí systému DNS. Pokud víte, jaký typ paketů hledáte, může použití filtrů pro zachytávání práci značně urychlit. (Tip: když oddělíte funkce Wiresharku pro zachytávání od funkcí pro analýzu, budete moci zachytávat větší množství dat.) Použijte tcpdump, tshark nebo jiný nástroj specifický pro váš operační systém k zachytávání dat do souboru. Poté co nahromadíte potřebné množství dat, otevřete tato data ve Wiresharku.

Procesor

Pro běh Wiresharku není výkonný počítač nezbytně nutný, nicméně úkoly jako vyhledávání řetězců ve velkém množství zachycených paketů budou rychlejší s použitím rychlejšího procesoru. Wireshark je neustále optimalizován pro vyšší výkon, ale nezpochybnitelným faktem zůstává skutečnost, že rychlejší procesor dokáže zpracovat více operací za vteřinu, což snižuje množství času stráveného čekáním na zpracování paketů nebo potřebného k nalezení určitých částí textu v zachycených paketech.

Paměť

Nejefektivnějším způsobem, jak urychlit běh Wiresharku, je přidat mu více paměti RAM. Tato komponenta nabývá na významu s tím, jak roste velikost souborů se zachycenými daty, s kterými pracujete. Jako všechny aplikace, i Wireshark potřebuje určité množství paměti, kde může podržet zpracovávaná data. Když operační systém nemá dostatek paměti k udržení aplikace, začne dočasně ukládat data z paměti do stránkovacího souboru (swap nebo též page file) umístěného na pevném disku. Neustálý přesun dat z paměti RAM do stránkovacího souboru samozřejmě nějaký čas trvá. Dojde-li k tomu, že Wireshark nemá k dispozici dostatek fyzické paměti, swapování (práce s virtuální pamětí) zpomalí celý systém i další aplikace.

Pokročilé techniky zachytávání dat

Existují určité alternativy k používání Wiresharku v podobě větvení portů nebo používání kabelových odboček. Naneštěstí tyto techniky mohou být zneužity útočníky k zachytávání hesel a jiných dat z vaší sítě.

Dsniff

Dsniff je nástrojem Duga Songa určeným pro zachytávání dat. Dsniff a mnoho podobných jsou dostupné na stránce *www.monkey.org/~dugsong/dsniff*. Dsniff je asi nejznámější svou schopností zachytávat komunikaci ověřovacího procesu (jako uživatelská jména a hesla) a celkově propracovaným zachytáváním dat. Aktuální verze Dsniffu dokáže dekódovat přihlašovací informace následujících protokolů:

- America Online (AOL) Instant Messenger (IM) (Citrix Wireframe)
- CVS
- File Transfer Protocol (FTP)
- HTTP
- I seek You (ICQ)
- IMAP
- IRC
- Lightweight Directory Access Protocol (LDAP)
- Remote Procedure Call (RPC) mount requests
- Napster
- Network News Transfer Protocol (NNTP)
- Oracle SQL*Net
- Open Shortest Path First (OSPF)
- PC Anywhere
- Post Office Protocol (POP)
- PostgreSQL
- Routing Information Protocol (RIP)
- Remote Login (rlogin)
- Windows NT plaintext Server Message Block (SMB)

- Network Associates Sniffer Pro (remote)
- Simple Network Management Protocol (SNMP)
- Socks
- Telnet
- X11
- RPC yppasswd

Vzhledem k nasazení přepínaných sítí a šifrovaných protokolů není zachytávání hesel vždy tak efektivní, jak bychom si přáli. Balík Dsniff obsahuje několik nástrojů pro přesměrování provozu a útoky typu MITM (*man-in-the-middle*, tedy situace, kdy je útočník připojen k bodu sítě nacházejícímu se mezi dvěma komunikujícími systémy), které se dají použít jednak pro přesměrování datové komunikace a také pro dešifrování relací.

Prvním nástrojem je *arpspoof* (původně byl znám pod názvem *arpredirect*), který je používán hostitelskými systémy k nalezení adresy MAC místního směrovače. Podvržením paketů protokolu ARP lze přesvědčit sousedící počítače, že počítač, který používáte, je směrovačem. Pro vás to však znamená, že takto získané pakety musíte přeposílat legitimnímu směrovači. Nicméně v mezičase, který uplyne mezi obdržením „ukradeného" paketu a jeho přeposláním legitimnímu směrovači, má Dsniff šanci daný paket zpracovat. Tento způsob dobře funguje na místních sítích s přepínačem a také na sítích, které používají kabelový modem. Nicméně tento způsob není zcela dokonalý. V podstatě se snažíte dalším počítačům na síti vnutit informaci o místní adrese MAC. Důsledkem této akce je, že datový tok procházející vaším počítačem může být občas narušen. Tato technika může být snadno odhalena síťovým systémem pro detekci průniku (NIDS). Například program Sniffer Pro obsahuje expertní diagnostický mód, který označí podezřelá data příznakem „duplicitní adresa IP" (např. když je na síti více počítačů, které tvrdí, že jejich adresa IP je adresou směrovače).

Utilita *dnsspoof* přesměrovává datový provoz falšováním odpovědí místního serveru systému DNS. Když v prohlížeči otevřete webovou stránku, například www.example.com, váš počítač pošle místnímu serveru DNS žádost o překlad hostitelského názvu na adresu IP. Tento překlad většinou chvilku trvá, čehož využívá právě nástroj dnsspoof, který zašle odpověď dříve, než by byla zaslána z legitimního serveru DNS. Oběť tohoto typu útoku přijme informaci, která dorazila jako první, a všechny další již bude ignorovat. Podvržená odpověď na dotaz DNS obsahuje jinou adresu IP, než je ta, která by byla součástí odpovědi z legitimního serveru. Většinou je podvržená adresa adresou IP útočníkova počítače, přičemž útočník pravděpodobně na svém počítači používá některý z dalších nástrojů pro útok typu MITM z balíku Dsniff. Název útoku MITM má své kořeny v kryptografii a popisuje situaci, kdy někdo zachytí komunikaci probíhající mezi dvěma subjekty, pozmění informace obsažené v této komunikaci a přepošle je příjemci. Nástroje balíku Dsniff, které můžete pro útoky MITM použít, jsou *webmitm* pro provoz protokolu HTTP (včetně protokolu Secure Sockets Layer – SSL) a *sshmitm* (pro protokol Secure Shell – SSH). Protokoly SSH a SSL jsou považovány za bezpečné šifrované protokoly, u kterých nelze zachytávání dat efektivně použít. Nástroje MITM však pracují tím způsobem, že klientům účastnícím se komunikace prostřednictvím protokolů SSH/SSL vnutí svůj vlastní šifrovací klíč. Takto je možné dešifrovat datový provoz, zachytit hesla a poté kompromitovaná data opět zašifrovat s použitím originálního šifrovacího klíče legitimního serveru. Teoreticky se proti tomuto typu útoku můžete chránit kontrolováním platnosti certifikátu serveru, ale prakticky tuto kontrolu v podstatě nikdo neprovádí. Dsniff je schopen zachytávat hesla a jakýkoliv jiný datový provoz v podobě prostého textu.

Utilita *mailsnarf* slouží k zachytávání e-mailové komunikace (podobně jako systém síťových odposlechů Carnivore americké FBI), kterou po zachycení rekonstruuje do formátu *mbox* čitelného většinou aplikací učených pro práci s elektronickou poštou.

Pro zachytávání komunikace klientů ICQ, IRC, Yahoo!, Messenger a AOL IM slouží nástroj *msgsnarf.*

Filesnarf disponuje schopností zachytávat data souborů přenášených pomocí systému Network File Systém (NFS).

Urlsnarf umí zachytávat adresy Uniform Resource Locator (URL) putující sítí.

Nástroj *webspy* dokáže v reálném čase posílat takto zachycené adresy URL do webového prohlížeče Netscape, čímž umožňuje útočníkovi sledovat, co dělá jeho oběť v prohlížeči a jaké stránky právě prohlíží.

Utilita *macof* je určena k posílání enormního množství adres MAC, což je jeden ze způsobů, jak napadnout ethernetový přepínač. Většina přepínačů disponuje pouze omezenou velikostí tabulky pro uchovávání adres MAC, kterých můžou uchovat „jen" 4 000. Když je přepínač zahlcen, může se přepnout do stavu *fail open* (odemčení při chybovém stavu – pozn. překl.) a začne vysílat všechny pakety na všechny porty, čímž de facto umožní útočníkovi zachytávání dat.

Program *tcpkill* se používá k násilnému ukončení relací TCP a může být použit jako útok typu odepření služby (DoS). Můžete jej například nastavit tak, aby ukončil každý pokus vašeho souseda o připojení. Tcpkill může být také integrován do systému pro detekci průniku tak, aby odpojil relace případného hackera. Utilita *tcpnice* je velmi podobná nástroji tcpkill, avšak s tím rozdílem, že místo ukončení relace ji pouze zpomalí (příkladem budiž situace, kdy zfalšujete zdrojovou adresu paketů protokolu Internet Control Message Protocol [ICMP] tak, aby představovala adresu IP směrovače vašeho souseda, díky čemuž se vám dostane vyšší kapacity přenosového pásma na úkor kapacity sousedovy linky).

Ettercap

Program Ettercap má podobné vlastnosti jako dsniff. Má mnoho společných funkcí (např. schopnost útoku MITM proti protokolům SSH a SSL a zachytávání hesel). Navíc má však další funkce pro provádění útoků MITM proti běžným relacím protokolu TCP, jako je vkládání příkazu do datového proudu. Ettercap byl vytvořen Albertem Ornaghim a Marcem Vallerim. Stáhnout jej můžete na adrese *http://ettercap.sourceforge.net.*

Útoky MITM

Nejúčinnější obranou proti zachytávání dat je používání šifrovaných protokolů, jakými jsou SSL a SSH. Avšak poslední verze balíků dsniff a Ettercap obsahují nástroje pro obelstění šifrování, známé jako útoky MITM. Stejná technika může být použita na šifrovaný protokol, když si útočník nainstaluje server, který bude odpovídat na požadavky klientů. Například server bude odpovídat požadavkům na připojení k serveru na https://www.amazon.com. Uživatel připojující se k tomuto serveru se bude domnívat, že navázal šifrované spojení s webovým serverem Amazon.com. V tentýž okamžik naváže útočník spojení se skutečným serverem Amazon.com a předstírá, že je uživatelem. V tomto případě hraje útočník obě dvě role, přičemž dešifruje data přicházející od uživatele a znovu je šifruje pro

přenos k původnímu příjemci. Teoreticky šifrovací protokoly disponují obranou proti tomuto typu útoku. Server, který se vydává za Example.com, musí nějak dokázat, že je skutečně serverem Example.com. Avšak ve skutečnosti většina uživatelů tento mechanismus ignoruje. Útoky typu MITM se v praxi ukázaly jako velice efektivní způsob napadení.

Cracking

Nástroje jako dsniff a Ettercap zachytávají hesla v nešifrované i šifrované podobě. Teoreticky zachytávání šifrovaných hesel postrádá smysl. Avšak uživatelé někdy volí tzv. slabá hesla (např. slova, která najdete ve slovníku) a útočníkovi zabere jen několik vteřin, aby prošel slovník o 100 000 výrazech a porovnal zašifrovanou podobu každého výrazu se zachyceným zašifrovaným heslem. Pokud je nalezena shoda, útočník zjistil původní heslo. Existují nástroje pro prolamování hesel a jak dsniff, tak Ettercap umí ukládat výstup se zachycenými hesly do souboru v takovém formátu, který je těmto prolamovacím nástrojům srozumitelný.

Triky s přepínačem

Mnoho lidí se domnívá, že když používají na síti přepínač, je pro případného útočníka nemožné zachytit jakékoliv informace. Následující sekce pojednává o metodách zachytávání dat na přepínaných sítích.

Falšování zpráv protokolu ARP (ARP Spoofing)

Při pokusu o zachytávání dat na přepínané síti narazíte na závažný problém: Přepínač bude omezovat datový provoz procházející vaším síťovým segmentem. Přepínač si udržuje vnitřní seznam adres MAC všech počítačů, jejichž připojení na daný port bylo v minulosti zaregistrováno. Datový provoz je tak zasílán na konkrétní port pouze v případě, že cílový počítač je na daném portu připojen. Na mnoha operačních systémech je možné přepsat obsah paměti ARP takovým způsobem, aby adresa MAC vašeho počítače byla spojena s adresou IP výchozí brány. Díky tomu bude veškerý datový provoz určený cílovému hostiteli přenášen na váš počítač. Určitě byste měli ručně přidat záznam s adresou skutečné výchozí brány do tabulky ARP vašeho počítače, aby mohl být datový tok dále posílán funkční výchozí bráně, a také byste měli mít zapnutou funkci přeposílání paketů protokolu IP (IP forwarding). Mnoho sítí založených na kabelovém modemu je náchylných k tomuto typu útoku, protože síť s kabelovým modemem je v podstatě ethernetová síť s kabelovým modemem v roli síťového mostu. Stručně řečeno, na tento typ útoku neexistuje přiměřená obrana. Kabelové modemy nové generace používají jiné mechanismy pro připojení uživatele k síti. Sniffer Dsniff (vyvinutý Dugem Songem) obsahuje přesně pro tento účel i aplikaci s názvem *arpspoof* (původně *arpredirect*). Arpspoof přesměrovává pakety od cílového počítače (nebo všech počítačů) na místním segmentu sítě tím, že falšuje odpovědi protokolu ARP. Tento způsob zachytávání dat je mimořádně efektivní.

Zahlcování tabulky s adresami MAC (MAC Flooding)

Aby byl přepínač schopen plnit svůj účel, potřebuje si udržovat v paměti tabulku adres MAC (Ethernetových adres) počítačů připojených na jeho porty. Pokud se na jednom portu objeví velký počet adres a zaplní takto kapacitu tabulky adres, přepínač ztratí informaci o tom, na kterém portu je připojena adresa MAC počítače oběti. Jedná se o stejnou situaci, jako když je poprvé do sítě připojen nový počítač a přepínač se musí nejprve naučit, kde tato adresa je. Dokud přepínač nezná informa-

ci o portu, na kterém je daný počítač připojen, musí rozesílat kopie rámců určených pro konkrétní adresu MAC na všechny své porty. Tato technika se nazývá *zahlcování* (flooding).

Balík Dsniff obsahuje program s názvem *macof*, který dokáže zahltit přepínač náhodně generovanými adresami MAC. Macof zahlcuje místní síť náhodnými adresami MAC, čímž donutí některé přepínače, aby se přepnuly do odemčeného módu kvůli chybovému stavu. Autorem konverze z původního Perl Net::RawIP macof do jazyka C je Ian Vitek <ian.vitek@infosec.se>.

Hrajeme si se směrováním

Jedním ze způsobů, jak zajistit směrování datové komunikace tak, aby procházela vaším počítačem, je změna směrovací tabulky na hostiteli, jehož komunikaci chcete sledovat. Toho může být dosaženo zasíláním podvržených oznamovacích zpráv směrovače protokolem RIP, kde bude obsažena informace, že váš počítač je výchozí bránou. Pokud se tento záměr podaří realizovat, veškerý datový provoz bude procházet přes váš počítač. Ujistěte se, že máte povoleno přeposílání paketů protokolu IP (IP forwarding), aby zachycená odchozí data mohla být posílána dále na legitimní výchozí bránu. Pokud jste nezměnili směrovací tabulku na výchozí bráně tak, aby veškerá příchozí data byla směrována na váš počítač, nebudete schopni tato data zachytávat.

Ochrana sítě proti zachytávání dat

V tomto bodě možná zvažujete preventivní odpojení celé sítě, protože nechcete dopustit, aby sniffer jako Wireshark (nebo jiná podobně zločinná aplikace) byl použit proti vám. Ještě s těmi štípacími kleštěmi chvíli vydržte: jsou i jiné způsoby, jak zabezpečit síť proti odposlechu, které navíc nezhorší funkčnost vaší sítě.

Použití šifrování

Naštěstí pro bezpečnost sítě existuje šifrování, které pokud je použito správným způsobem, působí jako silný štít, který činí všechny snahy o zachytávání dat zbytečnými. Použití šifrování (za předpokladu, že jeho mechanismus je správný) postaví značnou překážku do cesty útočníkovi, jenž se snaží pasivně monitorovat provoz na vaší síti.

Mnoho existujících síťových protokolů má svůj protějšek, který spoléhá na silné šifrování anebo všezahrnující mechanismus (např. IPSec a OpenVPN), jenž šifrování poskytuje všem protokolům. Bohužel IPSec není v prostředí Internetu využíván zrovna masově, zejména mimo velké firemní sítě.

SSH

Protokol SSH je kryptograficky bezpečnou náhradou standardní implementace protokolu Telnet na unixových systémech, konkrétně příkazů rlogin, Remote Shell (RSH) a Remote Copy Protocol (RCP). Komunikace prostřednictvím protokolu SSH zahrnuje klientský systém a server, který používá šifrování pomocí veřejného klíče k zajištění šifrované komunikace. Navíc poskytuje možnost forwardování libovolných portů protokolu TCP prostřednictvím šifrovaného spojení, jež se hodí zejména pro forwardování X11 Windows a dalších spojení.

Protokolu SSH se dostalo širokého uplatnění jakožto bezpečnostnímu mechanismu pro vzdálený interaktivní přístup k systému. Byl koncipován a vyvinut finským vývojářem Tatu Ylönenem. Původní verze SSH

byla časem transformována v komerční produkt, a i když je původní verze stále dostupná zdarma, licence pro její použití se stala poněkud přísnější. Byly uveřejněny specifikace tohoto protokolu, díky čemuž vzniklo více různých verzí klientských a serverových aplikací kompatibilních s SSH, které nejsou natolik omezeny licencí (asi nejvíce patrné je odstranění licenčních ujednání, která omezují komerční použití). Volně dostupná verze programu kompatibilního s SSH (OpenSSH) vyvinutá v rámci projektu OpenBSD OS je dostupná ke stažení na adrese *www.openssh.com.* Nové komerční SSH je možno zakoupit od firmy SSH Communications Security (*www.ssh.com*), která uvolnila komerční verzi pro renomované univerzity zdarma. Operační systém Mac OS X již obsahuje vestavěné aplikace pro provoz SSH. Zdarma dostupnou alternativou komerčních produktů SSH je také aplikace PuTTy pro operační systém Windows. Přestože byl původně vyvinut pro protokoly používající prostý text, jako je Telnet, stal se mezi administrátory rychle velmi populárním. Můžete jej stáhnout na adrese *www.chiark.greenend.org.uk/~sgtatham/putty.*

SSL

Protokol SSL poskytuje služby pro ověřování a šifrování a může být použit také pro tvorbu virtuálních privátních sítí (VPN). Z hlediska zachytávání dat je protokol SSL zranitelný zejména útoky typu MITM. Útočník může nainstalovat transparentní server proxy mezi vámi a webovým serverem. Tento server proxy může být nakonfigurován tak, aby dešifroval relace protokolu SSL, zachytil data a opět je zašifroval. Pokud k tomuto dojde, uživateli je zobrazena výzva s informací, že certifikát SSL nebyl vystaven důvěryhodnou autoritou. Problémem je opět fakt, že většina uživatelů toto varování ignoruje a pokračují v komunikaci.

Pretty Good Privacy a Secure/Multipurpose Internet Mail Extensions

Pretty Good Privacy (PGP) a Secure/Multipurpose Internet Mail Extensions (S/MIME) jsou standardními protokoly pro šifrování zpráv elektronické pošty. Pokud jsou správně použity, zabraňují zachytávání e-mailových dat programy, jako je dsniff nebo Carnivore, a také interpretaci těchto zpráv. Jak odesílatel, tak příjemce musí používat speciální aplikaci, která umožňuje šifrování i dešifrování zpráv. Ve Spojených státech FBI vyvinula aplikaci typu trojský kůň s názvem „Magic Lantern", která umí zaznamenávat stisknuté klávesy. Tento záznam se později použije k pokusu o získání hesla. Když FBI takto získá heslo, může konečně dešifrovat e-mailovou zprávu. Ve Velké Británii jsou uživatelé povinni dát státním orgánům na vyžádání své šifrovací klíče.

Přepínání

Síťové přepínače činí sledování sítě pro útočníky obtížnější, ale ne příliš. Přepínače jsou někdy doporučovány jako řešení problémů se zachytáváním dat. Avšak jejich skutečný účel je zlepšení výkonu sítě, nikoliv poskytování zabezpečení. Jak bylo vysvětleno v kapitole „Pokročilé techniky zachytávání dat", každý útočník s těmi správnými nástroji může monitorovat data i na přepínané síti, pokud je připojen na stejném segmentu sítě nebo přepínači jako cílový počítač.

Nasazení detekčních metod

Ale co když z nějakého důvodu na vaší síti nemůžete použít šifrování? Co budete dělat? V tomto případě se budete muset spolehnout na odhalování všech síťových karet, které mohou operovat takovým způsobem, jenž by jim umožnil zachytávání dat.

Místní detekce

Mnoho operačních systémů poskytuje mechanismus pro zjišťování, zda síťová karta pracuje v promiskuitním módu. Tato skutečnost je většinou vyjádřena stavem příznaku, který je spojen s každým síťovým rozhraním a je udržován jádrem operačního systému. Tuto informaci můžeme na unixových systémech získat pomocí příkazu **ifconfig**.

Následující příklad ukazuje síťové rozhraní operačního systému Linux, které není nastaveno na promiskuitní mód:

```
eth0 Link encap:Ethernet HWaddr 00:60:08:C5:93:6B
inet addr:10.0.0.21 Bcast:10.0.0.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:1492448 errors:2779 dropped:0 overruns:2779 frame:2779
TX packets:1282868 errors:0 dropped:0 overruns:0 carrier:0
collisions:10575 txqueuelen:100
Interrupt:10 Base address:0x300
```

Všimněte si, že vlastnosti tohoto rozhraní nezmiňují nic o promiskuitním módu. Když rozhraní do promiskuitního módu přepneme (jak je ukázáno na dalším výpisu), objeví se v sekci atributů klíčové slovo **PROMISC**:

```
eth0 Link encap:Ethernet HWaddr 00:60:08:C5:93:6B
inet addr:10.0.0.21 Bcast:10.0.0.255 Mask:255.255.255.0
UP BROADCAST RUNNING PROMISC MULTICAST MTU:1500 Metric:1
RX packets:1492330 errors:2779 dropped:0 overruns:2779 frame:2779
TX packets:1282769 errors:0 dropped:0 overruns:0 carrier:0
collisions:10575 txqueuelen:100
Interrupt:10 Base address:0x300
```

Je důležité se zmínit o tom, že pokud útočník kompromitoval zabezpečení systému, na kterém spouštíte příkaz **ifconfig**, může snadno ovlivnit výstup tohoto příkazu. Důležitou součástí útočníkovy sady nástrojů je náhrada za příkaz **ifconfig**, která nehlásí rozhraní nastavené pro práci v promiskuitním módu.

Síťová detekce

Je mnoho metod různé přesnosti sloužících ke zjištění, zda počítač sleduje veškerý síťový provoz. Ale neexistuje žádná zaručená metoda, jak zjistit přítomnost snifferu.

Dotazy DNS

Většina programů vytvořených za účelem sledování síťového provozu provádí dotazy zpětného vyhledávání v DNS, když produkují výstup, který se skládá ze zdrojového a cílového počítače podílejících se na síťovém připojení. Při procesu provádění vyhledávání v DNS je generován další síťový provoz. Zejména se na něm podílejí dotazy DNS vyhledávající síťové adresy. Je možné sledovat síť a vyhledávat počítače, které vykonávají velký počet dotazů DNS (a téměř žádný jiný datový provoz negenerují). Avšak tento jev se může objevit i náhodně a nemusí vést nutně k odhalení počítače se snifferem. Jednodušší cesta, která vyústí ve 100% přesnost, spočívá v navázání falešného síťového připojení na adresu počítače, který nevykonává žádnou práci na lokální síti. Poté můžete sledovat síť na dotazy DNS, které se pokoušejí přeložit zfalšovanou adresu a které tímto prozradí počítač zachytávající data.

Latence

Druhou metodou, jak odhalit na síti počítač, který monitoruje síťový provoz, je kontrola změn latence v odezvě hostitelů na síti (např. ping). I když je tato technika náchylná k množství chyb (např. když je odezva hostitele ovlivněna běžným provozem), může pomoci ke zjištění, zda nějaký počítač zachytává síťová data. Ideálním způsobem použití je nejprve změřit dobu odezvy podezřelého počítače a zaznamenat naměřené hodnoty. V dalším kroku potom vygenerujte velký objem dat speciálně upravených tak, aby byly pro počítač zachytávající ověřovací údaje zajímavé. Nakonec opět změřte dobu odezvy podezřelého počítače a porovnejte ji s hodnotou naměřenou na začátku. Pokud je zde značný rozdíl, pravděpodobně jste identifikovali počítač se spuštěným snifferem.

Chyby v ovladači

Někdy k detekci síťové karty pracující v promiskuitním módu mohou pomoci i chyby v ovladači síťové karty pro daný operační systém. V jednom případě společnost CORE-SDI (argentinská firma zabývající se výzkumem v oblasti zabezpečení) odhalila chybu v běžném linuxovém ovladači pro ethernetová zařízení. Zjistili, že když počítač s tímto ovladačem pracuje v promiskuitním módu, operační systém není schopen správně provést kontrolu ethernetové adresy, aby se ujistil, že paket byl poslán na jedno z jeho rozhraní. Místo toho kontrola proběhla až na úrovni protokolu IP a paket byl přijat, pokud byl adresován na jedno z hostitelských rozhraní. Za normálních okolností jsou pakety, které neodpovídají ethernetové adrese hostitele, zahazovány na hardwarové úrovni. Avšak v promiskuitním módu se tomu neděje. Můžete odhalit počítač pracující v promiskuitním módu tím, že mu zašlete paket protokolu ICMP (ping) s platnou adresou IP a neplatnou ethernetovou adresou. Pokud počítač na žádost protokolu ICMP odpoví, je zřejmé, že jeho síťová karta je nastavena na promiskuitní mód.

NetMon

Program NetMon, dostupný pro operační systém Windows NT, má schopnost sledovat, kdo aktivně provozuje další instance této aplikace na síti. Také udržuje informace o tom, kdo nainstaloval NetMon na který systém. Je schopen odhalit pouze další kopie programu NetMon, takže pokud útočník používá nějaký jiný sniffer, je nutné použít některou z výše uvedených metod detekce. Většina síťových systémů pro detekci průniku je schopna odhalit běžící instance programu NetMon.

Shrnutí

V této kapitole jste získali přehled o Wiresharku a jeho podpůrných programech. Hovořili jsme o minulosti Wiresharku, jeho kompatibilitě s dalšími sniffery a podporovanými protokoly. Grafickým uživatelským rozhraním Wiresharku a možnostmi filtrování jsme se zatím zabývali jen stručně, protože tyto oblasti budou probrány podrobněji v dalších kapitolách. Také jsme upozornili na programy, které jsou dodávány k Wiresharku, a na jejich rozšiřující funkce pro manipulaci se soubory zachycených dat.

Prozkoumali jsme několik scénářů používání Wiresharku v architektuře vaší sítě. Určitě věnujte čas důkladnému seznámení se s vaší sítí a způsobem zapojení. Znalost segmentace sítě vám umožní nasadit Wireshark na místě vhodném pro zachytávání dat, která potřebujete.

Také jsme prozkoumali široké spektrum rolí, ve kterých může být Wireshark použit síťovými administrátory a administrátory zabezpečení. Wireshark může být také použit dalšími osobami, ne vždy s dobrými úmysly. Zmínili jsme téma zabezpečení a optimalizace Wiresharku pro jeho efektivní nasazení v pracovním procesu. Přestože je Wireshark jako aplikace robustní a stabilní, existuje několik jednoduchým úkonů, které vám pomohou vylepšit jeho výkon.

Nakonec jsme se podívali na příklady metodologie řešení problémů se sítí. Je vhodné tuto metodologii používat při řešení všech typů síťových problémů. Je nutné opakovaně zdůraznit, že poznání vaší sítě a protokolů, které zde pracují, je základním předpokladem pro úspěšné řešení problémů v budoucnosti.

Rychlá řešení

Co je Wireshark?

- Wireshark je volně dostupný síťový analyzátor s množstvím funkcí, který je schopen úspěšně konkurovat komerčním produktům.
- Wireshark umí dekódovat více než 750 protokolů.
- Wireshark je kompatibilní s více než 25 dalšími sniffery a nástroji pro zachytávání dat.
- Pro třídění datové komunikace lze použít zobrazovací filtry nebo filtry pro zachytávání.
- E-mailové konference jsou skvělým zdrojem informací a podpory.
- Wireshark je šířen zdarma a je možné jej volně modifikovat.

Podpůrné programy

- Wireshark je instalován společně s podpůrnými programy (např. tshark).
- Editcap
- Mergecap
- A text2pcap
- Tshark je obdobou aplikace Wireshark ovládanou z příkazové řádky.
- Editcap se používá pro odstraňování paketů ze souboru a pro překlad formátu souboru se zachycenými daty.
- Mergecap slouží ke spojování několika souborů se zachycenými daty.

- Text2pcap konvertuje výpisy ASCII hexdump do výstupních souborů ve formátu libpcap.

Používání Wiresharku ve vaší síťové architektuře

- Správné umístění Wiresharku v architektuře vaší sítě je zcela zásadní pro zachycení potřebných dat.
- Pro připojení Wiresharku k síti lze použít kabelové odbočky, rozbočovače a přepínače s větvením portů.
- Můžete si sestavit vlastní sadu nástrojů pro řešení problémů, sestávající z malého rozbočovače, kabelové odbočky a přímého i kříženého síťového kabelu.
- Instalace Wiresharku na laptop usnadňuje práci v terénu.

Řešení systémových problémů a problémů se zabezpečením

- Dodržováním systematického procesu řešení problémů můžete výrazně snížit množství času potřebného pro vyřešení problému.
- Identifikace a ověřování příčiny problému často zahrnuje i prohledávání informačních zdrojů na Internetu nebo kontaktování pracovníků technické podpory příslušného výrobce hardwaru a softwaru.
- Někdy vyřešení jednoho problému způsobí problém jiný.
- Detailní záznamy o postupech vedoucích k odstranění problému mohou být při řešení problémů v budoucnosti velmi užitečné.

Zabezpečení a optimalizace Wiresharku

- Zachytávejte pakety s použitím přiměřených oprávnění; zachycená data však analyzujte s použitím nejnižších možných oprávnění.
- Když se objeví chyba v zabezpečení Wiresharku, nainstalujte pro tento produkt aktualizaci.
- Lepší doby odezvy aplikace při analýze velkého množství paketů docílíte přidáním systémové paměti do počítače.

Pokročilé techniky zachytávání dat

- Techniky pro manipulaci s MAC a ARP mohou být efektivně využity pro zachytávání dat na přepínané síti, a to beze změn ve stávající architektuře.
- Útoky typu MITM mohou být použity pro zachycení datové komunikace.
- Existují nástroje, které mohou zachytávat data a zároveň prolamovat zachycená hesla.

Ochrana sítě proti zachytávání dat

- Pokud jsou dva počítače navzájem ověřené, může šifrované spojení VPN typu host-to-host efektivně skrýt přenášená data před sniffery. V závislosti na použitém způsobu mohou být skryta i čísla portů a protokoly.
- Šifrování dat na aplikační úrovni například prostřednictvím protokolu SSL chrání také data obsažená v paketech.

- Zachytávání dat na přepínaných sítích je o něco obtížnější než na sítích, které používají rozbočovače.

Nasazení detekčních metod

- Nastavení promiskuitního módu na síťovém rozhraní počítače může indikovat běžící sniffer.
- Některé sniffery je možné odhalit pomocí jejich vlivu na chování sítě, jako je nadměrný výskyt dotazů DNS, zvýšení doby odezvy, problémy s ovladači i samotnými aplikacemi.
- Žádný nástroj pro odhalení snifferu není sám o sobě dostatečně efektivní, pokud nemá podporu v zásadách zabezpečení, které obsahují instrukce pro přiměřené techniky zachytávání dat v síti.

Časté dotazy

Následující často kladené otázky byly zodpovězeny autorem této publikace a jsou zde umístěny proto, abyste mohli zhodnotit své porozumění danému tématu, a také vám pomohou při uplatnění teoretických znalostí v podmínkách reálného prostředí.

Otázka: V minulosti se už mnoho open-source produktů stalo komerčními. Co se stane, když se tým vývojářů kolem Wiresharku rozhodne nadále vyvíjet Wireshark jako komerční produkt? Bude Wireshark i nadále dostupný zdarma?

Odpověď: Wireshark byl vyvinut a uvolněn pod licencí GNU GPL; zdrojový kód aplikace vyvinuté pod licencí GPL bude vždy dostupný zdarma.

Otázka: Proč bych měl používat právě Wireshark, když je na trhu množství dalších komerčních produktů, které má firma upřednostňuje?

Odpověď: Wireshark vás samozřejmě nenutí k tomu, abyste jej používali. Každopádně není nikdy na škodu mít jej k dispozici ve své sadě nástrojů pro řešení problémů.

Otázka: Myslím, že jsem na své síti nalezl vetřelce, a rád bych ochránil firemní data. Může mi v tomto Wireshark pomoci?

Odpověď: Nejlepší reakcí na takovouto situaci je postup definovaný plánem reakce na bezpečnostní incidenty, který by vaše firma měla mít vypracován právě pro tyto případy. Pokud jej nemá, vězte, že nejlepší čas pro vypracování takového plánu je předtím, než jej budete potřebovat.

Otázka: Jak můžu vědět, že verze Wiresharku, kterou jsem si stáhl z Internetu, neobsahuje virus nebo nějaký jiný zákeřný program?

Odpověď: Dobrým začátkem je rozhodně stahování z nějakého renomovaného webového sídla. Ale ať už stáhnete program odkudkoliv, zkontrolujte jeho pravost pomocí nástrojů **md5sum** a **sha1sum**, které spustíte na staženém souboru. Zkontrolujte hashe získané těmito programy s hashi dostupnými v souboru SIGNATURE v adresáři Wiresharku. Měli byste také ověřit pravost samotných hashů, a to třeba pomocí nástroje GnuPG (www.gnupg.org).

Otázka: Jak mohu vytvořit pakety, které bych pak Wiresharkem zachytával?

Odpověď: Například s použitím utility ping nebo tím, že navštívíte nějakou webovou stránku. Pokud chcete vidět specifický druh datového provozu, bude nejlepší, když si vytvoříte prostředí, ve kterém k takovému druhu provozu pravděpodobně dojde. Jinak řečeno: Chcete-li zachytávat datovou komunikaci webových protokolů, nainstalujte si webový server a webový prohlížeč. Jestliže vám jde o vytvoření speciálně upravených paketů, můžete použít jeden z modulů jazyka Perl s názvem **Raw::IP**. Můžete jej stáhnout na adrese *www.ic.al.lg.ua/~ksv/index.shtml.*

Otázka: Jak mohu zachytávat všechna data ve své síti, když přepínač zachytávání nepodporuje nebo je nespravovaný?

Odpověď: Tak či onak, budete se muset dostat k síťové cestě, kterou data putují. Buď můžete spustit sniffer na počítači, který vidí alespoň většinu síťového provozu, nahradit přepínač rozbočovačem nebo přepínačem s podporou zachytávání dat, případně trošku experimentovat s protokolem ARP pomocí nástroje dsniff.

Otázka: Je možné používat Wireshark, aniž bych jej musel instalovat?

Odpověď: Dobrou volbou je použití bootovatelného CD nebo DVD. Když si stáhnete a vypálíte obraz disku, jako např. Backtrack (*www.remote-exploit.org*) nebo Helix (*www.e-fense.com/helix/*), můžete používat Wireshark i další bezpečnostní nástroje bez nutnosti instalace. Avšak tato bootovatelná média by rozhodně neměla být používána k porušování bezpečnostních pravidel vaší organizace definujících používání programů třetích stran.