HTTP 1.1 Server

CS425A
Computer Networks

Implemented Features :-

1) Implemented the GET method
2) Handled all the extension given in the project
3) Send data back to client for all the extensions given in the project
4) Implemented persistent connection
5) Implemented handling of multiple clients
6) Implemented all the error types 200, 400, 404, 500, 501
7) Server Port Initialised at start up.
8) Implemented the base directory.
9) Include the date and server field in the response header.
10) Made a log of all the requests that the client makes.

Running The code

$ make
$./http-server <port> <base directory>
$make clean

Sai Kishan Pampana
13458
pkishan@iitk.ac.in

Optional Features

- ## Allow the server port to be initialised at start up

    Using the **atoi** function, we can store a given number in string to that of an integer. So, in the terminal we can give an argument for the port number and then convert that argument from string to integer using the **atoi** function call, and then we give this argument in the address of the server.

- ## Allow the document base directory to be initialised at start up

    This is again similar to how we implemented the server port on initialisation, so here what we do is to give the base directory at the initialisation and then take this string and append to it the requested file name to get the path of the file that we need.

- ## Include the Date and Server fields in the Response message header

    For including Date fields in the response header, we can use the function strftime, this function gives the data in the format that we want it. For the server field, we first name our server and store that in a string. Once we have both the date and server string we just need to append them to the response message and send them, this is done in the function sendhead.

- ## Log all client requests to a file

    For logging all the request of the file we create a new file and a new function. The new file has all the log requests in it and the new function is for writing the requests in the log file. To keep appending the logs to the log file all we need to do is open the file in the mode "a". Once we opened it in mode "a" we just need to keep writing as the requests come.

IMPLEMENTATION OF THE PROJECT

Browser Used:

The browser that was used for this project is Google Chrome Version 52.0.2743.116 (64-bit).
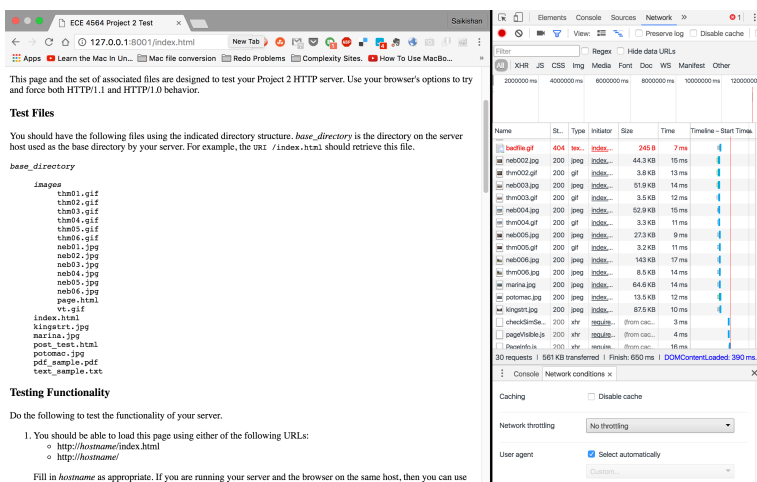
Test Procedure Used:

For the testing of this project the procedure that was used was to start the server on the terminal and then call for files using the client Google Chrome.

For checking the persistent connection I have made multiple request on the same client and checked if the connection is closed or if it is still in operation.

To check if the requests are send or not we can verify this by going to "Inspect Element" and then going to "Networks". Here if we preserve logs then we can look at all the request made and the kind of response that the server sends them.

The image provided is an example where we run the index.html file using the server, as you can see except the badimage.jpg the remaining images have been successfully transferred and have a status code of 200 but the bad image has a status code of 404.
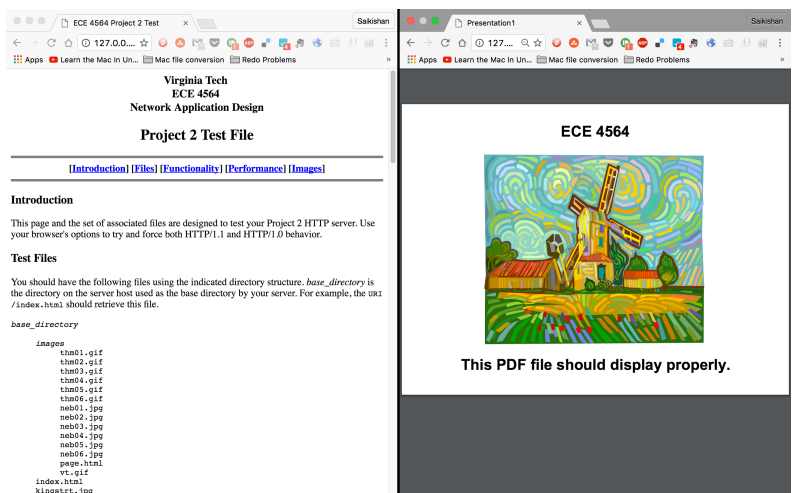
So using this above images we can show that the file index.html is working i.e we are able to get all the information through the server.

To check if the server is able to server multiple clients or not we can verify by opening two tabs and checking if both the request in both the tabs are being served or not.
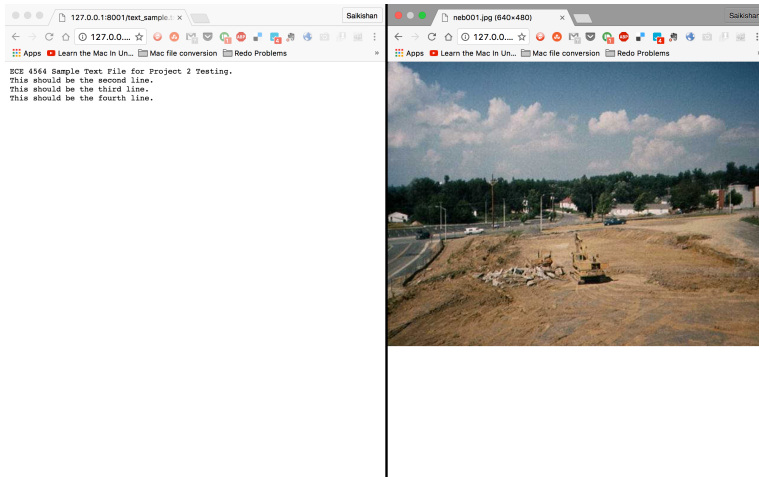
This is because since we have persistent connection if we are able to server requests on both the tabs then we can show that the server is serving multiple clients

Using this image we have also shown that the test case of .pdf is also working
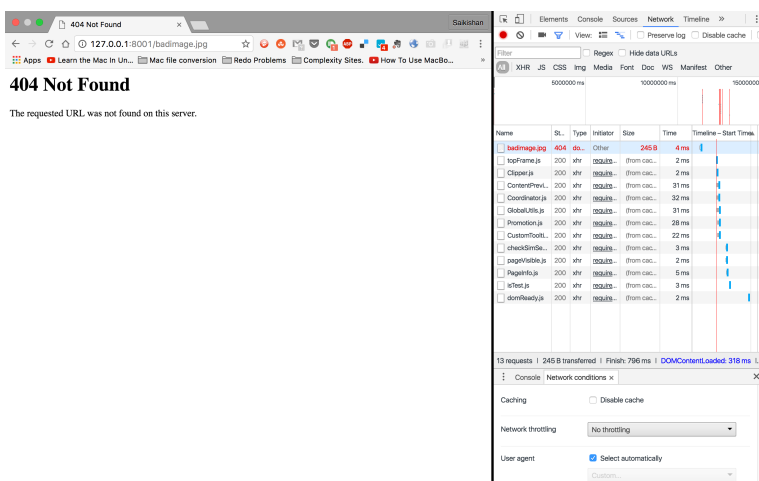

So using the above images we can show that the server is able to server multiple clients.

Now that we have show test results for the case where we are serving multiple clients and also the images showing how we are testing the working or not working of the connection let us now look at all the different test cases that were provided to us. We have already showed that the index.html file and the .pdf file are working, so now let us show test results for the remaining test cases.

This images is the result is shows that both the .txt file and the .jpg files have been successfully received from the server

We can see the names of the files on the tabs to verify that they have actually been received form the server.

Here in this image we can see that given a file name that is not present in the directory of the server we have returned an error stating that we have not found the file on the server.

Also we can verify this by looking at the networks option in the inspect element, where we can see that the request for the badimage.jpg has failed and is highlighted in red.

SERVER CODE :-

```c
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <pthread.h>
#include <assert.h>
#include <time.h>

//Defining all the function that we will be using in the code

void getval(char* data, char* buf);
void sendhead(int client_socket, int flag, char* Type,int filesize,char *contype);
void datawrite(int client_socket, FILE *fp1, int filesize);
void logfile(FILE *lgfp, char *uri, char *ip);

int main(int argc, char **argv)              //Giving argument in the terminal for the port and
the base directory
{
   int server_socket, client_socket;

   socklen_t addrlen;

   char buf[1024],head[1024];
   struct sockaddr_in address;

   if(argc != 3)                      //Checking if both the port number and the base
directory are given or not
   {
     printf("Give both the argument of port number and base directory");
     exit(1);
   }

   if ((server_socket = socket(AF_INET, SOCK_STREAM, 0)) < 0)     //Creating the server
socket which will listen for client requests
   {
     printf("The socket was not created\n");
     exit(1);
   }
```

```c
    printf("Socket Established \n");

    address.sin_family = AF_INET;
    address.sin_addr.s_addr = INADDR_ANY;
    address.sin_port = htons(atoi(argv[1]));   //Setting up the port on which the server will be
listening

    if (bind(server_socket, (struct sockaddr *) &address, sizeof(address)) == 0)     //Binding
the server sockets to the adress that was defined above
    {
        printf("Binding Socket\n");
    }

    if (listen(server_socket, 10) < 0)        //Make the server to start listening to the requests
of the clients
    {
        perror("server: listen");
        exit(1);
    }



    while (1)
    {
        if ((client_socket = accept(server_socket, (struct sockaddr *) &address, &addrlen)) < 0)
//Creating a new socket on which client requests are servered
        {
            perror("Could not create new socket \n");
            exit(1);
        }

        else
        {
            pid_t pid;
            pid = fork();                      //Using fork so that we create child process so that we
can server multiple clients at the same time

            if(pid !=0)                        //This if condition is to make the parent process go to
the start of the loop and wait for other clients
            {
                close(client_socket);
                continue;
            }

            FILE *lgfp = fopen("log.txt","a");        //This is a log file in which all the requests made
to this server are noted down

            while(1)                          //This is the infinite while loop which the child uses to
keep a persistent connection with client
```

```c
        {
            if(recv(client_socket, buf, 1024, 0) < 0)     //Receiving the request from the client
and storing that as a string in buf
            {
              printf("No message is received from the client \n");
              fclose(lgfp);
              exit(1);
            }

            struct stat st;                    //This struct will later be used to find the size of the file
which we have to send to client

            char *data1;                       //Defining fields in which data from the client request
will be stored
            char *data2;
            char *data3;
            char check[4];

            int filesize;                      //This stores the file size within it

            char *basedir;                     //In this we store a string corresponding to the base
directory

            basedir = (char*)malloc(20*sizeof(char));
            bzero((char *)basedir, sizeof(basedir));
            basedir = argv[2];                 //Storing the base directory that was given as one of
the arguments in the terminals

            strncpy(check, buf, 3);            //Copy the first three characters of the string buf
            check[3] = '\0';


            if(strcmp(check, "GET") !=0)       //Checking if the method that was used is the
GET method or not
            {
              sendhead(client_socket, 4, " ", 0 , "Keep-Alive");    //If not get method then we
send a 501 error
              continue;
            }

            data1 = (char *)malloc(1024*sizeof(char));
            getval(data1, buf);                        //Copying the data from buf into data1 string

            data1 = data1 + 5;                 //Changing the pointer of data1 to get the name
of the file

            char *Type;                        //This is used to get the type of file that the client
has requested
            Type = (char *)malloc(20*sizeof(char));
            bzero((char *)Type, sizeof(Type));
```

```
        Type = strstr(data1, ".");

        char *ip;                           //This string is used to store the client's host ip
address
        ip = (char*)malloc(1024*sizeof(char));
        bzero((char *)ip, sizeof(ip));

        sprintf(ip, "%d.%d.%d.%d", (int)(address.sin_addr.s_addr&0xFF), (int)
((address.sin_addr.s_addr&0xFF00)>>8),(int)
((address.sin_addr.s_addr&0xFF0000)>>16),(int)
((address.sin_addr.s_addr&0xFF000000)>>24));        //This call is used to store the client's
host ip in the string ip

        logfile(lgfp, data1,ip);                //This function is called to log the request that
was made by the client

        if(strcmp(Type, ".html") == 0 || strcmp(Type, "htm")==0)    //This if conditional is to
get Content-Type in the response header
        {
            Type = "text/html";                //If file is .html then response Content-Type is
text/html
        }

        if(strcmp(Type, ".txt") == 0)
        {
            Type = "text/plain";               //If file is .txt then response Content-Type is text/
plain
        }

        if(strcmp(Type, ".jpeg") == 0 || strcmp(Type, ".jpg")==0)
        {
            Type = "image/jpeg";                   //If file is .jpg or .jpeg then response Content-
Type is image/jpeg
        }

        if(strcmp(Type, ".gif") == 0)
        {
            Type = "image/gif";                //If file is .gif then response Content-Type is
image/gif
        }

        if(strcmp(Type, ".pdf") == 0)
        {
            Type = "Application/pdf";               //If file is .pdf then response Content-Type is
Application/pdf
        }
```

```c
        data2 = (char *)malloc(1024*sizeof(char));  //Data 2 is used to find the pointer
where the connection type is mentioned in the clinet request
        data2 = strstr(buf, "Connection:");

        data2 = data2 + 12;

        data3 = (char *)malloc(1024*sizeof(char));  //data 3 store ths value of the
connection type
        strcpy(data3, data2);

        int k = 0;

        while(1)                          //Loop to store the value of the connection type in the
data 3 field
        {
          if(data3[k] == '\r')
          {
            data3[k] = '\0';
            break;
          }
          k = k + 1;
        }
        char *base;                       //The base variable is used to give the complete
path in which the file is present
        base = (char *)malloc(20*sizeof(char));
        bzero((char *)base, sizeof(base));
        strcpy(base, basedir);

        if(strcmp(base, "/") ==0)          //If the base directory given is the current
directory the we replace that with ""
          bzero((char *)base, sizeof(base));

        strcat(base, data1);               //Concatenate the base directory along with the
file name to get the complete path
        FILE *fp = fopen(base, "rb");       //Opennig the file that was requested

        if(fp == NULL)                     //Checking if the file exists in the directory or not
        {
          sendhead(client_socket, 1, Type,0,data3);
          continue;
        }

        stat(base, &st);                   //This is used to file the size of the file that we have
to read
        filesize = st.st_size;             //Filesize is a variable which has the size of the file
stored in it

        sendhead(client_socket, 0, Type,filesize, data3);   //This function is used to send a
response header to the client
```

```c
        datawrite(client_socket, fp, filesize);          //This function is used to write the
chunks of data to client after sending the response header

        fclose(fp);                          //Closing the file after all the readin and the writing
part is done

    }
  }
    close(client_socket);
  }
  close(server_socket);
  return 0;
}


void getval(char* data, char* buf)          //This function is used to get the name of the file
from the request sent by client
{
    strcpy(data, buf);

    int i=0;

    while(i< strlen(buf))              //This while loop is to set the data string to the name of
the file
    {
      if(data[i]=='\r')
      {
        data[i]='\0';
        break;
      }
      i = i + 1;
    }

    int a = strlen(data)-9;          //This is done to remove the version from the http
request sent by the clients

    data[a] = '\0';

}


void datawrite(int client_socket, FILE *fp1, int filesize)          //This function is used to send
data in chunks to the client
{
  char writeBuff[1023];

  int left = filesize%1023;

  if(filesize%1023 ==0)
  {
```

```c
      filesize = filesize/1023;
    }

  else
  {
    filesize = filesize/1023 +1 ;
  }

  int i =0;

  while(i<filesize)                              //In each iteration we send a buffer of size
1023 bytes to the client
{
    bzero((char *)writeBuff, sizeof(writeBuff));
    int b = fread(writeBuff,1,1023,fp1);
    fflush(fp1);
    printf("Bytes read %d \n", b);

    if(b > 0 && i < filesize -1)
    {
        write(client_socket,writeBuff,1023);
    }

    if(b > 0 && i == filesize-1)                 //This is to handle the last chunk of the
file
    {
        write(client_socket, writeBuff, left);
    }

    if (i == filesize -1 || b < 1023 )           //Checking if we have any error while
writing the file
    {
        if (feof(fp1))
            printf("File transfer complete\n");

        if (ferror(fp1))
            printf("Incomplete file transfer, error\n");
        break;
    }
    i = i + 1;

}

}

void sendhead(int client_socket, int flag, char* Type,int filesize,char *contype)      //This
function is used to send the various status code in the response header
{
  char *resp;
  char *file_size;
```

```c
resp = (char*)malloc(1024*sizeof(char));
file_size = (char*)malloc(1024*sizeof(char));

bzero((char *)resp, sizeof(resp));
bzero((char *)file_size, sizeof(file_size));

char text[100];
time_t now = time(NULL);

struct tm *t = localtime(&now);

strftime(text, sizeof(text)-1, "%a, %d %b %Y %H:%M", t);     //This is used to find the
date and time

if(flag == 0)
{
    sprintf(file_size, "%d", filesize);                    //Flag 0 is the case when the status
code is supposed to be 200
    strcat(resp, "HTTP/1.1 200 OK\r\n");
    strcat(resp, "Date: ");
    strcat(resp, text);
    strcat(resp, "\r\n");
    strcat (resp, "Server: Random server\r\n");
    strcat(resp, "Content-Length: ");
    strcat(resp, file_size);
    strcat(resp, "\r\n");
    strcat(resp, "Content-Type: ");
    strcat(resp, Type);
    strcat(resp, "\r\n");
    strcat(resp, "Connection: ");
    strcat(resp, contype);
    strcat(resp, "\r\n");
    strcat(resp, "\r\n");
    write(client_socket, resp,strlen(resp));
}

if(flag == 1)
{
    strcat(resp, "HTTP/1.1 404 Not Found\r\n");            //Flag 1 is the case when we did
not find the file s
    strcat(resp, "Date: ");
    strcat(resp, text);
    strcat(resp, "\r\n");
    strcat(resp, "Content-Length: 143");
    strcat(resp, "\r\n");
    strcat(resp, "Content-Type: text/html");
    strcat(resp, "\r\n");
    strcat(resp, "\r\n");
```

```c
    strcat(resp, "<html><head><TITLE>404 Not Found</TITLE></head> \r
\n<body><H1>404 Not Found</H1>\r\nThe requested URL was not found on this server.\r
\n</body></html>");
    write(client_socket, resp,strlen(resp));
  }

  if(flag == 2)
  {
    strcat(resp, "HTTP/1.1 400 Bad Request\r\n");        //Flag 2 is when we made a bad
request to the client
    strcat(resp, "Date: ");
    strcat(resp, text);
    strcat(resp, "\r\n");
    strcat(resp, "Content-Length: 143");
    strcat(resp, "\r\n");
    strcat(resp, "Content-Type: text/html");
    strcat(resp, "\r\n");
    strcat(resp, "\r\n");
    strcat(resp, "<html><head><TITLE>404 Not Found</TITLE></head> \r
\n<body><H1>404 Not Found</H1>\r\nThe requested URL was not found on this server.\r
\n</body></html>");
    write(client_socket, resp,strlen(resp));
  }

  if(flag == 3)
  {
    strcat(resp, "HTTP/1.1 500 Internal Server Error\r\n");   //Flag 3 is when we are having
a problem is the server itself
    strcat(resp, "Date: ");
    strcat(resp, text);
    strcat(resp, "\r\n");
    strcat(resp, "Content-Length: 155");
    strcat(resp, "\r\n");
    strcat(resp, "Content-Type: text/html");
    strcat(resp, "\r\n");
    strcat(resp, "\r\n");
    strcat(resp, "<html><head><TITLE>500 Internal Server Error</TITLE></head> \r
\n<body><H1>500 Internal Server Error</H1>\r\nThere is some error on this server.\r\n</
body></html>");
    write(client_socket, resp,strlen(resp));
  }

  if(flag == 4)
  {
    strcat(resp, "HTTP/1.1 501 Not Implemented\r\n");        //Flag 4 is for the class of
requests which have not been implemented
    strcat(resp, "Date: ");
    strcat(resp, text);
    strcat(resp, "\r\n");
    strcat(resp, "Content-Length: 163");
```

```c
        strcat(resp, "\r\n");
        strcat(resp, "Content-Type: text/html");
        strcat(resp, "\r\n");
        strcat(resp, "\r\n");
        strcat(resp, "<html><head><TITLE>501 Not Implemented</TITLE></head> \r
\n<body><H1>501 Not Implemented</H1>\r\nThe requested method is not implemented
on this server.\r\n</body></html>");
        write(client_socket, resp,strlen(resp));
    }

}


void logfile(FILE *lgfp, char *uri, char *ip )          //This function is used to write the
request in the log file
{
  char *resp;

  resp = (char*)malloc(1024*sizeof(char));
  bzero((char *)resp, sizeof(resp));


  char text[100];
  time_t now = time(NULL);

  struct tm *t = localtime(&now);
  strftime(text, sizeof(text)-1, "%a, %d %b %Y %H:%M", t);  //This is used to get the time at
which the reques is made.

  strcat(resp, text);
  strcat(resp, " ");
  strcat(resp, ip);
  strcat(resp, " ");
  strcat(resp, uri);
  strcat(resp, "\n");

  fwrite(resp,1,strlen(resp), lgfp);          //Thi is used to write to the file that we use
as log
  fflush(lgfp);
}
```

LOG FILE :-

Fri, 19 Aug 2016 14:42 0.0.0.0 index.html
Fri, 19 Aug 2016 14:42 0.0.0.0 images/vt.gif
Fri, 19 Aug 2016 14:42 127.0.0.1 images/neb001.jpg
Fri, 19 Aug 2016 14:42 127.0.0.1 images/thm001.gif
Fri, 19 Aug 2016 14:42 127.0.0.1 images/neb002.jpg
Fri, 19 Aug 2016 14:42 127.0.0.1 images/thm002.gif
Fri, 19 Aug 2016 14:42 0.0.0.0 images/neb003.jpg
Fri, 19 Aug 2016 14:42 127.0.0.1 images/thm003.gif
Fri, 19 Aug 2016 14:42 127.0.0.1 images/thm004.gif
Fri, 19 Aug 2016 14:42 127.0.0.1 images/neb004.jpg
Fri, 19 Aug 2016 14:42 127.0.0.1 images/neb005.jpg
Fri, 19 Aug 2016 14:42 127.0.0.1 images/thm005.gif
Fri, 19 Aug 2016 14:42 0.0.0.0 images/thm006.jpg
Fri, 19 Aug 2016 14:42 127.0.0.1 images/neb006.jpg
Fri, 19 Aug 2016 14:42 127.0.0.1 marina.jpg
Fri, 19 Aug 2016 14:42 127.0.0.1 potomac.jpg
Fri, 19 Aug 2016 14:42 0.0.0.0 kingstrt.jpg
Fri, 19 Aug 2016 14:42 127.0.0.1 images/badfile.gif
Fri, 19 Aug 2016 16:00 127.0.0.1 images/thm002.gif
Fri, 19 Aug 2016 16:00 127.0.0.1 potomac.jpg
Fri, 19 Aug 2016 16:00 127.0.0.1 marina.jpg
Fri, 19 Aug 2016 16:00 127.0.0.1 images/neb006.jpg
Fri, 19 Aug 2016 16:00 127.0.0.1 images/badfile.gif
Fri, 19 Aug 2016 16:00 127.0.0.1 images/badfile.gif
Fri, 19 Aug 2016 16:00 127.0.0.1 images/badfile.gif
Fri, 19 Aug 2016 16:00 127.0.0.1 images/badfile.gif
Fri, 19 Aug 2016 16:00 0.0.0.0 kingstrt.jpg
Fri, 19 Aug 2016 18:30 127.0.0.1 index.html
Fri, 19 Aug 2016 18:30 127.0.0.1 images/vt.gif
Fri, 19 Aug 2016 18:30 127.0.0.1 images/neb001.jpg
Fri, 19 Aug 2016 18:30 127.0.0.1 images/thm001.gif
Fri, 19 Aug 2016 18:30 127.0.0.1 images/neb002.jpg
Fri, 19 Aug 2016 18:30 127.0.0.1 images/thm002.gif
Fri, 19 Aug 2016 18:30 127.0.0.1 images/neb003.jpg
Fri, 19 Aug 2016 18:30 127.0.0.1 images/thm003.gif
Fri, 19 Aug 2016 18:30 127.0.0.1 images/neb004.jpg
Fri, 19 Aug 2016 18:30 127.0.0.1 images/thm004.gif
Fri, 19 Aug 2016 18:30 127.0.0.1 images/neb005.jpg
Fri, 19 Aug 2016 18:30 127.0.0.1 images/thm005.gif
Fri, 19 Aug 2016 18:30 127.0.0.1 images/neb006.jpg
Fri, 19 Aug 2016 18:30 127.0.0.1 images/thm006.jpg
Fri, 19 Aug 2016 18:30 127.0.0.1 marina.jpg
Fri, 19 Aug 2016 18:30 127.0.0.1 potomac.jpg
Fri, 19 Aug 2016 18:30 127.0.0.1 kingstrt.jpg

Fri, 19 Aug 2016 18:30 127.0.0.1 images/badfile.gif
Fri, 19 Aug 2016 18:31 127.0.0.1 index.html
Fri, 19 Aug 2016 18:31 127.0.0.1 images/vt.gif
Fri, 19 Aug 2016 18:31 127.0.0.1 images/neb001.jpg
Fri, 19 Aug 2016 18:31 127.0.0.1 images/thm001.gif
Fri, 19 Aug 2016 18:31 127.0.0.1 images/neb002.jpg
Fri, 19 Aug 2016 18:31 127.0.0.1 images/thm002.gif
Fri, 19 Aug 2016 18:31 127.0.0.1 images/neb003.jpg
Fri, 19 Aug 2016 18:31 127.0.0.1 images/thm003.gif
Fri, 19 Aug 2016 18:31 127.0.0.1 images/neb004.jpg
Fri, 19 Aug 2016 18:31 127.0.0.1 images/thm004.gif
Fri, 19 Aug 2016 18:31 127.0.0.1 images/neb005.jpg
Fri, 19 Aug 2016 18:31 127.0.0.1 images/thm005.gif
Fri, 19 Aug 2016 18:31 127.0.0.1 images/neb006.jpg
Fri, 19 Aug 2016 18:31 127.0.0.1 images/thm006.jpg
Fri, 19 Aug 2016 18:31 127.0.0.1 marina.jpg
Fri, 19 Aug 2016 18:31 127.0.0.1 potomac.jpg
Fri, 19 Aug 2016 18:31 127.0.0.1 kingstrt.jpg
Fri, 19 Aug 2016 18:31 127.0.0.1 images/badfile.gif
Fri, 19 Aug 2016 18:40 127.0.0.1 images/thm006.jpg
Fri, 19 Aug 2016 18:40 127.0.0.1 marina.jpg
Fri, 19 Aug 2016 18:40 127.0.0.1 images/neb006.jpg
Fri, 19 Aug 2016 18:40 127.0.0.1 potomac.jpg
Fri, 19 Aug 2016 18:40 127.0.0.1 images/badfile.gif
Fri, 19 Aug 2016 18:40 127.0.0.1 images/badfile.gif
Fri, 19 Aug 2016 18:40 127.0.0.1 images/badfile.gif
Fri, 19 Aug 2016 18:40 127.0.0.1 images/badfile.gif
Fri, 19 Aug 2016 18:40 127.0.0.1 images/badfile.gif
Fri, 19 Aug 2016 18:40 127.0.0.1 kingstrt.jpg
Fri, 19 Aug 2016 18:40 127.0.0.1 index.html
Fri, 19 Aug 2016 18:40 127.0.0.1 images/vt.gif
Fri, 19 Aug 2016 18:40 127.0.0.1 images/neb001.jpg
Fri, 19 Aug 2016 18:40 127.0.0.1 images/thm001.gif
Fri, 19 Aug 2016 18:40 127.0.0.1 images/neb002.jpg
Fri, 19 Aug 2016 18:40 127.0.0.1 images/neb003.jpg
Fri, 19 Aug 2016 18:40 127.0.0.1 images/thm002.gif
Fri, 19 Aug 2016 18:40 127.0.0.1 images/thm003.gif
Fri, 19 Aug 2016 18:40 127.0.0.1 images/neb004.jpg
Fri, 19 Aug 2016 18:40 127.0.0.1 images/thm004.gif
Fri, 19 Aug 2016 18:40 127.0.0.1 images/neb005.jpg
Fri, 19 Aug 2016 18:40 127.0.0.1 images/thm005.gif
Fri, 19 Aug 2016 18:40 127.0.0.1 images/neb006.jpg
Fri, 19 Aug 2016 18:40 127.0.0.1 images/thm006.jpg
Fri, 19 Aug 2016 18:40 127.0.0.1 marina.jpg
Fri, 19 Aug 2016 18:40 127.0.0.1 potomac.jpg
Fri, 19 Aug 2016 18:40 127.0.0.1 kingstrt.jpg
Fri, 19 Aug 2016 18:40 127.0.0.1 images/badfile.gif
Fri, 19 Aug 2016 18:40 127.0.0.1 test.pdf
Fri, 19 Aug 2016 18:40 127.0.0.1 test.pdf
Fri, 19 Aug 2016 18:46 127.0.0.1 images/neb001.jpg

Fri, 19 Aug 2016 18:46 127.0.0.1 images/neb002.jpg
Fri, 19 Aug 2016 18:46 127.0.0.1 images/thm002.gif
Fri, 19 Aug 2016 18:46 127.0.0.1 images/thm006.jpg
Fri, 19 Aug 2016 18:46 127.0.0.1 test.pdf
Fri, 19 Aug 2016 18:46 127.0.0.1 log.txt
Fri, 19 Aug 2016 18:47 127.0.0.1 post_test.html
Fri, 19 Aug 2016 18:47 127.0.0.1 index.html
Fri, 19 Aug 2016 18:47 127.0.0.1 images/vt.gif
Fri, 19 Aug 2016 18:47 127.0.0.1 images/neb001.jpg
Fri, 19 Aug 2016 18:47 127.0.0.1 images/thm001.gif
Fri, 19 Aug 2016 18:47 127.0.0.1 images/neb002.jpg
Fri, 19 Aug 2016 18:47 127.0.0.1 images/thm002.gif
Fri, 19 Aug 2016 18:47 127.0.0.1 images/neb003.jpg
Fri, 19 Aug 2016 18:47 127.0.0.1 images/thm003.gif
Fri, 19 Aug 2016 18:47 127.0.0.1 images/neb004.jpg
Fri, 19 Aug 2016 18:47 127.0.0.1 images/thm004.gif
Fri, 19 Aug 2016 18:47 127.0.0.1 images/neb005.jpg
Fri, 19 Aug 2016 18:47 127.0.0.1 images/thm005.gif
Fri, 19 Aug 2016 18:47 127.0.0.1 images/neb006.jpg
Fri, 19 Aug 2016 18:47 127.0.0.1 images/thm006.jpg
Fri, 19 Aug 2016 18:47 127.0.0.1 marina.jpg
Fri, 19 Aug 2016 18:47 127.0.0.1 potomac.jpg
Fri, 19 Aug 2016 18:47 127.0.0.1 kingstrt.jpg
Fri, 19 Aug 2016 18:47 127.0.0.1 images/badfile.gif
Fri, 19 Aug 2016 18:47 127.0.0.1 index.html
Fri, 19 Aug 2016 18:47 127.0.0.1 images/vt.gif
Fri, 19 Aug 2016 18:47 127.0.0.1 images/neb001.jpg
Fri, 19 Aug 2016 18:47 127.0.0.1 images/thm001.gif
Fri, 19 Aug 2016 18:47 127.0.0.1 images/neb002.jpg
Fri, 19 Aug 2016 18:47 127.0.0.1 images/thm002.gif
Fri, 19 Aug 2016 18:47 127.0.0.1 images/neb003.jpg
Fri, 19 Aug 2016 18:47 127.0.0.1 images/thm003.gif
Fri, 19 Aug 2016 18:47 127.0.0.1 images/neb004.jpg
Fri, 19 Aug 2016 18:47 127.0.0.1 images/thm004.gif
Fri, 19 Aug 2016 18:47 127.0.0.1 images/neb005.jpg
Fri, 19 Aug 2016 18:47 127.0.0.1 images/thm005.gif
Fri, 19 Aug 2016 18:47 127.0.0.1 images/neb006.jpg
Fri, 19 Aug 2016 18:47 127.0.0.1 images/thm006.jpg
Fri, 19 Aug 2016 18:47 127.0.0.1 marina.jpg
Fri, 19 Aug 2016 18:47 127.0.0.1 potomac.jpg
Fri, 19 Aug 2016 18:47 127.0.0.1 kingstrt.jpg
Fri, 19 Aug 2016 18:47 127.0.0.1 images/badfile.gif
Fri, 19 Aug 2016 18:49 127.0.0.1 text_sample.txt
Fri, 19 Aug 2016 18:49 127.0.0.1 text_sample.txt
Fri, 19 Aug 2016 18:50 127.0.0.1 images/neb001.jpg
Fri, 19 Aug 2016 18:55 127.0.0.1 images/thm006.jpg
Fri, 19 Aug 2016 18:55 127.0.0.1 images/neb006.jpg
Fri, 19 Aug 2016 18:55 127.0.0.1 potomac.jpg
Fri, 19 Aug 2016 18:55 127.0.0.1 marina.jpg
Fri, 19 Aug 2016 18:55 127.0.0.1 images/badfile.gif

Fri, 19 Aug 2016 18:55 127.0.0.1 images/badfile.gif
Fri, 19 Aug 2016 18:55 127.0.0.1 images/badfile.gif
Fri, 19 Aug 2016 18:55 127.0.0.1 images/neb001.jpg
Fri, 19 Aug 2016 18:55 127.0.0.1 badimage.jpg
Fri, 19 Aug 2016 19:14 127.0.0.1 badimage.jpg
Fri, 19 Aug 2016 19:14 127.0.0.1 badimage.jpg
Fri, 19 Aug 2016 20:52 0.0.0.0 index.html
Fri, 19 Aug 2016 20:52 0.0.0.0 images/vt.gif
Fri, 19 Aug 2016 20:52 127.0.0.1 images/neb001.jpg
Fri, 19 Aug 2016 20:52 127.0.0.1 images/thm001.gif
Fri, 19 Aug 2016 20:52 127.0.0.1 images/neb002.jpg
Fri, 19 Aug 2016 20:52 0.0.0.0 images/thm002.gif
Fri, 19 Aug 2016 20:52 127.0.0.1 images/neb003.jpg
Fri, 19 Aug 2016 20:52 0.0.0.0 images/thm003.gif
Fri, 19 Aug 2016 20:52 127.0.0.1 images/neb004.jpg
Fri, 19 Aug 2016 20:52 127.0.0.1 images/thm004.gif
Fri, 19 Aug 2016 20:52 127.0.0.1 images/neb005.jpg
Fri, 19 Aug 2016 20:52 127.0.0.1 images/thm005.gif
Fri, 19 Aug 2016 20:52 127.0.0.1 images/neb006.jpg
Fri, 19 Aug 2016 20:52 127.0.0.1 images/thm006.jpg
Fri, 19 Aug 2016 20:52 127.0.0.1 marina.jpg
Fri, 19 Aug 2016 20:52 0.0.0.0 potomac.jpg
Fri, 19 Aug 2016 20:52 127.0.0.1 kingstrt.jpg
Fri, 19 Aug 2016 20:52 127.0.0.1 images/badfile.gif
Fri, 19 Aug 2016 20:53 127.0.0.1 index.html
Fri, 19 Aug 2016 20:53 127.0.0.1 images/neb001.jpg
Fri, 19 Aug 2016 20:53 127.0.0.1 images/vt.gif
Fri, 19 Aug 2016 20:53 0.0.0.0 images/thm001.gif
Fri, 19 Aug 2016 20:53 127.0.0.1 images/neb002.jpg
Fri, 19 Aug 2016 20:53 127.0.0.1 images/thm002.gif
Fri, 19 Aug 2016 20:53 127.0.0.1 images/neb003.jpg
Fri, 19 Aug 2016 20:53 127.0.0.1 images/thm003.gif
Fri, 19 Aug 2016 20:53 127.0.0.1 images/thm004.gif
Fri, 19 Aug 2016 20:53 0.0.0.0 images/neb004.jpg
Fri, 19 Aug 2016 20:53 127.0.0.1 images/neb005.jpg
Fri, 19 Aug 2016 20:53 127.0.0.1 images/thm005.gif
Fri, 19 Aug 2016 20:53 0.0.0.0 images/neb006.jpg
Fri, 19 Aug 2016 20:53 127.0.0.1 images/thm006.jpg
Fri, 19 Aug 2016 20:53 127.0.0.1 marina.jpg
Fri, 19 Aug 2016 20:53 127.0.0.1 potomac.jpg
Fri, 19 Aug 2016 20:53 127.0.0.1 kingstrt.jpg
Fri, 19 Aug 2016 20:53 127.0.0.1 images/badfile.gif
Fri, 19 Aug 2016 20:53 127.0.0.1 index.html
Fri, 19 Aug 2016 20:53 127.0.0.1 images/vt.gif
Fri, 19 Aug 2016 20:53 127.0.0.1 images/neb001.jpg
Fri, 19 Aug 2016 20:53 0.0.0.0 images/thm001.gif
Fri, 19 Aug 2016 20:53 127.0.0.1 images/neb002.jpg
Fri, 19 Aug 2016 20:53 127.0.0.1 images/thm002.gif
Fri, 19 Aug 2016 20:53 127.0.0.1 images/neb003.jpg
Fri, 19 Aug 2016 20:53 127.0.0.1 images/thm003.gif

Fri, 19 Aug 2016 20:53 127.0.0.1 images/neb004.jpg
Fri, 19 Aug 2016 20:53 127.0.0.1 images/thm004.gif
Fri, 19 Aug 2016 20:53 127.0.0.1 images/neb005.jpg
Fri, 19 Aug 2016 20:53 127.0.0.1 images/thm005.gif
Fri, 19 Aug 2016 20:53 127.0.0.1 images/neb006.jpg
Fri, 19 Aug 2016 20:53 127.0.0.1 images/thm006.jpg
Fri, 19 Aug 2016 20:53 127.0.0.1 marina.jpg
Fri, 19 Aug 2016 20:53 127.0.0.1 potomac.jpg
Fri, 19 Aug 2016 20:53 127.0.0.1 kingstrt.jpg
Fri, 19 Aug 2016 20:53 0.0.0.0 images/badfile.gif
Fri, 19 Aug 2016 20:53 127.0.0.1 index.html
Fri, 19 Aug 2016 20:53 127.0.0.1 images/vt.gif
Fri, 19 Aug 2016 20:53 0.0.0.0 images/neb001.jpg
Fri, 19 Aug 2016 20:53 127.0.0.1 images/thm001.gif
Fri, 19 Aug 2016 20:53 127.0.0.1 images/neb002.jpg
Fri, 19 Aug 2016 20:53 127.0.0.1 images/thm002.gif
Fri, 19 Aug 2016 20:53 127.0.0.1 images/neb003.jpg
Fri, 19 Aug 2016 20:53 127.0.0.1 images/thm003.gif
Fri, 19 Aug 2016 20:53 127.0.0.1 images/neb004.jpg
Fri, 19 Aug 2016 20:53 127.0.0.1 images/thm004.gif
Fri, 19 Aug 2016 20:53 127.0.0.1 images/neb005.jpg
Fri, 19 Aug 2016 20:53 0.0.0.0 images/thm005.gif
Fri, 19 Aug 2016 20:53 127.0.0.1 images/neb006.jpg
Fri, 19 Aug 2016 20:53 127.0.0.1 images/thm006.jpg
Fri, 19 Aug 2016 20:53 0.0.0.0 marina.jpg
Fri, 19 Aug 2016 20:53 127.0.0.1 potomac.jpg
Fri, 19 Aug 2016 20:53 127.0.0.1 kingstrt.jpg
Fri, 19 Aug 2016 20:53 127.0.0.1 images/badfile.gif
Fri, 19 Aug 2016 20:59 0.0.0.0 index.html
Fri, 19 Aug 2016 20:59 0.0.0.0 images/vt.gif
Fri, 19 Aug 2016 20:59 0.0.0.0 images/neb001.jpg
Fri, 19 Aug 2016 20:59 127.0.0.1 images/thm001.gif
Fri, 19 Aug 2016 20:59 0.0.0.0 images/thm002.gif
Fri, 19 Aug 2016 20:59 127.0.0.1 images/neb002.jpg
Fri, 19 Aug 2016 20:59 127.0.0.1 images/neb003.jpg
Fri, 19 Aug 2016 20:59 0.0.0.0 images/thm003.gif
Fri, 19 Aug 2016 20:59 127.0.0.1 images/neb004.jpg
Fri, 19 Aug 2016 20:59 127.0.0.1 images/thm004.gif
Fri, 19 Aug 2016 20:59 127.0.0.1 images/neb005.jpg
Fri, 19 Aug 2016 20:59 127.0.0.1 images/thm005.gif
Fri, 19 Aug 2016 20:59 127.0.0.1 images/neb006.jpg
Fri, 19 Aug 2016 20:59 127.0.0.1 images/thm006.jpg
Fri, 19 Aug 2016 20:59 127.0.0.1 marina.jpg
Fri, 19 Aug 2016 20:59 127.0.0.1 potomac.jpg
Fri, 19 Aug 2016 20:59 127.0.0.1 kingstrt.jpg
Fri, 19 Aug 2016 20:59 127.0.0.1 images/badfile.gif
Fri, 19 Aug 2016 20:59 127.0.0.1 test.pdf
Fri, 19 Aug 2016 20:59 127.0.0.1 test.pdf
Fri, 19 Aug 2016 21:02 0.0.0.0 test.pdf
Fri, 19 Aug 2016 21:02 0.0.0.0 test.pdf

Fri, 19 Aug 2016 21:03 0.0.0.0 test.pdf
Fri, 19 Aug 2016 21:03 0.0.0.0 test.pdf
Fri, 19 Aug 2016 21:03 0.0.0.0 test.pdf
Fri, 19 Aug 2016 21:05 0.0.0.0 test.pdf
Fri, 19 Aug 2016 21:05 0.0.0.0 test.pdf
Fri, 19 Aug 2016 21:06 0.0.0.0
Fri, 19 Aug 2016 21:06 0.0.0.0 index.html
Fri, 19 Aug 2016 21:06 0.0.0.0 images/vt.gif
Fri, 19 Aug 2016 21:06 127.0.0.1 images/neb001.jpg
Fri, 19 Aug 2016 21:06 127.0.0.1 images/thm001.gif
Fri, 19 Aug 2016 21:06 127.0.0.1 images/neb002.jpg
Fri, 19 Aug 2016 21:06 0.0.0.0 images/thm002.gif
Fri, 19 Aug 2016 21:06 127.0.0.1 images/neb003.jpg
Fri, 19 Aug 2016 21:06 127.0.0.1 images/thm003.gif
Fri, 19 Aug 2016 21:06 127.0.0.1 images/neb004.jpg
Fri, 19 Aug 2016 21:06 127.0.0.1 images/thm004.gif
Fri, 19 Aug 2016 21:06 0.0.0.0 images/neb005.jpg
Fri, 19 Aug 2016 21:06 127.0.0.1 images/thm005.gif
Fri, 19 Aug 2016 21:06 127.0.0.1 images/neb006.jpg
Fri, 19 Aug 2016 21:06 127.0.0.1 images/thm006.jpg
Fri, 19 Aug 2016 21:06 0.0.0.0 marina.jpg
Fri, 19 Aug 2016 21:06 127.0.0.1 potomac.jpg
Fri, 19 Aug 2016 21:06 127.0.0.1 kingstrt.jpg
Fri, 19 Aug 2016 21:06 127.0.0.1 images/badfile.gif
Fri, 19 Aug 2016 21:06 127.0.0.1
Fri, 19 Aug 2016 21:07 0.0.0.0 index.html
Fri, 19 Aug 2016 21:07 0.0.0.0 images/vt.gif
Fri, 19 Aug 2016 21:07 127.0.0.1 images/neb001.jpg
Fri, 19 Aug 2016 21:07 127.0.0.1 images/thm001.gif
Fri, 19 Aug 2016 21:07 127.0.0.1 images/neb002.jpg
Fri, 19 Aug 2016 21:07 127.0.0.1 images/thm002.gif
Fri, 19 Aug 2016 21:07 127.0.0.1 images/neb003.jpg
Fri, 19 Aug 2016 21:07 127.0.0.1 images/thm003.gif
Fri, 19 Aug 2016 21:07 127.0.0.1 images/neb004.jpg
Fri, 19 Aug 2016 21:07 127.0.0.1 images/thm004.gif
Fri, 19 Aug 2016 21:07 127.0.0.1 images/neb005.jpg
Fri, 19 Aug 2016 21:07 127.0.0.1 images/thm005.gif
Fri, 19 Aug 2016 21:07 127.0.0.1 images/neb006.jpg
Fri, 19 Aug 2016 21:07 127.0.0.1 images/thm006.jpg
Fri, 19 Aug 2016 21:07 127.0.0.1 marina.jpg
Fri, 19 Aug 2016 21:07 127.0.0.1 potomac.jpg
Fri, 19 Aug 2016 21:07 127.0.0.1 kingstrt.jpg
Fri, 19 Aug 2016 21:07 127.0.0.1 images/badfile.gif
Fri, 19 Aug 2016 21:07 127.0.0.1
Fri, 19 Aug 2016 21:08 0.0.0.0 test.pdf
Fri, 19 Aug 2016 21:08 0.0.0.0 test.pdf
Fri, 19 Aug 2016 21:08 0.0.0.0 test.pdf
Fri, 19 Aug 2016 21:08 0.0.0.0
Fri, 19 Aug 2016 21:10 0.0.0.0 test.pdf
Fri, 19 Aug 2016 21:10 0.0.0.0 test.pdf

Fri, 19 Aug 2016 21:10 0.0.0.0 favicon.ico
Fri, 19 Aug 2016 21:10 0.0.0.0 test.pdf
Fri, 19 Aug 2016 21:10 0.0.0.0 test.pdf
Fri, 19 Aug 2016 21:10 0.0.0.0
Fri, 19 Aug 2016 21:11 0.0.0.0 index.html
Fri, 19 Aug 2016 21:11 0.0.0.0 images/vt.gif
Fri, 19 Aug 2016 21:11 127.0.0.1 images/neb001.jpg
Fri, 19 Aug 2016 21:11 127.0.0.1 images/thm001.gif
Fri, 19 Aug 2016 21:11 127.0.0.1 images/neb002.jpg
Fri, 19 Aug 2016 21:11 0.0.0.0 images/thm002.gif
Fri, 19 Aug 2016 21:11 127.0.0.1 images/neb003.jpg
Fri, 19 Aug 2016 21:11 127.0.0.1 images/thm003.gif
Fri, 19 Aug 2016 21:11 0.0.0.0 images/neb004.jpg
Fri, 19 Aug 2016 21:11 127.0.0.1 images/thm004.gif
Fri, 19 Aug 2016 21:11 127.0.0.1 images/neb005.jpg
Fri, 19 Aug 2016 21:11 127.0.0.1 images/thm005.gif
Fri, 19 Aug 2016 21:11 0.0.0.0 images/neb006.jpg
Fri, 19 Aug 2016 21:11 127.0.0.1 images/thm006.jpg
Fri, 19 Aug 2016 21:11 127.0.0.1 marina.jpg
Fri, 19 Aug 2016 21:11 127.0.0.1 potomac.jpg
Fri, 19 Aug 2016 21:11 127.0.0.1 kingstrt.jpg
Fri, 19 Aug 2016 21:11 127.0.0.1 images/badfile.gif
Fri, 19 Aug 2016 21:11 127.0.0.1 test.pdf
Fri, 19 Aug 2016 21:11 127.0.0.1 test.pdf
Fri, 19 Aug 2016 21:11 127.0.0.1
Fri, 19 Aug 2016 21:18 0.0.0.0 test.pdf
Fri, 19 Aug 2016 21:18 0.0.0.0 test.pdf
Fri, 19 Aug 2016 21:18 0.0.0.0 index.html
Fri, 19 Aug 2016 21:18 0.0.0.0 images/vt.gif
Fri, 19 Aug 2016 21:18 0.0.0.0 images/neb001.jpg
Fri, 19 Aug 2016 21:18 127.0.0.1 images/thm001.gif
Fri, 19 Aug 2016 21:18 127.0.0.1 images/neb002.jpg
Fri, 19 Aug 2016 21:18 127.0.0.1 images/thm002.gif
Fri, 19 Aug 2016 21:18 127.0.0.1 images/neb003.jpg
Fri, 19 Aug 2016 21:18 127.0.0.1 images/thm003.gif
Fri, 19 Aug 2016 21:18 127.0.0.1 images/neb004.jpg
Fri, 19 Aug 2016 21:18 127.0.0.1 images/thm004.gif
Fri, 19 Aug 2016 21:18 127.0.0.1 images/neb005.jpg
Fri, 19 Aug 2016 21:18 0.0.0.0 images/thm005.gif
Fri, 19 Aug 2016 21:18 127.0.0.1 images/neb006.jpg
Fri, 19 Aug 2016 21:18 127.0.0.1 images/thm006.jpg
Fri, 19 Aug 2016 21:18 127.0.0.1 marina.jpg
Fri, 19 Aug 2016 21:18 127.0.0.1 potomac.jpg
Fri, 19 Aug 2016 21:18 127.0.0.1 kingstrt.jpg
Fri, 19 Aug 2016 21:18 0.0.0.0 images/badfile.gif
Fri, 19 Aug 2016 21:19 0.0.0.0 index.html
Fri, 19 Aug 2016 21:19 0.0.0.0 images/vt.gif
Fri, 19 Aug 2016 21:19 0.0.0.0 images/neb001.jpg
Fri, 19 Aug 2016 21:19 127.0.0.1 images/thm001.gif
Fri, 19 Aug 2016 21:19 127.0.0.1 images/thm002.gif

Fri, 19 Aug 2016 21:19 127.0.0.1 images/neb002.jpg
Fri, 19 Aug 2016 21:19 127.0.0.1 images/neb003.jpg
Fri, 19 Aug 2016 21:19 127.0.0.1 images/thm003.gif
Fri, 19 Aug 2016 21:19 127.0.0.1 images/neb004.jpg
Fri, 19 Aug 2016 21:19 127.0.0.1 images/thm004.gif
Fri, 19 Aug 2016 21:19 127.0.0.1 images/neb005.jpg
Fri, 19 Aug 2016 21:19 0.0.0.0 images/thm005.gif
Fri, 19 Aug 2016 21:19 127.0.0.1 images/neb006.jpg
Fri, 19 Aug 2016 21:19 127.0.0.1 images/thm006.jpg
Fri, 19 Aug 2016 21:19 127.0.0.1 marina.jpg
Fri, 19 Aug 2016 21:19 127.0.0.1 potomac.jpg
Fri, 19 Aug 2016 21:19 127.0.0.1 kingstrt.jpg
Fri, 19 Aug 2016 21:19 0.0.0.0 images/badfile.gif
Fri, 19 Aug 2016 21:19 127.0.0.1 index.html
Fri, 19 Aug 2016 21:19 127.0.0.1 images/vt.gif
Fri, 19 Aug 2016 21:19 0.0.0.0 images/neb001.jpg
Fri, 19 Aug 2016 21:19 127.0.0.1 images/thm001.gif
Fri, 19 Aug 2016 21:19 127.0.0.1 images/neb002.jpg
Fri, 19 Aug 2016 21:19 127.0.0.1 images/thm002.gif
Fri, 19 Aug 2016 21:19 127.0.0.1 images/neb003.jpg
Fri, 19 Aug 2016 21:19 127.0.0.1 images/thm003.gif
Fri, 19 Aug 2016 21:19 127.0.0.1 images/neb004.jpg
Fri, 19 Aug 2016 21:19 127.0.0.1 images/thm004.gif
Fri, 19 Aug 2016 21:19 0.0.0.0 images/neb005.jpg
Fri, 19 Aug 2016 21:19 127.0.0.1 images/thm005.gif
Fri, 19 Aug 2016 21:19 127.0.0.1 images/neb006.jpg
Fri, 19 Aug 2016 21:19 0.0.0.0 images/thm006.jpg
Fri, 19 Aug 2016 21:19 127.0.0.1 marina.jpg
Fri, 19 Aug 2016 21:19 127.0.0.1 kingstrt.jpg
Fri, 19 Aug 2016 21:19 127.0.0.1 potomac.jpg
Fri, 19 Aug 2016 21:19 0.0.0.0 images/badfile.gif
Fri, 19 Aug 2016 21:19 127.0.0.1 images/page.html
Fri, 19 Aug 2016 21:20 127.0.0.1 index.html
Fri, 19 Aug 2016 21:20 127.0.0.1 images/vt.gif
Fri, 19 Aug 2016 21:20 0.0.0.0 images/neb001.jpg
Fri, 19 Aug 2016 21:20 127.0.0.1 images/thm001.gif
Fri, 19 Aug 2016 21:20 127.0.0.1 images/neb002.jpg
Fri, 19 Aug 2016 21:20 127.0.0.1 images/thm002.gif
Fri, 19 Aug 2016 21:20 127.0.0.1 images/neb003.jpg
Fri, 19 Aug 2016 21:20 127.0.0.1 images/thm003.gif
Fri, 19 Aug 2016 21:20 0.0.0.0 images/neb004.jpg
Fri, 19 Aug 2016 21:20 127.0.0.1 images/thm004.gif
Fri, 19 Aug 2016 21:20 127.0.0.1 images/neb005.jpg
Fri, 19 Aug 2016 21:20 127.0.0.1 images/thm005.gif
Fri, 19 Aug 2016 21:20 127.0.0.1 images/neb006.jpg
Fri, 19 Aug 2016 21:20 127.0.0.1 images/thm006.gif
Fri, 19 Aug 2016 21:20 127.0.0.1 marina.jpg
Fri, 19 Aug 2016 21:20 127.0.0.1 potomac.jpg
Fri, 19 Aug 2016 21:20 0.0.0.0 kingstrt.jpg
Fri, 19 Aug 2016 21:20 127.0.0.1 images/badfile.gif

Fri, 19 Aug 2016 21:20 0.0.0.0 index.html
Fri, 19 Aug 2016 21:20 0.0.0.0 images/vt.gif
Fri, 19 Aug 2016 21:20 127.0.0.1 images/neb001.jpg
Fri, 19 Aug 2016 21:20 127.0.0.1 images/thm001.gif
Fri, 19 Aug 2016 21:20 127.0.0.1 images/neb002.jpg
Fri, 19 Aug 2016 21:20 127.0.0.1 images/thm002.gif
Fri, 19 Aug 2016 21:20 127.0.0.1 images/neb003.jpg
Fri, 19 Aug 2016 21:20 127.0.0.1 images/thm003.gif
Fri, 19 Aug 2016 21:20 127.0.0.1 images/neb004.jpg
Fri, 19 Aug 2016 21:20 127.0.0.1 images/thm004.gif
Fri, 19 Aug 2016 21:20 127.0.0.1 images/neb005.jpg
Fri, 19 Aug 2016 21:20 127.0.0.1 images/thm005.gif
Fri, 19 Aug 2016 21:20 127.0.0.1 images/neb006.jpg
Fri, 19 Aug 2016 21:20 0.0.0.0 images/thm006.jpg
Fri, 19 Aug 2016 21:20 127.0.0.1 marina.jpg
Fri, 19 Aug 2016 21:20 0.0.0.0 potomac.jpg
Fri, 19 Aug 2016 21:20 127.0.0.1 kingstrt.jpg
Fri, 19 Aug 2016 21:20 127.0.0.1 images/badfile.gif
Fri, 19 Aug 2016 21:20 127.0.0.1 index.html
Fri, 19 Aug 2016 21:20 127.0.0.1 images/vt.gif
Fri, 19 Aug 2016 21:20 127.0.0.1 images/neb001.jpg
Fri, 19 Aug 2016 21:20 0.0.0.0 images/thm001.gif
Fri, 19 Aug 2016 21:20 127.0.0.1 images/neb002.jpg
Fri, 19 Aug 2016 21:20 127.0.0.1 images/thm002.gif
Fri, 19 Aug 2016 21:20 127.0.0.1 images/neb003.jpg
Fri, 19 Aug 2016 21:20 127.0.0.1 images/thm003.gif
Fri, 19 Aug 2016 21:20 127.0.0.1 images/neb004.jpg
Fri, 19 Aug 2016 21:20 127.0.0.1 images/thm004.gif
Fri, 19 Aug 2016 21:20 127.0.0.1 images/neb005.jpg
Fri, 19 Aug 2016 21:20 127.0.0.1 images/thm005.gif
Fri, 19 Aug 2016 21:20 0.0.0.0 images/neb006.jpg
Fri, 19 Aug 2016 21:20 127.0.0.1 images/thm006.jpg
Fri, 19 Aug 2016 21:20 127.0.0.1 marina.jpg
Fri, 19 Aug 2016 21:20 127.0.0.1 potomac.jpg
Fri, 19 Aug 2016 21:20 127.0.0.1 kingstrt.jpg
Fri, 19 Aug 2016 21:20 0.0.0.0 images/badfile.gif
Fri, 19 Aug 2016 21:20 127.0.0.1 images/neb005.jpg
Fri, 19 Aug 2016 21:20 127.0.0.1 images/thm001.gif
Fri, 19 Aug 2016 21:20 127.0.0.1 pdf_sample.pdf
Fri, 19 Aug 2016 21:20 127.0.0.1 text_sample.txt
Fri, 19 Aug 2016 21:20 127.0.0.1 no_such_file.htm
Fri, 19 Aug 2016 21:29 0.0.0.0 index.html
Fri, 19 Aug 2016 21:29 0.0.0.0 images/vt.gif
Fri, 19 Aug 2016 21:29 0.0.0.0 images/neb001.jpg
Fri, 19 Aug 2016 21:29 127.0.0.1 images/thm001.gif
Fri, 19 Aug 2016 21:29 127.0.0.1 images/neb002.jpg
Fri, 19 Aug 2016 21:29 0.0.0.0 images/thm002.gif
Fri, 19 Aug 2016 21:29 127.0.0.1 images/neb003.jpg
Fri, 19 Aug 2016 21:29 127.0.0.1 images/thm003.gif
Fri, 19 Aug 2016 21:29 127.0.0.1 images/neb004.jpg

Fri, 19 Aug 2016 21:29 127.0.0.1 images/thm004.gif
Fri, 19 Aug 2016 21:29 127.0.0.1 images/neb005.jpg
Fri, 19 Aug 2016 21:29 127.0.0.1 images/thm005.gif
Fri, 19 Aug 2016 21:29 127.0.0.1 images/neb006.jpg
Fri, 19 Aug 2016 21:29 127.0.0.1 images/thm006.jpg
Fri, 19 Aug 2016 21:29 127.0.0.1 marina.jpg
Fri, 19 Aug 2016 21:29 127.0.0.1 potomac.jpg
Fri, 19 Aug 2016 21:29 127.0.0.1 kingstrt.jpg
Fri, 19 Aug 2016 21:29 127.0.0.1 images/badfile.gif
Fri, 19 Aug 2016 21:35 0.0.0.0 index.html
Fri, 19 Aug 2016 21:36 0.0.0.0 images/vt.gif
Fri, 19 Aug 2016 21:36 127.0.0.1 images/neb001.jpg
Fri, 19 Aug 2016 21:36 127.0.0.1 images/thm001.gif
Fri, 19 Aug 2016 21:36 127.0.0.1 images/neb002.jpg
Fri, 19 Aug 2016 21:36 127.0.0.1 images/thm002.gif
Fri, 19 Aug 2016 21:36 0.0.0.0 images/neb003.jpg
Fri, 19 Aug 2016 21:36 127.0.0.1 images/thm003.gif
Fri, 19 Aug 2016 21:36 127.0.0.1 images/neb004.jpg
Fri, 19 Aug 2016 21:36 127.0.0.1 images/thm004.gif
Fri, 19 Aug 2016 21:36 127.0.0.1 images/neb005.jpg
Fri, 19 Aug 2016 21:36 127.0.0.1 images/thm005.gif
Fri, 19 Aug 2016 21:36 127.0.0.1 images/neb006.jpg
Fri, 19 Aug 2016 21:36 127.0.0.1 images/thm006.jpg
Fri, 19 Aug 2016 21:36 127.0.0.1 marina.jpg
Fri, 19 Aug 2016 21:36 127.0.0.1 potomac.jpg
Fri, 19 Aug 2016 21:36 127.0.0.1 kingstrt.jpg
Fri, 19 Aug 2016 21:36 0.0.0.0 images/badfile.gif
Fri, 19 Aug 2016 21:36 127.0.0.1 test.pdf
Fri, 19 Aug 2016 21:36 127.0.0.1 test.pdf
Fri, 19 Aug 2016 21:36 127.0.0.1 test.pdf