Network Compression Detection

Generated by Doxygen 1.8.6

Sun Dec 7 2014 00:40:19

Contents

Chapter 1

NCD

Network Compression Detection

A Project for Comp 429 @ CSUN, to detect compression along a network transmission path.

The project will proceed in phases, and incrementally build to be an application suitable to an uncoperative environment.

We will use low and high entropy data sent along the network to detect if compression occurs. If a node compresses the data, the low entropy transmission times will differ significantly from the high entropy data transmission times, and give us a reliable way to detect compression.

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Data Structure Documentation

4.1 pseudo_header Struct Reference

struct for udp pseudo header

```
#include <ncd.h>
```

Data Fields

- u_int32_t source
- u_int32_t dest
- u_int8_t zero
- u_int8_t proto
- uint16_t len

4.1.1 Detailed Description

struct for udp pseudo header

Definition at line 32 of file ncd.h.

4.1.2 Field Documentation

4.1.2.1 u_int32_t pseudo_header::dest

Definition at line 34 of file ncd.h.

4.1.2.2 uint16_t pseudo_header::len

Definition at line 37 of file ncd.h.

4.1.2.3 u_int8_t pseudo_header::proto

Definition at line 36 of file ncd.h.

4.1.2.4 u_int32_t pseudo_header::source

Definition at line 33 of file ncd.h.

4.1.2.5 u_int8_t pseudo_header::zero

Definition at line 35 of file ncd.h.

The documentation for this struct was generated from the following files:

- ncd.h
- tracert.c

Chapter 5

File Documentation

5.1 client.c File Reference

```
#include <ctype.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netinet/udp.h>
#include <stdio.h>
#include <stdlib.h>
#include <arpa/inet.h>
#include <string.h>
#include <fcntl.h>
#include "icmp.h"
```

Functions

- int main (int argc, char ∗argv[])

  Detects compression by sending a stream of low entropy and high entropy packets, and comparing their transmission time to determine if compression occurs along the transmission path.

5.1.1 Detailed Description

Author

: Paul Kirth

Comp 429 Project Phase I

Definition in file client.c.

5.1.2 Function Documentation

5.1.2.1 int main (int *argc,* char ∗ *argv[]*)

Detects compression by sending a stream of low entropy and high entropy packets, and comparing their transmission time to determine if compression occurs along the transmission path.

[1] Destination IP address [2] Number of data packets to send in each train [3] TIME_WAIT

Definition at line 29 of file client.c.

5.2 ncd.c File Reference

```
#include "ncd.h"
```

Functions

- double get_time (void)
- int comp_det (char ∗address, char ∗port, char hl, size_t data_size, size_t num_packets, ushort ttl, size_t time_wait, int n_tail)

    Determines if compression occurs along the current transmission path to host by sending data to a remote location.
- int send_data (char ∗address, char ∗port_name, char hl, size_t data_size, size_t num_packets, ushort ttl, size_t time_wait, int n_tail)

    sends data train to the end host with leading and trailing icmp timestamps
- int recv_data (double ∗time)

    Receives ICMP responses from end host and records times.
- uint16_t ip_checksum (void ∗vdata, size_t length)

    calculates the ip cheksum for some buffer of size length
- int main (int argc, char ∗argv[ ])

    Main function only calls comp_detection()

5.2.1 Function Documentation

5.2.1.1 int comp_det (char ∗ *address,* char ∗ *port,* char *hl,* size_t *data_size,* size_t *num_packets,* ushort *ttl,* size_t *time_wait,* int *n_tail*)

Determines if compression occurs along the current transmission path to host by sending data to a remote location.

two data trains are sent each with leading and trailing ICMP timestamp messages

the first data train will be low entropy data, to encourage compression the second train will have high entropy data, which should not be compressed if the times are significantly different, we have reasonable evidence compression exists along this path.

Parameters

address	the address of the end host stored in a char array (cstring)
port	the port number or service name
num_packets	the number of packets in each data train
time_wait	the wait between trains

Returns

0 success, 1 error/failure

Definition at line 12 of file ncd.c.

5.2.1.2 double get_time (void)

Definition at line 3 of file ncd.c.

5.2.1.3 uint16_t ip_checksum (void ∗ *vdata,* size_t *length*)

calculates the ip cheksum for some buffer of size length

Parameters

vdata	a pointer to the buffer to be checksummed
length	the length in bytes of the data to be checksummed

Returns

the IP checksum of the buffer

Definition at line 323 of file ncd.c.

5.2.1.4 int main (int *argc,* char ∗ *argv[]*)

Main function only calls comp_detection()

[1] Destination IP address [2] Port Number [3] High or low entropy data 'H' or 'L' [4] Size of udp data packet [5] Number of packets in UDP Data Train [6] Time to Live [7] Wait time in milliseconds [8] number of tail ICMP messages

Definition at line 391 of file ncd.c.

5.2.1.5 int recv_data (double ∗ *time*)

Receives ICMP responses from end host and records times.

Parameters

time	returns the time in ms between head echo response and first processed tail echo response to a resolution of microseconds (10^-6 sec)

Returns

0 success, -1 error/failure

Definition at line 244 of file ncd.c.

5.2.1.6 int send_data (char ∗ *address,* char ∗ *port,* char *hl,* size_t *data_size,* size_t *num_packets,* ushort *ttl,* size_t *time_wait,* int *n_tail*)

sends data train to the end host with leading and trailing icmp timestamps

sets up a connection using raw sockets and sends a head ICMP echo request followed by a UDP data train. After the data train is sent, a series of ICMP echo responses is sent.

Parameters

address	the address of the end host stored in a char array (cstring)
port	the port number or service name
hl	the entropy of the data (either 'H' for high entropy or 'L' for low entropy
data_size	the size in bytes of the data to be sent in the udp train
num_packets	the number of packets in each data train
ttl	the time to live for each packet (max size 255)
time_wait	the wait between ICMP tail messages
n_tail	the number of ICMP tail messages to be sent

Returns

0 success, -1 error/failure

Definition at line 45 of file ncd.c.

5.3 ncd.h File Reference

```
#include <ctype.h>
#include <sys/types.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <time.h>
#include <errno.h>
#include <sys/socket.h>
#include <netinet/udp.h>
#include <netinet/in.h>
#include <netinet/ip.h>
#include <netinet/ip_icmp.h>
#include <netdb.h>
#include <arpa/inet.h>
#include <signal.h>
#include <fcntl.h>
```

Data Structures

- struct pseudo_header

  struct for udp pseudo header

Macros

- #define SIZE 1500

  maximum ip packet size

Functions

- int comp_det (char ∗address, char ∗port, char hl, size_t data_size, size_t num_packets, ushort ttl, size_t time_wait, int n_tail)

  Determines if compression occurs along the current transmission path to host by sending data to a remote location.
- int send_data (char ∗address, char ∗port, char hl, size_t data_size, size_t num_packets, ushort ttl, size_t time_wait, int n_tail)

  sends data train to the end host with leading and trailing icmp timestamps
- int recv_data (double ∗time)

  Receives ICMP responses from end host and records times.
- uint16_t ip_checksum (void ∗vdata, size_t length)

  calculates the ip cheksum for some buffer of size length

5.3.1 Macro Definition Documentation

5.3.1.1 #define SIZE 1500

maximum ip packet size

Definition at line 26 of file ncd.h.

5.3.2 Function Documentation

5.3.2.1 int comp_det (char ∗ *address,* char ∗ *port,* char *hl,* size_t *data_size,* size_t *num_packets,* ushort *ttl,* size_t *time_wait,* int *n_tail*)

Determines if compression occurs along the current transmission path to host by sending data to a remote location.

two data trains are sent each with leading and trailing ICMP timestamp messages

the first data train will be low entropy data, to encourage compression the second train will have high entropy data, which should not be compressed if the times are significantly different, we have reasonable evidence compression exists along this path.

Parameters

address	the address of the end host stored in a char array (cstring)
port	the port number or service name
num_packets	the number of packets in each data train
time_wait	the wait between trains

Returns

0 success, 1 error/failure

Definition at line 12 of file ncd.c.

5.3.2.2 uint16_t ip_checksum (void ∗ *vdata,* size_t *length*)

calculates the ip cheksum for some buffer of size length

Parameters

vdata	a pointer to the buffer to be checksummed
length	the length in bytes of the data to be checksummed

Returns

the IP checksum of the buffer

Definition at line 323 of file ncd.c.

5.3.2.3 int recv_data (double ∗ *time*)

Receives ICMP responses from end host and records times.

Parameters

time	returns the time in ms between head echo response and first processed tail echo response to a resolution of microseconds (10^-6 sec)

Returns

0 success, -1 error/failure

Definition at line 244 of file ncd.c.

5.3.2.4 int send_data (char ∗ *address,* char ∗ *port,* char *hl,* size_t *data_size,* size_t *num_packets,* ushort *ttl,* size_t *time_wait,* int *n_tail*)

sends data train to the end host with leading and trailing icmp timestamps sets up a connection using raw sockets and sends a head ICMP echo request followed by a UDP data train. After the data train is sent, a series of ICMP echo responses is sent.

Parameters

address	the address of the end host stored in a char array (cstring)
port	the port number or service name
hl	the entropy of the data (either 'H' for high entropy or 'L' for low entropy
data_size	the size in bytes of the data to be sent in the udp train
num_packets	the number of packets in each data train
ttl	the time to live for each packet (max size 255)
time_wait	the wait between ICMP tail messages
n_tail	the number of ICMP tail messages to be sent

Returns

> 0 success, -1 error/failure

Definition at line 45 of file ncd.c.

5.4 ping.c File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include <unistd.h>
#include <sys/time.h>
#include <string.h>
#include <netinet/ip.h>
#include <netinet/ip_icmp.h>
#include <netdb.h>
```

Macros

- #define SIZE 1024

Functions

- double get_time (void)
- double sub_time (struct timeval ∗out, struct timeval ∗in)
- uint16_t ip_checksum (void ∗vdata, size_t length)

  calculates the ip cheksum for some buffer of size length
- void debug_print (struct icmp ∗icmp, ssize_t n)
- int main (int arc, char ∗argv[ ])

Variables

- size_t nsent

5.4.1 Macro Definition Documentation

5.4.1.1 #define SIZE 1024

Definition at line 96 of file ping.c.

5.4.2 Function Documentation

5.4.2.1 void debug_print (struct icmp ∗ *icmp*, ssize_t *n*)

Definition at line 199 of file ping.c.

5.4.2.2 double get_time (void)

Definition at line 19 of file ping.c.

5.4.2.3 uint16_t ip_checksum (void ∗ *vdata*, size_t *length*)

calculates the ip cheksum for some buffer of size length

Parameters

vdata	a pointer to the buffer to be checksummed
length	the length in bytes of the data to be checksummed

Returns

the IP checksum of the buffer

Definition at line 38 of file ping.c.

5.4.2.4 int main (int *arc*, char ∗ *argv[]*)

Definition at line 98 of file ping.c.

5.4.2.5 double sub_time (struct timeval ∗ *out*, struct timeval ∗ *in*)

Definition at line 28 of file ping.c.

5.4.3 Variable Documentation

5.4.3.1 size_t nsent

Definition at line 92 of file ping.c.

5.5 ping_ip.c File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include <unistd.h>
#include <sys/time.h>
#include <string.h>
#include <netinet/ip.h>
#include <netinet/ip_icmp.h>
#include <netdb.h>
```

Macros

- #define SIZE 1024

Functions

- double get_time (void)
- double sub_time (struct timeval ∗out, struct timeval ∗in)
- uint16_t ip_checksum (void ∗vdata, size_t length)

  calculates the ip cheksum for some buffer of size length
- int main (int arc, char ∗argv[ ])
- void debug_print (struct icmp ∗icmp, ssize_t n)

Variables

- size_t nsent

5.5.1 Macro Definition Documentation

5.5.1.1 #define SIZE 1024

Definition at line 98 of file ping_ip.c.

5.5.2 Function Documentation

5.5.2.1 void debug_print (struct icmp ∗ *icmp,* ssize_t *n*)

Definition at line 251 of file ping_ip.c.

5.5.2.2 double get_time (void)

Definition at line 19 of file ping_ip.c.

5.5.2.3 uint16_t ip_checksum (void ∗ *vdata,* size_t *length*)

calculates the ip cheksum for some buffer of size length

Parameters

vdata	a pointer to the buffer to be checksummed
length	the length in bytes of the data to be checksummed

Returns

the IP checksum of the buffer

Definition at line 38 of file ping_ip.c.

5.5.2.4 int main (int *arc,* char ∗ *argv[ ]*)

Definition at line 100 of file ping_ip.c.

5.5.2.5 double sub_time (struct timeval ∗ *out,* struct timeval ∗ *in*)

Definition at line 28 of file ping_ip.c.

5.5.3 Variable Documentation

5.5.3.1 size_t nsent

Definition at line 92 of file ping_ip.c.

5.6 README.md File Reference

5.7 server.c File Reference

```
#include <ctype.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netinet/udp.h>
#include <stdio.h>
#include <stdlib.h>
#include <arpa/inet.h>
#include <string.h>
#include <time.h>
```

Functions

- clock_t procs_msg (size_t num_msg, int sockfd, struct sockaddr_in client)
- int main (int argc, char ∗argv[])

  sets up a server whos clients will send a series of data packages so that compression along the transmission path can be detected.

5.7.1 Detailed Description

Author

: Paul Kirth

Comp 429 Project Phase I

Definition in file server.c.

5.7.2 Function Documentation

5.7.2.1 int main (int *argc,* char ∗ *argv[ ]*)

sets up a server whos clients will send a series of data packages so that compression along the transmission path can be detected.

Definition at line 32 of file server.c.

5.7.2.2 clock_t procs_msg (size_t *num_msg,* int *sockfd,* struct sockaddr_in *client*)

Parameters

num_msg	the number of messages to expect
sockfd	the socket file descriptor to expect communication from
client	a data structure to hold client information

Returns

the time spent gathering packets in the train

Definition at line 140 of file server.c.

5.8 test.c File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include <unistd.h>
#include <sys/time.h>
#include <string.h>
#include <netinet/ip.h>
#include <netinet/ip_icmp.h>
#include <netdb.h>
```

Macros

- #define SIZE 1024

Functions

- double get_time (void)
- double sub_time (struct timeval ∗out, struct timeval ∗in)
- uint16_t ip_checksum (void ∗vdata, size_t length)

  calculates the ip cheksum for some buffer of size length
- void debug_print (struct icmp ∗icmp, ssize_t n)
- int main (int arc, char ∗argv[])

Variables

- size_t nsent

5.8.1 Macro Definition Documentation

5.8.1.1 #define SIZE 1024

Definition at line 96 of file test.c.

5.8.2 Function Documentation

5.8.2.1 void debug_print (struct icmp ∗ *icmp,* ssize_t *n*)

Definition at line 192 of file test.c.

5.8.2.2 double get_time (void)

Definition at line 19 of file test.c.

5.8.2.3 uint16_t ip_checksum (void ∗ *vdata,* size_t *length*)

calculates the ip cheksum for some buffer of size length

Parameters

vdata	a pointer to the buffer to be checksummed
length	the length in bytes of the data to be checksummed

Returns

the IP checksum of the buffer

Definition at line 38 of file test.c.

5.8.2.4 int main (int *arc,* char ∗ *argv[ ]*)

Definition at line 98 of file test.c.

5.8.2.5 double sub_time (struct timeval ∗ *out,* struct timeval ∗ *in*)

Definition at line 28 of file test.c.

5.8.3 Variable Documentation

5.8.3.1 size_t nsent

Definition at line 92 of file test.c.

5.9 tracert.c File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include <unistd.h>
#include <sys/time.h>
#include <string.h>
#include <netinet/ip.h>
#include <netinet/ip_icmp.h>
#include <netinet/udp.h>
#include <netdb.h>
#include <fcntl.h>
```

Data Structures

- struct pseudo_header

  struct for udp pseudo header

Macros

- #define SIZE 1500

Functions

- double get_time (void)
- double sub_time (struct timeval ∗out, struct timeval ∗in)
- uint16_t ip_checksum (void ∗vdata, size_t length)

  calculates the ip cheksum for some buffer of size length
- void debug_print (struct icmp ∗icmp, ssize_t n)
- int main (int arc, char ∗argv[ ])

Variables

- size_t nsent

5.9.1 Macro Definition Documentation

5.9.1.1 #define SIZE 1500

Definition at line 95 of file tracert.c.

5.9.2 Function Documentation

5.9.2.1 void debug_print (struct icmp ∗ *icmp,* ssize_t *n*)

Definition at line 355 of file tracert.c.

5.9.2.2 double get_time (void)

Definition at line 18 of file tracert.c.

5.9.2.3 uint16_t ip_checksum (void ∗ *vdata,* size_t *length*)

calculates the ip cheksum for some buffer of size length

Parameters

vdata	a pointer to the buffer to be checksummed
length	the length in bytes of the data to be checksummed

Returns

the IP checksum of the buffer

Definition at line 37 of file tracert.c.

5.9.2.4 int main (int *arc,* char ∗ *argv[ ]*)

Definition at line 106 of file tracert.c.

5.9.2.5 double sub_time (struct timeval ∗ *out,* struct timeval ∗ *in*)

Definition at line 27 of file tracert.c.

5.9.3 Variable Documentation

5.9.3.1 size_t nsent

Definition at line 91 of file tracert.c.

Index