CARNet

HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

# Analiza alata Wireshark

## NCERT-PUBDOC-2010-09-312

Nacionalni CERT

u suradnji s

LSS

Sigurnosni problemi u računalnim programima i operativnim sustavima područje je na kojem Nacionalni CERT kontinuirano radi.

Rezultat toga rada je i ovaj dokument, koji je nastao suradnjom Nacionalnog CERT–a i LS&S–a, a za koji se nadamo se da će Vam koristiti u poboljšanju sigurnosti Vašeg sustava.

## **Nacionalni CERT**, www.cert.hr

Nacionalno središte za **sigurnost računalnih mreža** i sustava.

## **LS&S**, www.LSS.hr

Laboratorij za sustave i signale pri Zavodu za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu.

# Sadržaj

# 1. Uvod

Programski alat Wireshark koristi se za analizu mrežnih paketa. Radi se o alatu koji hvata podatke koji u paketima putuju mrežom i prikazuje ih na najdetaljniji mogući način. U prošlosti, alati slični Wiresharku su bili skupi i najčešće komercijalni. Dolaskom alata Wireshark na tržište situacija se promijenila. Wireshark je danas vjerojatno najbolji besplatni i *open source* alat dostupan na tržištu. Neki od primjera korištenja ovog alata su:

- otklanjanje problema na mreži,
- analiza sigurnosnih ranjivosti,
- razvoj i implementacija novih protokola te
- učenje o mrežnim protokolima.

Wireshark je tzv. „*cross-platform*" mrežni alat, što znači da može raditi na različitim platformama. Osim što radi na operacijskom sustavu Microsoft Windows, podržan je i na različitim Unix operacijskim sustavima među kojima su Linux, Mac OS X, BSD i Solaris. Također, postoji i inačica bez grafičkog sučelja (eng. *Graphical User Interface*) nazvana TShark. Wireshark i TShark su besplatni alati pod uvjetima GNU General Public licence (najraširenija licenca za slobodan softver). U ovom dokumentu su opisane mogućnosti ovog moćnog alata zajedno s odgovarajućim primjerima korištenja. Osim toga, dana je usporedba Wiresharka sa sličnim alatima, kao i pregled njegovih sigurnosnih problema i ranjivosti.

***Slika 1. Logo alata Wireshark***
***Izvor: Onlineitclass's Blog***

# 2. Povijest i razvoj alata *Wireshark*

1997. godine Gerald Combs je zbog potrebe za alatom za praćenje prometa na mreži i želje da nauči više o upravljanju i administriranju mreže počeo pisati program zvan Ethereal, preteču današnjeg Wiresharka. Alatom Ethereal htio je riješiti oba gore navedena problema. Ethereal je inicijalno pušten na tržište, nakon nekoliko stanki u razvoju, u srpnju 1998. godine u inačici 0.2.0. Vremenom se Ethereal nadogradio, ispravljene su postojeće greške i polako se počeo nazirati uspjeh projekta. Nedugo nakon toga, sistemski inženjer Gilbert Ramirez je uočio potencijal Ethereala i posao prvi suradnik projekta u kojeg je implementirao nekoliko nadogradnji.

U listopadu 1998. godine, Guy Harris iz tvrtke Network Appliance pokušao je pronaći bolji alat za administriranje mreže od dotad korištenog alata tcpview, te je također počeo razvijati zakrpe i poboljšanja za Ethereal. Krajem 1998. godine i Richard Sharpe, stručnjak s područja TCP/IP protokola, je uočio potencijal ovog alata te počeo pisati zakrpe i unaprjeđenja za protokole koji su mu bili potrebni. Do danas se lista ljudi koji su pridonijeli razvitku alat znatno povećala [1]. Većina tih ljudi je započela s novim protokolima koji su im bili potrebni, a koje Ethereal, ili kasnije Wireshark, nisu još podržavali. To je dovelo do velikog broja protokola koje Wireshark podržava danas.

2006. godine projekt se restrukturirao pod današnjim imenom Wireshark. U proljeće 2008. godine, nakon deset godina razvoja, Wireshark je napokon izašao u inačici 1.0. Ta inačica je bila prva potpuna inačica koja je izašla na tržište, ali je isto tako bila i inačica s minimalnim značajkama. Dakle, predstavljala je osnovnu inačicu s mogućnošću nadogradnje. Inačica 1.0. izašla je istovremeno s održavanjem prve Wireshark konferencije za programere i korisnike nazvale SharkFest. Magazin eWEEK (eng. *The Enterprise Newsweekly*, tjedni poslovni informatički magazin) je proglasio Wireshark „najutjecajnijom *open source* aplikacijom svih vremena".

Mogućnosti alata Wireshark su različite, a najbitnije, koje i njegov proizvođač ističe, su:

- hvatanje podatkovnih paketa s mrežnog sučelja,
- prikazivanje paketa s vrlo detaljnim informacijama o mrežnom protokolu,
- otvaranje i spremanje paketa,
- uvoz i izvoz podataka u druge slične programe,
- pretraga i filtriranje paketa po raznolikim kriterijima i
- kreiranje različitih statistika.

***Slika 2. Logo alata Ethereal***
***Izvor: Ethereal***

Kao što je već spomenuto, Wireshark je izdan pod GNU GPL licencom. Sav izvorni kod se može besplatno preuzeti na web stranicama projekta [2].

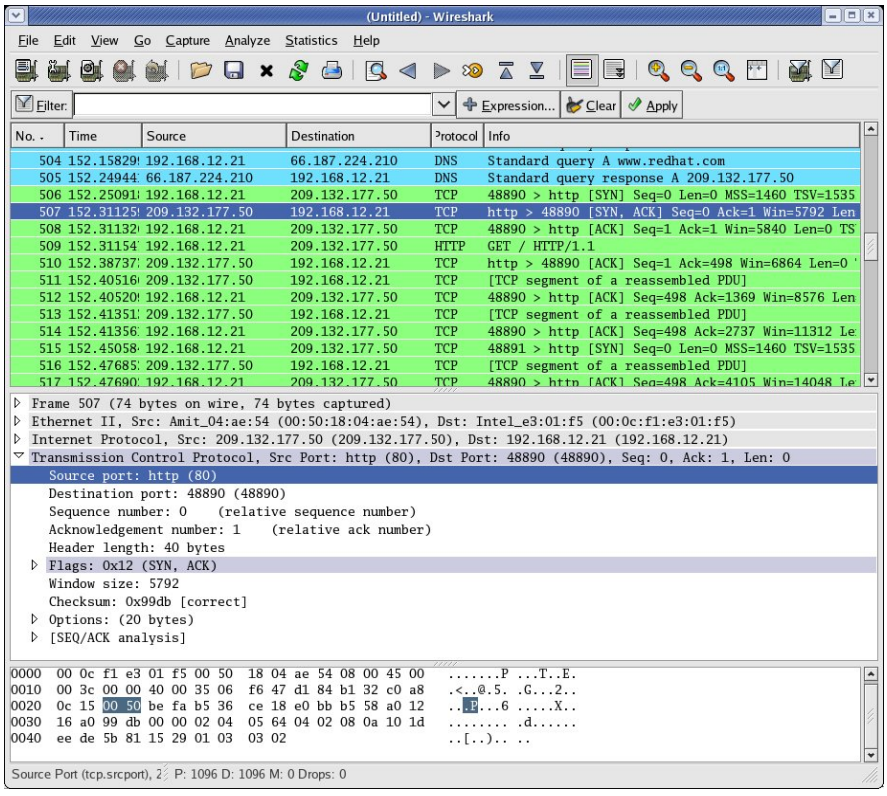
Gotovo svi dijelovi Wiresharka su implementirani u programskom jeziku C. Osim uobičajenog razvoja programa u jeziku C, neki programski alati, kasnije dodavani u Wireshark, napisani su u drugim programskim jezicima. Neki od tih alata, odnosno programskih jezika, su:

- *Perl* - služi za izradu dokumentacije,
- *Pyton* i *Sed* - mogu poslužiti za generiranje nekih protokola, funkcija ili biblioteka te
- *Flex* i *Bison* - mogu se koristiti pri izradi biblioteka.


***Slika 3. Logo GNU General Public Licence***
***Izvor: General Public Licence***

# 3. Način rada i mogućnosti alata *Wireshark*

Wireshark je softverski alat koji „razumije“ strukturu različitih mrežnih protokola. Iz tog razloga sposoban je prikazati podatke iz paketa specifičnih za različite protokole. Wireshark koristi biblioteku koda pcap (eng. *Packet capture*) za hvatanje paketa, što znači da može hvatati samo pakete s mreža koje pcap podržava (Ethernet, IEEE 802.11, …). Pcap je biblioteka koja raznim programima pruža programsko sučelje (eng. API – Application Programming Interface) za dohvaćanje paketa s mrežnih sučelja na operacijskim sustavima Windows i Linux/Unix. Podaci se mogu uhvatiti izravno s aktivne mrežne veze ili se mogu učitati iz datoteke u kojoj su pohranjeni već uhvaćeni paketi. Uhvaćeni podaci mogu biti prikazani preko grafičkog korisničkog sučelja ili preko terminala (komandne linije) kod korištenja TSharka. Podaci se mogu programski uređivati preko komandne linije ili pomoću potprograma „*editcap*“. Wireshark sadrži i filter za prikaz podataka pomoću kojega se može prikazati i samo dio podataka, ovisno o uvjetu filtriranja. Budući da je Wireshark *open source* alat, relativno je jednostavno implementirati programske dodatke za nove protokole.

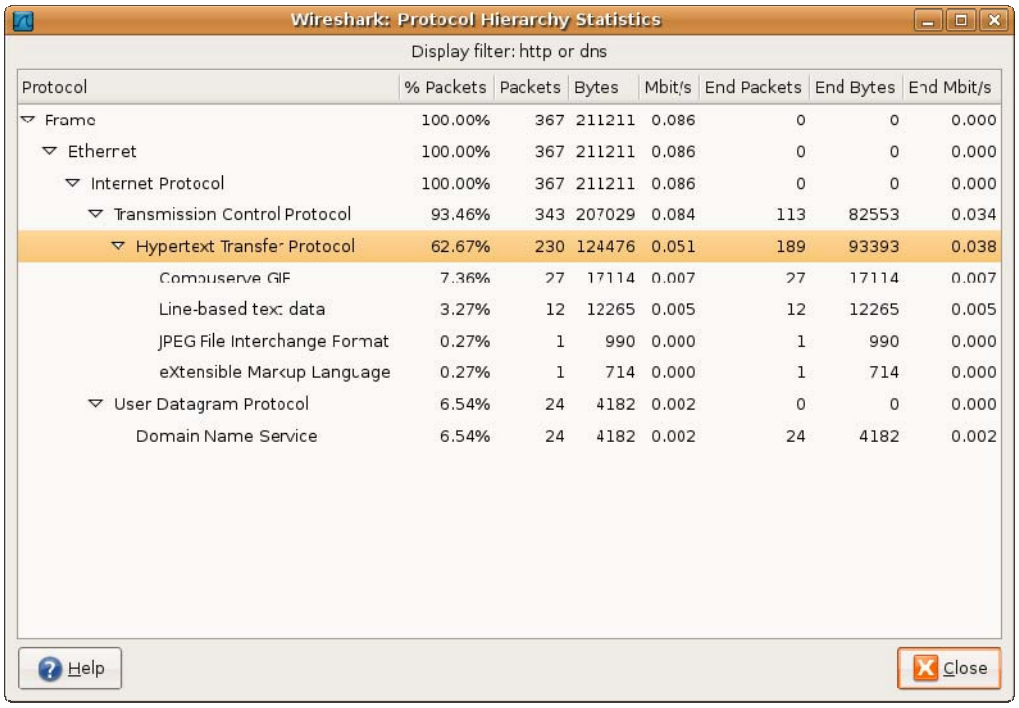



***Slika 4. Izgled grafičkog korisničkog sučelja Wiresharka***
***Izvor: Wireshark***

Podatke unutar mrežnih paketa Wireshark može očitati s više različitih vrsta mreža, a najpoznatije koje podržava su:

- **Ethernet** – najučestalija LAN (eng. *Local Area Networking*) tehnologija koja se, s 10 gigabitnom izvedbom, koristi i kao WAN (eng. *Wide Area Networking*) tehnologija. Ethernet šalje pakete od pošiljaoca prema jednom (*Unicast*) ili više (*Multicast/Broadcast*) prijamnika.

- **IEEE 802.11** – skup standarda za bežičnu računalnu komunikaciju (WLAN, eng. *Wireless Local Area Network*) na frekvencijskim pojasevima od 2.4, 3.6 i 5 GHz. Razvijen je od strane IEEE LAN/MAN Standards Committee (IEEE 802).
- **PPP** – (eng. *Point-to-Point Protocol*) protokol koji se koristi za izravno povezivanje dvaju čvorova računalne mreže. Omogućuje povezivanje računala serijskim, telefonskim ili optičkim kabelom, pomoću mobilnih telefona te posebno oblikovanom radio ili satelitskom vezom.
- **Loop-back** – virtualno mrežno sučelje implementirano softverski.




***Slika 5. Wiresharkov prozor „Protocol Hierarchy“***
***Izvor: Wireshark***

Format datoteke za spremanje paketa uhvaćenih na mreži je standardni *libpcap* format podržan od strane mrežnih biblioteka Libpcap i WinPcap. To znači da Wireshark može pročitati i podatke iz aplikacija kao što su tcpdump i CA NetMaster koje također koriste isti format. Osim toga, podatke uhvaćene Wiresharkom može se pročitati i s drugim aplikacijama koje koriste Libpcap i WinPcap za čitanje uhvaćenih podataka. Tako Wireshark može čitati i podatke uhvaćene mrežnim analizatorima kao što su Snoop, Network General's Sniffer te Microsoft Network Monitor.

Neke od raširenijih porodica protokola koji se koriste u komunikacijskim mrežama, a koje Wireshark podržava su:

- Internet protokoli – TCP/IP skup protokola koji uključuju ARP, IP, TCP, itd.
- Protokoli mobilne telefonije – skup protokola sadržanih u GSM-u (WCDMA, CDMA2000,…).
- VOIP protokoli – skup protokola za prijenos zvuka mrežom (SIP, H323,…).
- WAP protokoli – skup WAP protokola za omogućavanje servisa na bežičnim komunikacijskim mrežama (WTP, WSP,..).

Više o vrstama protokola koje Wireshark podržava može se naći na stranicama alata:

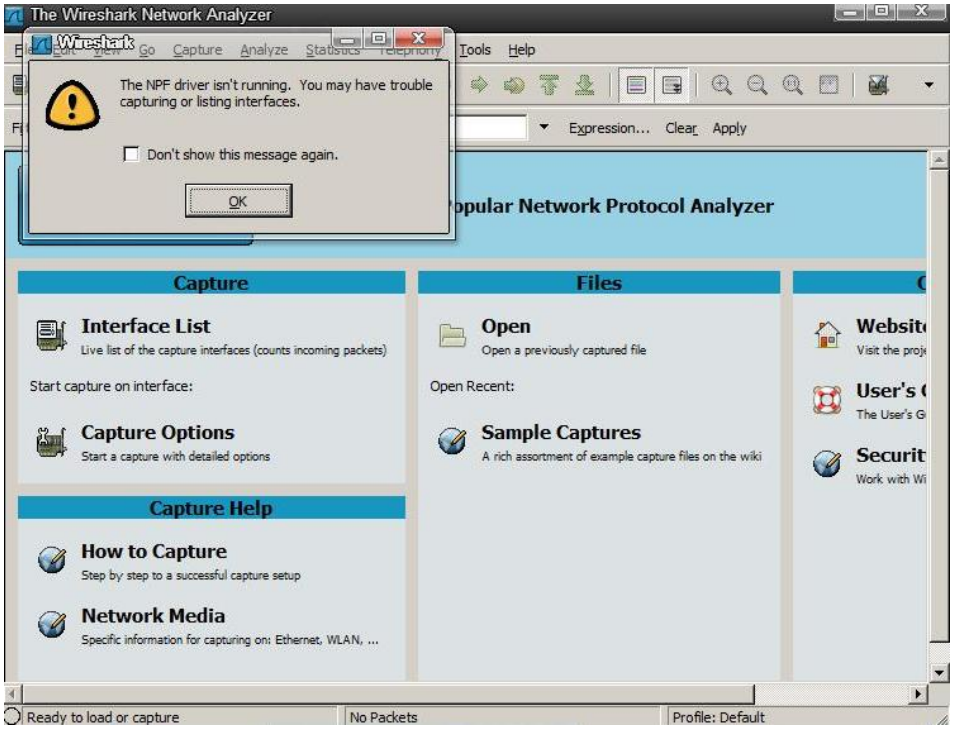
**http://wiki.wireshark.org/ProtocolReference**

```
117 packets captured
   1: 22:19:40.395762 802.1Q vlan#1 P0 192.168.68.254.23 > 192.168.68.105.35736: P 652770102:652770104(2) ack 694381841 win 8192
   2: 22:19:40.396128 802.1Q vlan#1 P0 192.168.68.105.35736 > 192.168.68.254.23: . ack 652770104 win 8576
   3: 22:19:40.396204 802.1Q vlan#1 P0 192.168.68.254.23 > 192.168.68.105.35736: P 652770104:652770113(9) ack 694381841 win 8192
   4: 22:19:40.396433 802.1Q vlan#1 P0 192.168.68.105.35736 > 192.168.68.254.23: . ack 652770113 win 8576
   5: 22:19:48.444984 802.1Q vlan#1 P0 192.168.68.40.49859 > 192.168.68.254.161:  udp 35
   6: 22:19:48.445640 802.1Q vlan#1 P0 192.168.68.254.161 > 192.168.68.40.49859:  udp 90
   7: 22:19:48.447074 802.1Q vlan#1 P0 192.168.68.40.49859 > 192.168.68.254.161:  udp 42
   8: 22:19:48.447425 802.1Q vlan#1 P0 192.168.68.254.161 > 192.168.68.40.49859:  udp 46
   9: 22:19:48.456336 802.1Q vlan#1 P0 192.168.68.40.49859 > 192.168.68.254.161:  udp 191
  10: 22:19:48.457938 802.1Q vlan#1 P0 192.168.68.254.161 > 192.168.68.40.49859:  udp 232
  11: 22:19:48.458472 802.1Q vlan#1 P0 192.168.68.40.49859 > 192.168.68.254.161:  udp 186
  12: 22:19:48.459876 802.1Q vlan#1 P0 192.168.68.254.161 > 192.168.68.40.49859:  udp 212
```

***Slika 6. Format datoteke 'libpcap' za spremanje uhvaćenih paketa***
***Izvor: Layer3.worldpress***

# 4. *Wireshark* u funkciji sigurnosti

Kao i nekim drugim alatima za analizu mreže, tako je i Wiresharkom moguće detektirati neke sigurnosne propuste i nepravilnosti. Sigurnosni propust se očituje u neovlaštenim, nedopuštenim, odnosno malicioznim radnjama koje mogu naštetiti mreži i njenim korisnicima. Wireshark sprječava sigurnosne propuste analizom mogućih problema i radnji koje mogu stvoriti probleme. Zadaci analize unutar Wireshark alata dijele se na preventivne i reaktivne. Preventivni zadaci (metode) uključuju *„baselining"* mrežne metode (najprimitivnija metoda za analizu mrežnih performansi) za očitavanje trenutnog statusa mreže i aplikacije. Preventivne metode se također mogu koristiti za uočavanje problema na mreži prije nego što ih korisnik mreže osjeti. Kao primjer toga, preventivne metode omogućuju uočavanje gubitka paketa prije nego taj gubitak počne utjecati na mrežnu komunikaciju i time se izbjegava problem prije nego je uočen od strane korisnika. Reaktivne metode analize koriste se nakon što su greške u radu mreže uočene. Primjerice, ukoliko postoji problem s nekim poslužiteljem, Wireshark će problem prijaviti tek nakon što pokuša uhvatiti pakete s mreže. Nažalost, u Wiresharku su još uvijek reaktivne analize zastupljenije od preventivnih što je loše jer reaktivne analize uočavaju problem prije nego on može utjecati na mrežu i korisnika, dok kod preventivnih to nije slučaj.




***Slika 7. Primjer reaktivnog zadatka u Wiresharku***
***Izvor: BootLand***

Neke analize koje korisnicima Wiresharka mogu poslužiti u funkciji sigurnosti i administracije mreže su:

- pronalaženje korisnika s najviše prometa na mreži,
- identificiranje protokola i aplikacija koje se trenutno koriste,
- određivanje prosječnog broja paketa u sekundi, prosječnog broja bajtova u sekundi ili ukupnog prometa na mreži,
- prikaz svih korisnika komunikacijske mreže,
- određivanje duljine paketa kojeg koristi aplikacija za prijenos podataka na mreži,
- prepoznavanje najčešćih problema na mreži (spora mreža, neprepoznavanje korisnika,…),
- prepoznavanje kašnjenja između korisničkog naloga za rad s mrežnim paketima i samog procesa rada s paketima,
- prepoznavanje krivo konfiguriranih korisnika (npr. dupliciranа IP adresa),
- određivanje mreže ili korisnika koji usporavaju promet na mreži,
- identificiranje asinkronog prijenosa na mreži,
- identificiranje neuobičajenog pregleda prometa na mreži,
- brzo identificiranje HTTP (eng. *HyperText Transfer Protocol*) grešaka koje indiciraju probleme korisnicima i poslužitelju.
- brzo identificiranje VoIP (eng. *Voice over Internet Protocol*) grešaka koje indiciraju probleme korisniku ili poslužitelju te globalne pogreške.
- izgradnja grafikona za usporedbu ponašanja prometa na mreži,
- izgradnja grafikona prometa aplikacije te usporedba s ukupnim prometom na mreži,
- identificiranje aplikacija koje ne šifriraju podatke koji se prenose,
- uočavanje neuobičajenih protokola,
- identificiranje prosječnog i neprihvatljivog vremena odziva mrežnih servisa (SRT, eng. *Service Response Time*) te
- izgradnja grafikona intervala periodičnog generiranja paketa aplikacija ili protokola.

Mreže jako variraju u prometu kojeg mogu podržati. Broj i vrsta analitičkih zadataka koji se mogu izvršiti na nekoj mreži ovisi o svojstvima prometa na mreži, tj. o vrsti prometa koji mreža može podržati.

# 5. Primjeri korištenja alata *Wireshark*

Wireshark je mrežni analizator i kao takav služi za hvatanje i analizu mrežnih paketa. Njegove mogućnosti svode se na:

- hvatanje mrežnih paketa *(eng. Capturing live network data),*
- uvoz/izvoz i ispis podataka *(eng. File input/output and printing)* te
- rad s uhvaćenim paketima *(eng. Working with captured packets).*

## 5.1. Hvatanje mrežnih paketa

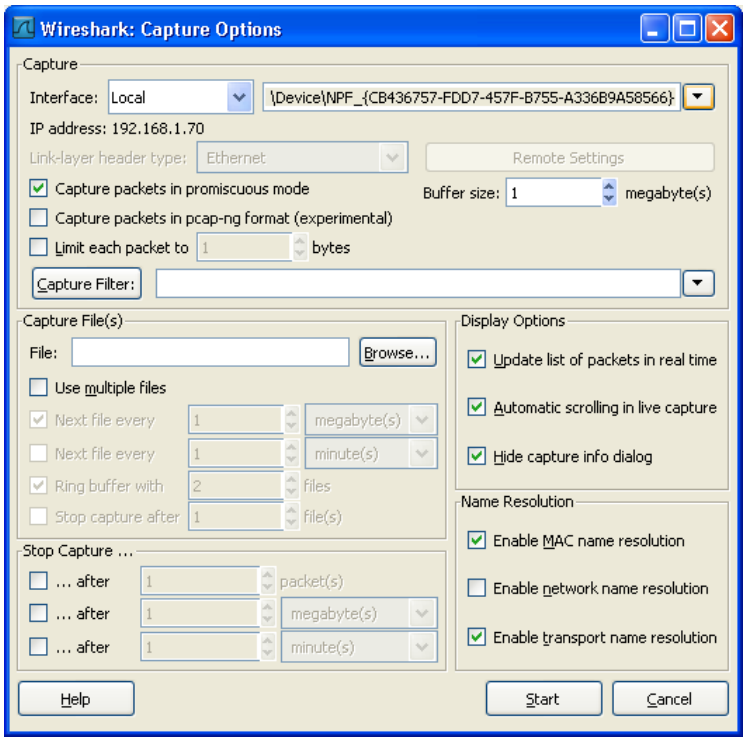
Hvatanje mrežnih paketa je jedna od glavnih značajki Wiresharka. Njegov mehanizam za hvatanje omogućuje:

- hvatanje s različitih vrsta mreža (Ethernet, TokenRing, ATM, …),
- prekid hvatanja uz različite uvjete (količina uhvaćenih podataka, vrijeme hvatanja, broj uhvaćenih paketa, …),
- simultano prikazivanje dekodiranih istovremeno s hvatanjem novih paketa,
- filtriranje paketa i
- reduciranje količine uhvaćenih podataka.

Hvatanje mrežnih paketa se odvija na vrlo jednostavan način. Označavanjem opcije „*Interfaces…*“ iz izbornika „*Capture*“ otvara se prozor „*Capture Interfaces*“ prikazan na slici 8.

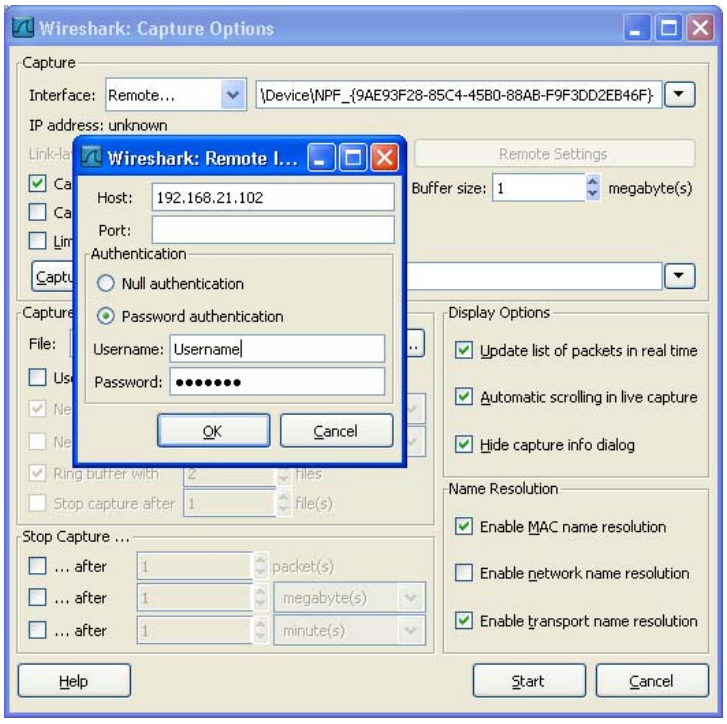

***Slika 8. „Capture Interfaces“ prozor na Microsoft Windows operacijskom sustavu***
***Izvor: Wireshark***

Odabirom opcije „*Start*“ pokreće se hvatanje paketa, dok se odabirom opcije „*Options*“ pristupa prozoru „*Capture Options*“ prikazanom na slici 9. Prozor „*Capture Options*“ služi za određivanje vrste mreže s koje s hvataju podaci te za uređivanje različitih prekidača za prekid hvatanja uz određeni uvjet.



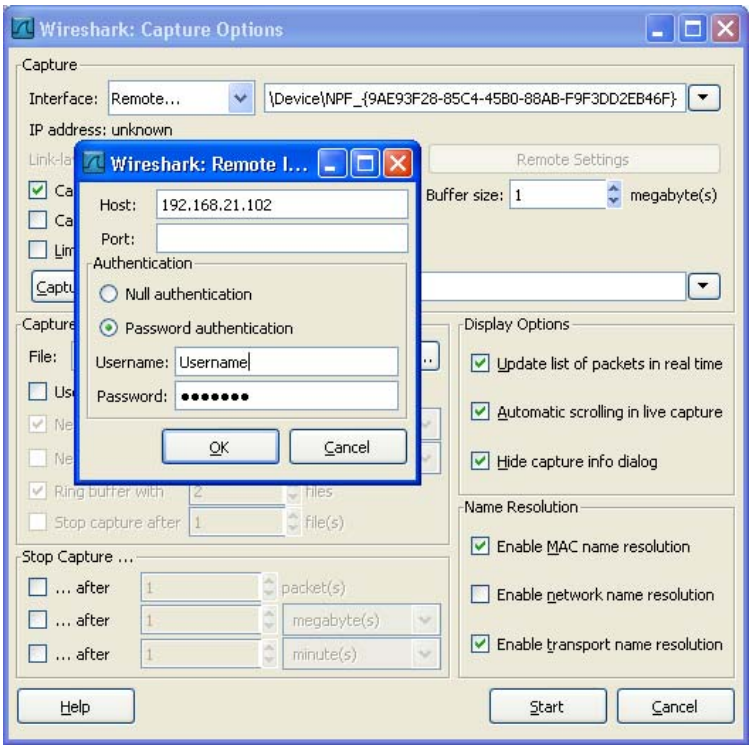

***Slika 9. Prozor „Capture Options" na operacijskom sustavu Microsoft Windows***
***Izvor: Wireshark***

Osim hvatanja podataka s lokalne mreže, Wireshark je sposoban hvatati podatke i sa udaljenih mrežnih sučelja. Na operacijskom sustavu Microsoft Windows tome služi tzv. *„Capture Daemon"* servis, dok se na operacijskim sustavima Unix/Linux isti učinak postiže preko SSH tunela (eng. *Secure Shell*). Za hvatanje podataka s udaljenih sučelja Wireshark koristi opciju *„Remote Capture Interfaces"* prikazanu na slici 10. U tom se prozoru, upisom udaljene IP adrese, pristupa mrežnom prometu i tada se, na isti način kao i kod lokalne mreže, mogu hvatati i analizirati mrežni podaci.

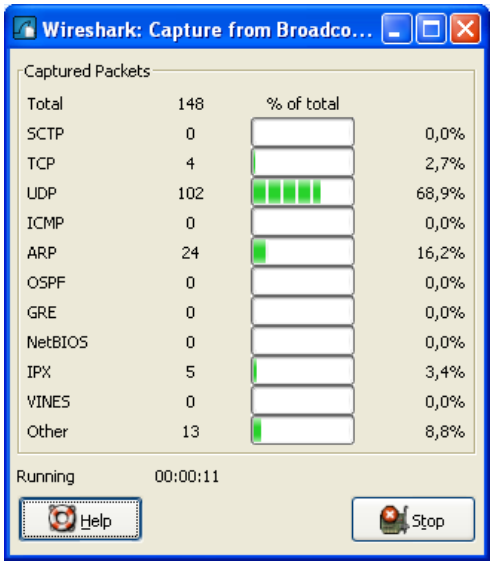



***Slika 10. Prozor „Remote Capture Interfaces" na operacijskom sustavu Microsoft Windows***
***Izvor: Wireshark***

Tokom hvatanja podataka na sučelju Wiresharka prikazuje se prozor „*Capture Info*“ prikazan na slici 11. „*Capture Info*“ služi za informiranje korisnika o broju uhvaćenih paketa kao i o vremenu proteklom od početka hvatanja. Osim toga, na samom prozoru je moguće prekinuti hvatanje podataka ili ga samo pauzirati.




***Slika 11. Prozor „Remote Capture Interfaces“ na operacijskom sustavu Microsoft Windows***
***Izvor: Wireshark***

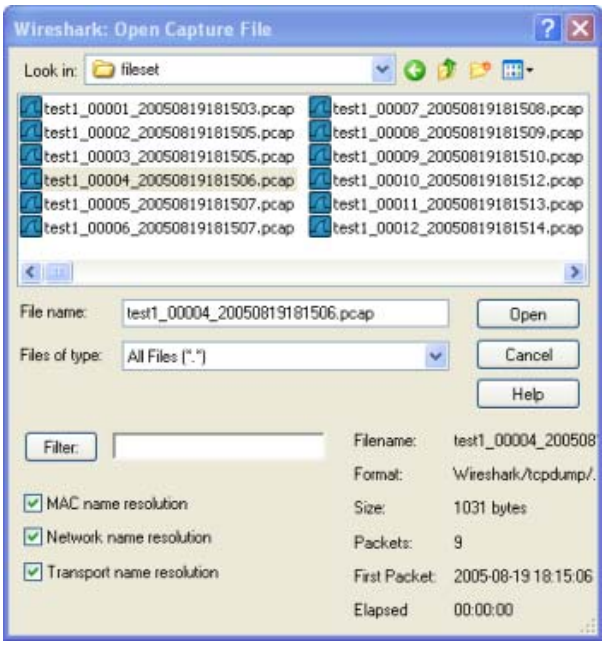



***Slika 12. Prozor „Capture Info“ prozor na operacijskom sustavu Microsoft Windows***
***Izvor: Wireshark***

## 5.2. Uvoz , izvoz i ispis podataka

Wireshark može pročitati i podatke iz uhvaćenih mrežnih paketa spremljenih u datoteku. Čitanje se obavlja intuitivno, u izborniku „*File/Open*“, čime se otvara prozor „*Open Capture File*“ prikazan na slici 12. Wireshark može pročitati razne formate datoteka iz drugih alata za mrežnu analizu, a najrašireniji i najpoznatiji među njima su:

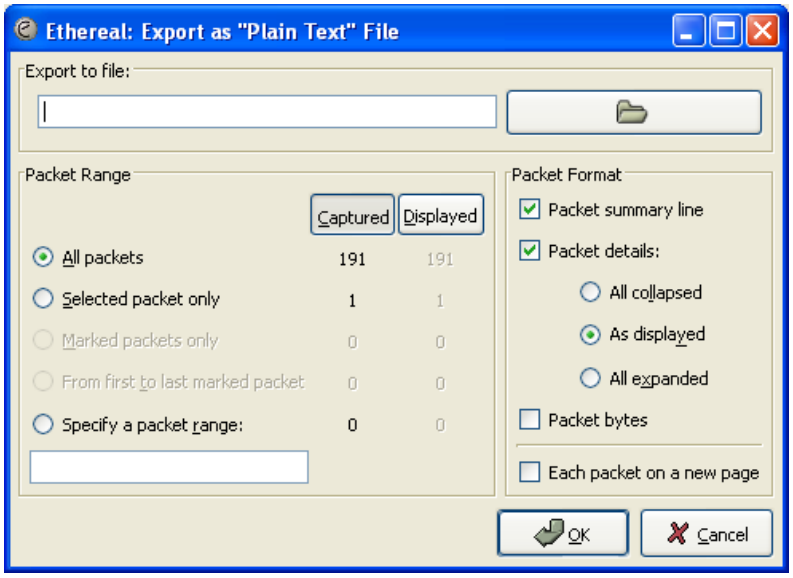
- *libpcap*, *tcpdump* i drugi alati koji koriste *tcpdump* format za hvatanje,
- *sun*, *snoop* i *atmsnoop*,
- Microsoft Network Monitor,
- Novell LANalyzer,
- Shomiti/Finisar Surveyor i mnogi drugi.




***Slika 13. „Open Capture File“ prozor na operacijskom sustavu Microsoft Windows***
***Izvor: Wireshark***

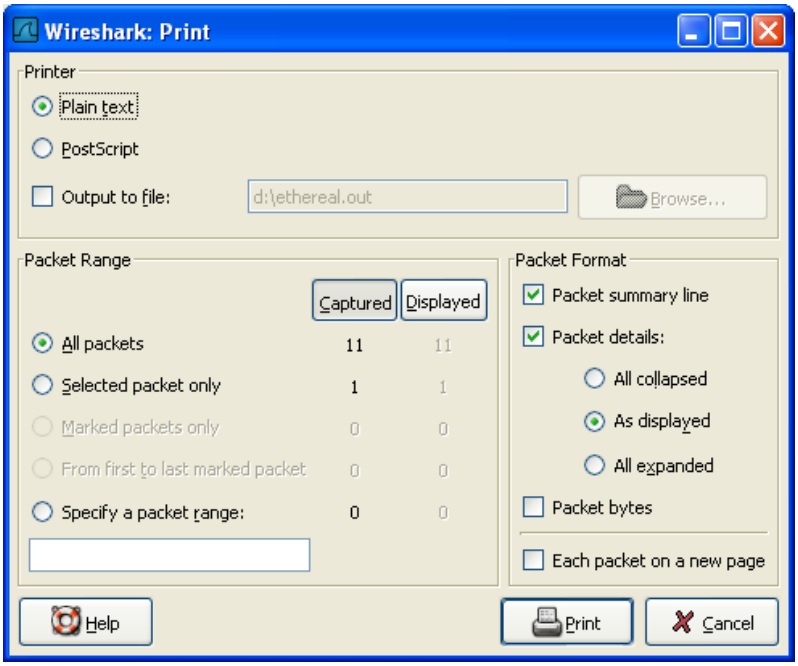
Wireshark podržava više formata za izvoz podataka. Najjednostavniji i najčešće korišteni način izvoza podataka je u ASCII tekstualnom formatu prikazanom na slici 13. Takav način izvoza podataka u Wiresharku je moguće napraviti odabirom opcije „*File/Export*“. Izvoz podataka iz Wiresharka moguć je i u drugim formatima, a oni su:

- PostScript,
- CSV (eng. *Comma Separated Values*),
- polja programskog jezika C,
- PSML i
- PDML.




***Slika 14. Prozor „Plain Text" na operacijskom sustavu Microsoft Windows***
***Izvor: Wireshark***

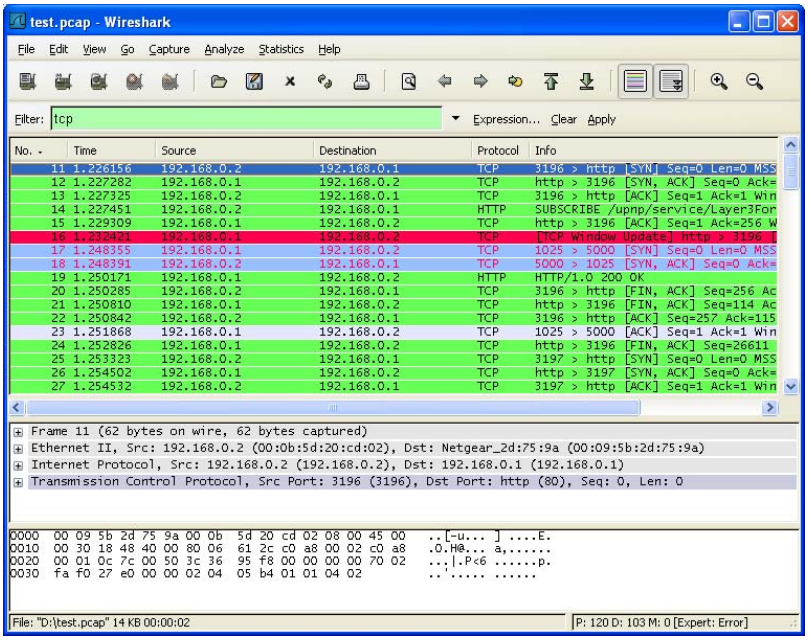
Wireshark ima i opciju ispisa uhvaćenih paketa. Odabirom opcije *„File/Print…"* otvara se prozor za ispis paketa prikazan na slici 14.




***Slika 15. „Print" prozor na operacijskom sustavu Microsoft Windows***
***Izvor: Wireshark***

## *5.3.* Rad s uhvaćenim paketima

Rad s uhvaćenim paketima u Wiresharku obuhvaća prikaz uhvaćenih paketa, njihovo filtriranje, označavanje ili ignoriranje te mnoge druge opcije. Wireshark pakete prikazuje u korisničkom sučelju prikazanom na slici 15.




***Slika 16. Prikaz uhvaćenih paketa u Wireshark-u***
***Izvor: Wireshark***

Uhvaćene pakete moguće je filtrirati po mnogim uvjetima. Wireshark omogućava izradu vlastitih filtera i uvjeta filtriranja te njihovo spremanje i kasnije korištenje. Izrada vlastitog filtra obavlja se unutar prozora „*Filter Expression*“ prikazanog na slici 16. Ugrađeni filtri unutar alata Wireshark omogućuju selekciju paketa po uvjetima kao što su:

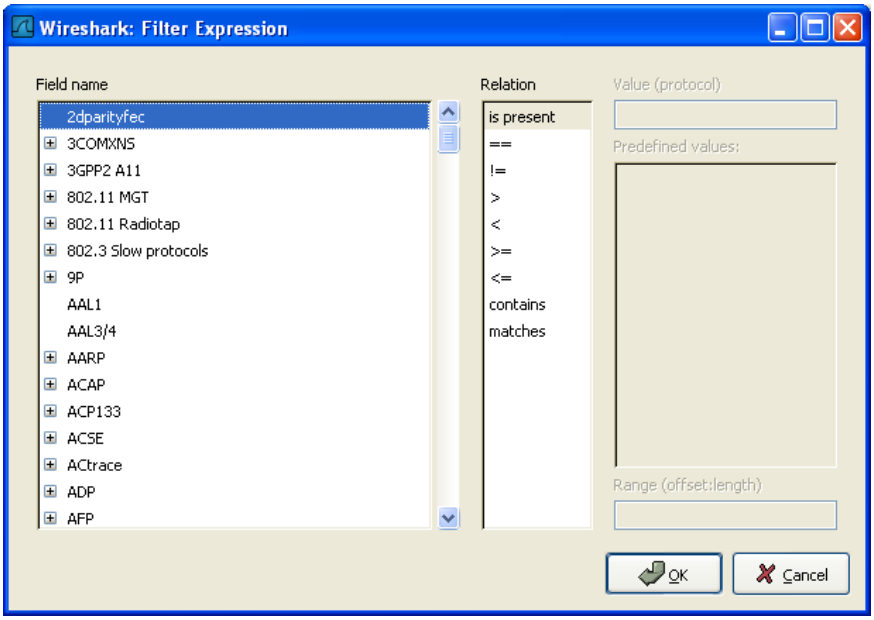
- protokol,
- postojanje podataka u paketu,
- vrijednost podatka,
- sličnost među podacima, kao i mnoge druge selekcije.

Primjerice, filter koji prikazuje promet samo SMTP (ulaz 25) i ICMP protokola može se zadati kao:

```
tcp.port eq 25 or icmp
```

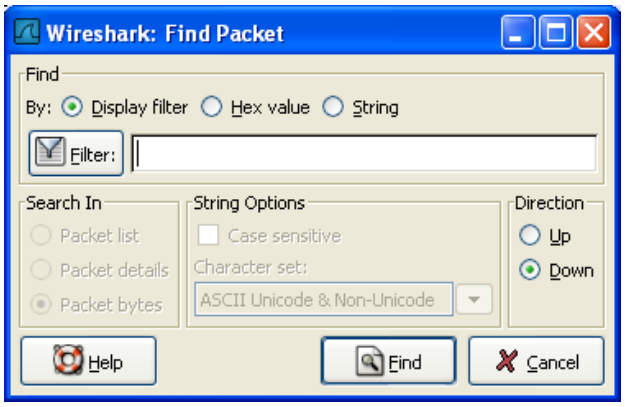
Zatim, filter koji će prikazivati samo pakete koji sadrže podatke je:

```
data
```

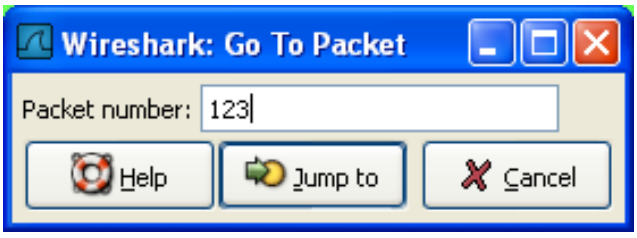



***Slika 17. „Filter Expression" prozor u Wiresharku***
***Izvor: Wireshark***

Wireshark omogućuje pronalazak uhvaćenog paketa preko opcije *„Find Packet"* (slika 17.). Na sličan način, opcija *„Go To Packet"* (slika 18.) omogućuje prikaz odabranog paketa. Osim ovih opcija moguće je označiti pakete da ih se istakne, ignorirati ih da se Wireshark ponaša kao da oni ne postoje i slično.




***Slika 18. „Find Packet" prozor u Wiresharku***
***Izvor: Wireshark***




***Slika 19. „Go To Packet" prozor u Wiresharku***
***Izvor: Wireshark***

# 6. Usporedba *Wiresharka* sa sličnim alatima

Wireshark je alat koji spada u skupinu besplatnih mrežnih analizatora. Neki od sličnih besplatnih alata za mrežnu analizu su redom:

- Capsa Free,
- Cain & Abel,
- dSniff,
- Ettercap,
- Microsoft Network Monitor,
- Ngrep,
- snoop i
- tcpdump.

***Slika 20. Logo alata dSniff***
***Izvor: monkey.org***

Izuzev Wiresharka, najpoznatiji i najrašireniji među mrežnim analizatorima su dSniff i tcpdump. Tcpdump je alat za analizu mreže koji se koristi za praćenje problema na mreži i nadgledanje aktivnosti. Nema grafičko sučelje, nego samo konzolno, pa je potrebno upisivati naredbe za korištenje. Wireshark, s druge strane, ima i grafičko sučelje te omogućuje upravljanje pomoću upisa naredbi. Tcpdump je besplatni alat licenciran BSD licencom, dok je Wireshark licenciran već spomenutom GNU GPL licencom. I Wireshark i tcpdump podržavaju većinu operacijskih sustava (Microsoft Windows, Mac OS X, Linux, Solaris itd.). Posebna inačica alata tcpdump za operacijski sustav Microsoft Windows zove se WinDump.


***Slika 21. Logo alata tcpdump***
***Izvor: tcpdump***

DSniff je alat po svojstvima vrlo sličan tcpdumpu. Informacije koje dSniff može pročitati su korisnička imena, lozinke, posjećene internetske stranice, sadržaj e-pošte i drugi. Nema grafičko sučelje, licenciran je od strane BSD Licence i primarno napravljen za Unix operacijske sustave.

Wireshark alat podržava velik broj protokola pa je tako moguće i vrlo jednostavno prenositi pakete uhvaćene od strane tcpdump-a i dSniff-a u Wireshark, kao i obrnuto.

# 7. Pregled sigurnosnih ranjivosti *Wiresharka*

U zadnje vrijeme Wireshark je u sigurnosnim člancima često spominjan u kontekstu popravljanja sigurnosnih grešaka te nadogradnje i poboljšanja sigurnosti. Sigurnosni problemi i sama ranjivost Wiresharka najviše ovisi o vrsti mreže na kojoj se koristi. Primjerice, mala SoHo (eng. *Small office / Home office*) mreža bit će manje kritična u usporedbi s web poslužiteljem neke tvrtke jer je hvatanje prometa s interne mreže vjerojatno sigurnije od hvatanja internetskog prometa itd. Sama ranjivost Wiresharka je puno manja od ostalih sličnih *open-source* programa.

Zadaci koje Wireshark obavlja, a koji su najkritičniji s gledišta sigurnosti su:

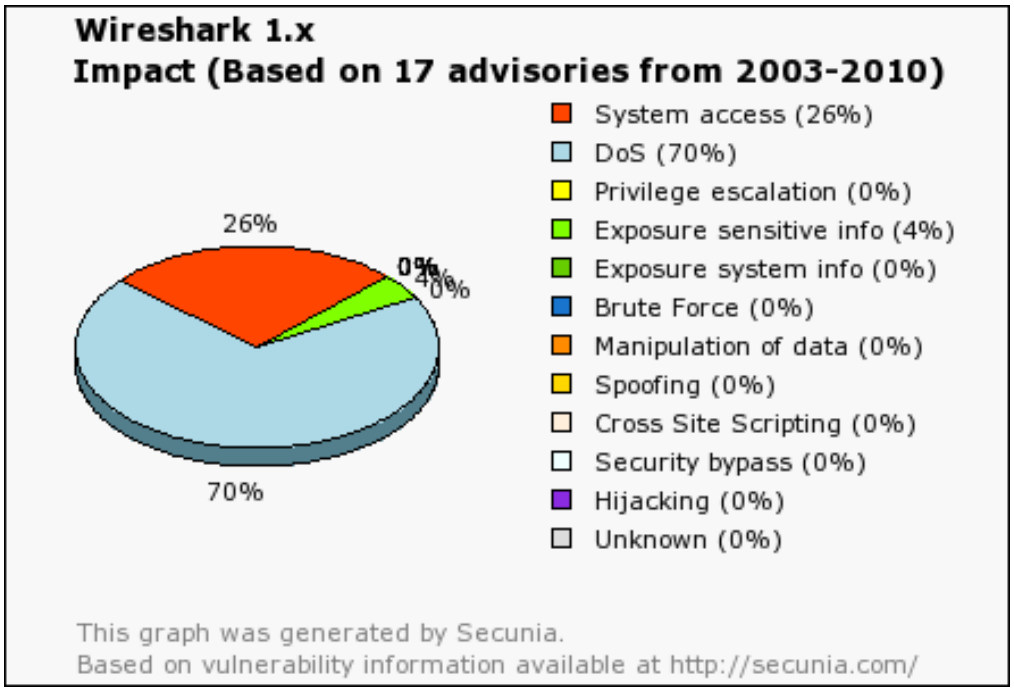
- otvaranje uhvaćenog paketa,
- korištenje opcije *„Update list of packets in real time"* za vrijeme hvatanja paketa i
- nekorištenje opcije *„Update list of packets in real time"* nakon završetka hvatanja.

Preporuke za zaštitu od sigurnosnih ranjivosti koje predlaže Wiresharkov programerski tim obuhvaćaju:

- stalno korištenje zadnje inačice Wiresharka,
- nekorištenje Wiresharka s administratorskim ovlastima te
- analiza uhvaćenih paketa u nekritičnom okruženju (npr. poseban korisnički račun).

Sigurnosne ranjivosti koje je izdvojila Secunia, tvrtka koja se bavi sigurnošću, prikazane su na grafu na slici 21. Kao glavne sigurnosne ranjivosti oni izdvajaju:

- DoS (eng. *Denial of Service*) (70%)
- neovlašteni pristup sustavu (26%) te
- izlaganje osjetljivih informacija (4%).




***Slika 22. Graf sigurnosnih ranjivosti Wireshark alata***
***Izvor: Secunia***

# 8. Zaključak

Wireshark je jedan od najboljih besplatnih alata za analizu mrežnog prometa na tržištu. Titulu jednog od najboljih zaslužio je iz više razloga. On ima, uz konzolu za upis naredbi, grafičko korisničko sučelje koje korisnicima uvelike olakšava rad i snalaženje u alatu. Wireshark podržava sve važnije mrežne protokole i ima mogućnost nadogradnje za nove protokole tako da se do sada broj podržanih protokola popeo na više od stotinu. Wireshark na vrlo jednostavan i intuitivan način omogućuje korisnicima povezivanje te uvoz i izvoz podataka na druge i s drugih sličnih mrežnih analizatora kao što su dSniff i tcpdump. Rad s paketima, kao što je hvatanje ili filtriranje paketa, u potpunosti je prilagođen grafičkom sučelju Wiresharka tako da je potrebno minimalno znanje i vrijeme za savladavanje osnovnih funkcionalnosti. Po pitanju sigurnosti Wireshark ima određenih poteškoća, ali se one svakim danom otklanjaju i svaka nova inačica Wiresharka je sve sigurnija. Dodatno, pošto je alat *open source* svaki korisnik koji uoči sigurnosni problem može ga na svojoj inačici Wiresharka i samostalno ukloniti. Planovi za budućnost Wiresharka uključuju stvaranje korisničke pomoći za svaki podržani protokol, dodatno prilagođavanje Wiresharka novim inačicama operacijskog sustava Microsoft Windows te uklanjanje grešaka u kodu, pogotovo onih koje mogu naštetiti sigurnosti korisnika.

# 9. Reference

[1] Wireshark's team: About Wireshark,
http://www.wireshark.org/about.html, kolovoz 2010.

[2] GNU General Public Licence: Wireshark,
http://www.gnu.org/licenses/gpl.html, lipanj 2007.

[3] Ethereal's team: Ethereal,
http://www.ethereal.com/, ožujak 2007.

[4] Wireshark's team: Wireshark User's Guide,
http://www.wireshark.org/docs/wsug_html_chunked/index.html, studeni 2006.

[5] Secunia's team: Vunerability Report: Wireshark 1.x
http://secunia.com/advisories/product/18083/?task=advisories, kolovoz 2010.

[6] Dug Song: dSniff
http://monkey.org/~dugsong/dsniff/, rujan 2005.

[7] Tcpdump/Libpcap: Tcpdump,
http://www.tcpdump.org/, ožujak 2009.

[8] eWEEK: The most important open-source apps of all time,
http://www.eweek.com/c/a/Linux-and-Open-Source/The-Most-Important-OpenSource-Apps-of-All-Time/5/, svibanj 2009.