TECHNOLOGIE SIECIOWE

Sprawozdanie

Lista 1

Grupa Poniedziałkowa 7:30-9:00

1. Cel

Celem tego sprawozdania jest zapoznanie z działaniem programów **ping**, **traceroute** oraz **wireshark**.

2.1. Program ping

a) Opis programu

Program wypisuje czas jaki upływa pomiędzy wysłaniem pakietu z naszego urządzenia, a odebraniem go na innym komputerze. Polecenie ping możemy użyć w celu sprawdzenia, czy dane urządzenie sieciowe jest dostępne w sieci. Polecenie ping to narzędzie, które może się przydad\ć do analizy połączenia sieciowego oraz diagnozowania problemów z funkcjonowaniem sieci. Ponadto podaje także informacje statystyczne, które mogą posłużyć do analizy połączenia.

b) Wybrane flagi wywołania programu

ping [-c liczba] [-i oczekiwanie] [-s rozmiar]

gdzie:

-i : Czas pomiędzy kolejnymi wysłaniami pakietu
-s : Rozmiar pakietu
-w : Ilość pakietów
-W : Czas oczekiwania na pakiet w sekundach
-c liczba – określa liczbę wysyłanych żądań;

c) Przykładowe wywołania:

```
ping -c 5 wp.pl
PING wp.pl (212.77.98.9) 56(84) bytes of data.
64 bytes from www.wp.pl (212.77.98.9): icmp_seq=1 ttl=55 time=22.9 ms
64 bytes from www.wp.pl (212.77.98.9): icmp_seq=2 ttl=55 time=94.1 ms
64 bytes from www.wp.pl (212.77.98.9): icmp_seq=3 ttl=55 time=26.0 ms
64 bytes from www.wp.pl (212.77.98.9): icmp_seq=4 ttl=55 time=25.9 ms
64 bytes from www.wp.pl (212.77.98.9): icmp_seq=5 ttl=55 time=69.2 ms

--- wp.pl ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
rtt min/avg/max/mdev = 22.947/47.665/94.179/28.918 ms
```

Wywołanie *ping -c 5 wp.pl* powoduje pięciokrotne wysłanie pakietu na adres www.wp.pl. Program wypisuje nam adres ip serwera, identyfikator pakietu (icmp_seq), oraz wyżej opisany czas (time).

d) Ustalanie liczby przeskoków:

W celu znalezienia Liczby przeskoków, czyli ile urządzeń jest pomiędzy użytkownikiem, a serverem docelowym, możemy użyć flagi „-t” z małą wartością (na początku może być jeden). Jeśli pojawi się komunikat „Time to live exceeded”, to znaczy, że pakiet nie doszedł, bo za niski ttl został ustawiony. Musimy użyć większej wartości. Robimy to tak długo, aż znajdziemy minimalną wartość przy fladze -t, dla której pakiet zostanie dostarczony bez błędów. Ta minimalna liczba to jest właśnie liczba przeskoków.
Poniżej przykłady ustalania liczby przeskoków:

Wywołanie pinga, gdy ustawiona wartość TTL-a jest za niska:

```
:~$ ping -c 3 -t 5 wp.pl
PING wp.pl (212.77.98.9) 56(84) bytes of data.
From rtr2.rtr-int-2.adm.wp-sa.pl (212.77.96.69) icmp_seq=1 Time to live exceeded
From rtr2.rtr-int-2.adm.wp-sa.pl (212.77.96.69) icmp_seq=2 Time to live exceeded
From rtr2.rtr-int-2.adm.wp-sa.pl (212.77.96.69) icmp_seq=3 Time to live exceeded

--- wp.pl ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2641ms
```

Wywołanie pinga, gdy wartość TTL-a jest odpowiednia:

```
~$ ping -c 3 -t 6 wp.pl
PING wp.pl (212.77.98.9) 56(84) bytes of data.
64 bytes from www.wp.pl (212.77.98.9): icmp_seq=1 ttl=55 time=33.4 ms
64 bytes from www.wp.pl (212.77.98.9): icmp_seq=2 ttl=55 time=110 ms
64 bytes from www.wp.pl (212.77.98.9): icmp_seq=3 ttl=55 time=30.3 ms

--- wp.pl ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 30.319/57.957/110.118/36.905 ms
```

Z powyższych przykładów wynika, że liczba przeskoków wynosi 6.

e) Ciekawe przypadki

- Przypadek blokowania pinga
Z uwagi na panującą cenzurę w Korei Północnej postanowiłem użyć pinga do zbadania strony o domenie północnokoreańskiej:

```
:~$ ping -c 5 cooks.org.kp
PING cooks.org.kp (175.45.176.81) 56(84) bytes of data.
From 219.158.39.42 icmp_seq=4 Packet filtered

--- cooks.org.kp ping statistics ---
5 packets transmitted, 0 received, +1 errors, 100% packet loss, time 4023ms
```

Jak widzimy powyżej pingowanie na tej stronie jest zablokowane. Nie mamy możliwości zbadania tej strony.

- Daleki Serwer
Wybrałem stronę internetową premiera państwa Papua Nowa- Gwinea w celu zbadania ile przeskoków dzieli mnie od serwerów na drugim krańcu świata.

ping -c 2 -t 18 pm.gov.pg
PING pm.gov.pg (202.95.202.9) 56(84) bytes of data.
64 bytes from proxy-rev.datec.net.pg (202.95.202.9): icmp_seq=1 ttl=42 time=662 ms
64 bytes from proxy-rev.datec.net.pg (202.95.202.9): icmp_seq=2 ttl=42 time=852 ms

--- pm.gov.pg ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1060ms
rtt min/avg/max/mdev = 662.054/757.254/852.454/95.200 ms

Dzieli nas 18 przeskoków. Czas przesyłania jest dużo większy niż w przypadku innych stron internetowych, co pozwala nam stwierdzić, że serwer jest bardzo daleko od nas.

2.2 Traceroute

a) Opis programu

Program pokazuje nam serwery z którymi nasz komputer musi się połączyć, aby dostać się do serwera docelowego. Pozwala na śledzenie trasy, którą przebiegają pakiety do hosta sieciowego. Program wykorzystuje pole TTL („time to live”) początkowo ustawiane na 1 i raportuje przejścia pakietu przez kolejne odcinki pomiędzy routerami na drodze od nadawcy do odbiorcy. Wartość TTL jest zmniejszana wraz z przechodzeniem przez kolejne routery na trasie. W momencie gdy TTL osiągnie wartość 0 dostajemy komunikat ICMP typu „Time Exceeded” informujący nas że pakiet został odrzucony przez router, co przyczynia się do uzyskania przez komputer źródłowy adresu IP pierwszego routera na trasie.

Program ten jest przeznaczony do stosowania w testowaniu, pomiarach i zarządzaniu siecią. Pozwala na ustalenie drogi pomiędzy urządzeniami. Nie zaleca się wykorzystywania traceroute w automatach (skryptach), gdyż powoduje on duże obciążenie sieci.

b) Wybrane flagi wywołania programu

traceroute [-m max_ttl] [-n] [-q nqueries] [-r] [-w waittime]

Flagi:
-m max_ttl: ustawia maksymalny TTL;
-n : adresy IP zamiast domen;
-w : czas oczekiwania na próbkę;
-q : ilość prób na każde ttl;
-r : sieć lokalna;

c) Przykładowe wywołania

```
traceroute nice-idea.org
traceroute to nice-idea.org (37.187.59.143), 30 hops max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  3.101 ms  3.326 ms  3.964 ms
 2  ip-79-110-0-1.static.tvk.wroc.pl (79.110.0.1)  16.121 ms  26.597 ms  27.288 ms
 3  do-BGP-1.tvk.wroc.pl (195.140.239.237)  24.536 ms  25.031 ms  25.785 ms
 4  * * *
 5  be100-1166.ams-1-a9.nl.eu (91.121.215.190)  54.871 ms  55.778 ms  56.681 ms
 6  be100-1102.fra-1-a9.de.eu (213.251.128.113)  52.235 ms * *
 7  * var-5-6k.pl.eu (178.33.100.158)  51.024 ms *
 8  * * *
 9  * * *
10  222.ip-37-187-44.eu (37.187.44.222)  58.831 ms  58.945 ms *
```

Analizując wywołanie t*raceroute nice-idea.org* dostajemy 10 linii wraz z adresami domen (o ile udało się je zidentyfikować), adresy IP oraz trzy wyniki czasowe, które oznaczają czas odpowiedzi danych routerów na drodze od naszego komputera do badanego urządzenia. Nasz przypadek pokazuję, że na drodze do serwera nice-idea.org znajduję się 10 urządzeń. Zauważmy, że w linii 4 oraz 8-9 zamiast informacji dostajemy gwiazdki. Oznaczają one brak odpowiedzi na zadany pakiet. Może to wynikać z celowych ustawień urządzeń lub przeciążenia.

2.3 Program Wireshark

a) Opis programu

Program Wireshark jest snifferem, który analizuje ruch sieciowy. Umożliwia przechwytywanie i nagrywanie pakietów danych, a także ich dekodowanie, przez określone interfejsy sieciowe. Dzięki dużej ilości pluginów potrafi rozpoznać i zdekodować wiele protokołów komunikacyjnych. Posiada graficzny interfejs. Dzięki filtrom można zawęzić znacząco wyniki poszukiwań wg ustalonych kryteriów. Będąc połączonym z siecią publicznym np. na Politechnice, jesteśmy w stanie zobaczyć, co inni użytkownicy sieci publicznej oglądają na internecie. Używając Wiresharka, udało mi się zobaczyć, że jeden z użytkowników publicznej sieci wchodzi na stronę „polwro.pl” i wyszukuje opinii o prowadzących. Z tego powodu ten program jest często wykorzystywany przez hackerów, służby specjalne, ale także administratorów do rozwiązywania problemów z siecią.

b) Przykładowe wywołanie

Frame 37: 318 bytes on wire (2544 bits), 318 bytes captured (2544 bits) on interface 0
Interface id: 0 (wlp7s0)
Encapsulation type: Ethernet (1)
Arrival Time: Mar 20, 2017 13:27:10.752955717 CET
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1490012830.752955717 seconds
[Time delta from previous captured frame: 0.032511023 seconds]
[Time delta from previous displayed frame: 0.152458901 seconds]
[Time since reference or first frame: 0.181863801 seconds]
Frame Number: 37
Frame Length: 318 bytes (2544 bits)
Capture Length: 318 bytes (2544 bits)
[Frame is marked: False]
[Frame is ignored: False]
[Protocols in frame: eth:ethertype:ip:tcp:http:data:data-text-lines]
[Coloring Rule Name: HTTP]
[Coloring Rule String: http || tcp.port == 80 || http2]
Ethernet II, Src: Fortinet_09:00:02 (00:09:0f:09:00:02), Dst: HonHaiPr_c7:23:6d (90:48:9a:c7:23:6d)
Destination: HonHaiPr_c7:23:6d (90:48:9a:c7:23:6d)
Source: Fortinet_09:00:02 (00:09:0f:09:00:02)
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 91.121.184.99, Dst: 172.16.86.81
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 304
Identification: 0xe89c (59548)
Flags: 0x02 (Don't Fragment)
Fragment offset: 0
Time to live: 50
Protocol: TCP (6)
Header checksum: 0x48ed [validation disabled]
Source: 91.121.184.99
Destination: 172.16.86.81
[Source GeoIP: France, AS16276 OVH SAS, 48.860001, 2.350000]
[Destination GeoIP: Unknown]
Transmission Control Protocol, Src Port: 80 (80), Dst Port: 64949 (64949), Seq: 1, Ack: 666, Len: 264
Source Port: 80
Destination Port: 64949
[Stream index: 1]
[TCP Segment Len: 264]
Sequence number: 1 (relative sequence number)
[Next sequence number: 265 (relative sequence number)]
Acknowledgment number: 666 (relative ack number)
Header Length: 20 bytes
Flags: 0x018 (PSH, ACK)
Window size value: 330
[Calculated window size: 330]
[Window size scaling factor: -1 (unknown)]
Checksum: 0xb66d [validation disabled]
Urgent pointer: 0
[SEQ/ACK analysis]
Hypertext Transfer Protocol
HTTP/1.1 200 OK\r\n
Server: nginx/1.2.1\r\n
Date: Mon, 20 Mar 2017 12:30:39 GMT\r\n
Content-Type: text/html; charset=latin2\r\n
Transfer-Encoding: chunked\r\n
Connection: keep-alive\r\n
X-Powered-By: PHP/5.4.45-0+deb7u2\r\n
Content-Encoding: gzip\r\n
\r\n
[HTTP response 1/11]
[Time since request: 0.152458901 seconds]
[Request in frame: 6]
[Next request in frame: 3563]
[Next response in frame: 3579]

HTTP chunked response
Content-encoded entity body (gzip): 24 bytes -> 4 bytes

Powyższy przykład pokazuje jak wireshark przechwytuje dane. Analizowana tutaj 37 bramka zawiera 318 bajtów danych. Każda z ramek zawiera informacje na temat adresu routera, źródła, adresy IP i MAC urządzeń biorących udział w transferze, porty, flagi, sumy kontrolne itp.

3. Podsumowanie

Reasumując, trzy powyższe programy są bardzo ciekawymi programami do badania sieci. Dzięki zastosowaniu odpowiednich parametrów (w przypadku pinga i tracerouta) czy filtrów (gdy mamy do czynienia z Wiresharkiem), mamy naprawdę wielkie polę możliwości do badania sieci i do zdobywania ogromnej ilości informacji. Dzięki temu sprawozdaniu nauczyłem się wyznaczać liczbę przeskoków z mojego komputera do najdalszych serwerów. Mimo, że może być to trudniejsze niż w przypadku programu Traceroute, który pokazuje nam wszystkie urządzenia po drodze, to jednak ping jest prostym narzędziem o szerokim spectrum zastosowań. Natomiast Wireshark pokazuje, jak niebezpieczne potrafi być surfowanie po publicznych sieciach. Zwróciło to moją szczególną uwagę na bezpieczeństwo w sieci. Zadania wykonane w sprawozdaniu sporo mnie nauczyły o sieciach.