

# KServer/KClient: Remote Clipboard Client-Server Console

**Author:** Kirti Chawla

**Email:** [kirti@cs.virginia.edu](mailto:kirti@cs.virginia.edu)

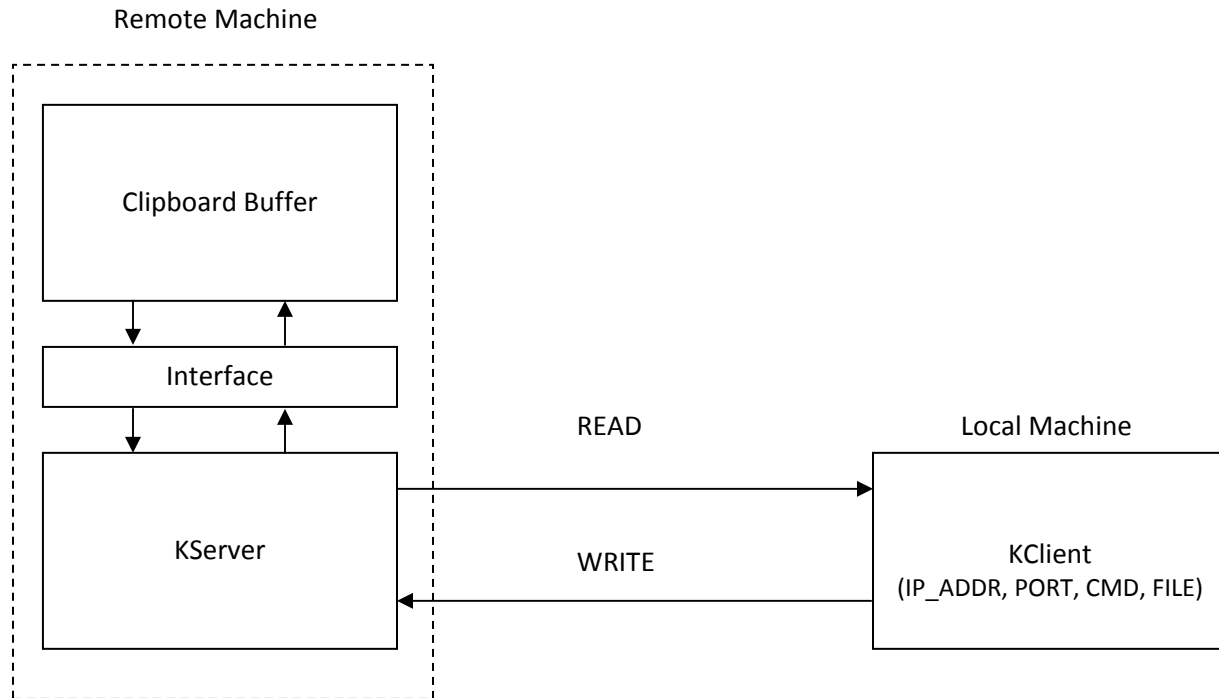
# Introduction

In our ever so busy and mundane life involving computing, we use clipboard extensively both locally and on web to copy/paste sensitive information (code, passwords, PINs, credit card details ...) between diverse applications. More often than not, we forget to clear the clipboard buffer. Furthermore, a target machine remains powered for quite a number of days in stretch. This is a potentially dangerous practice, which in the absence of clear buffer can open a door to sneak sensitive information out of safest of machines. Although, Windows Firewall needs to explicitly relax a port for demonstration of this tool. But perhaps there exists a programmatic way-around for this single line of defense. Indeed this tool can be used for malicious and good purposes. While the malicious purposes are self-evident, this tool can be used (with little programming effort) to remotely safe-wipe clipboard data.

In this document, a design and implementation of *KServer/KClient: Remote Clipboard Client-Server Console* is presented. First, a basic schematic of its architecture is given. Next, some illustrations of its working are given (as a proof of concept). The sole purpose of this experimental tool is to evaluate security threat posed by malicious process/service which reads/writes the clipboard buffer on the behest of a remote machine.

# Design

In this section, the design elements of *KServer/KClient: Remote Clipboard Client-Server Console* are presented. The details are given as under:

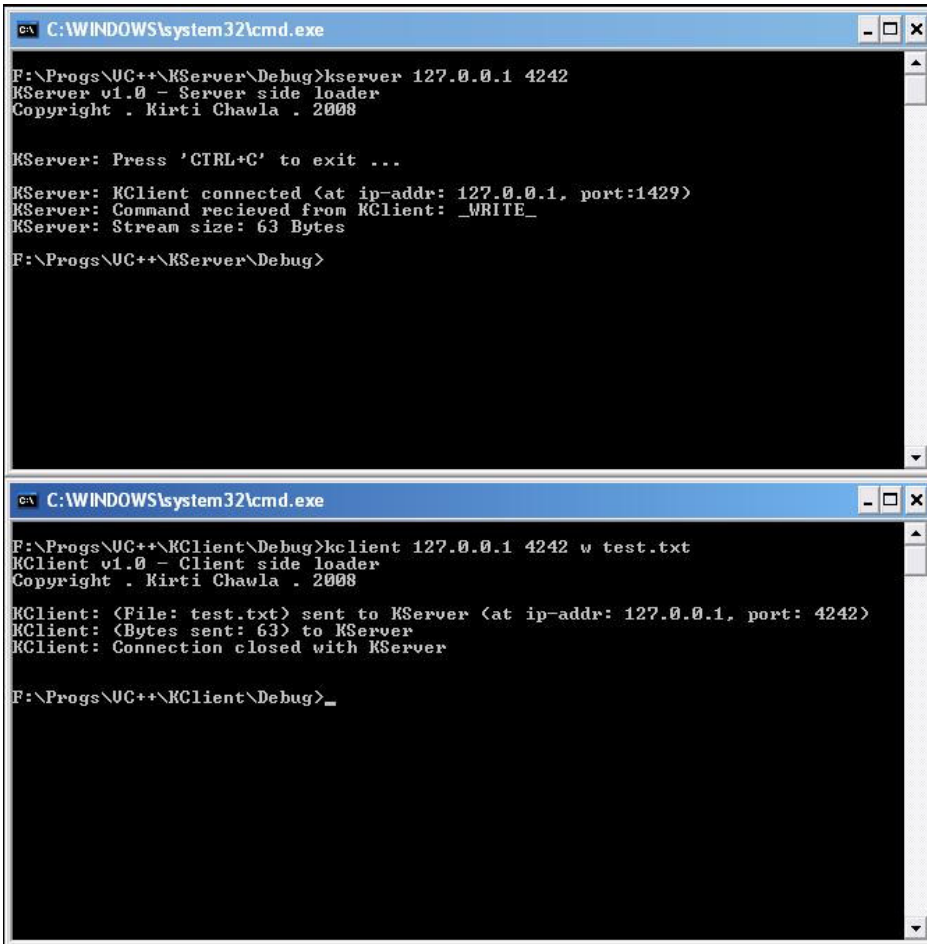


Where the terms are defined as under:

1. *Clipboard Buffer*: It is the buffer of clipboard, which is used to store data (image, audio, video, text ...)
2. *Interface*: It is programmatic methodology (API) through, which clipboard buffer can be manipulated.
3. *KServer*: It is the program that runs on remote machine, whose clipboard needs to be manipulated.
4. *KClient*: It is the program, which runs on local machine that wants to manipulate remote clipboard.
5. *IP\_ADDR*: It is the IP address of the machine on which *KServer* is in running state.
6. *PORT*: It is the port on which *KServer* is listening for incoming requests for connection.
7. *CMD*: It is the command that *KClient* wants to execute on *KServer*.
8. **READ/WRITE**: *KClient* and *KServer* communicate using command/response mechanism.
9. *FILE*: It is name of local filename that is used to read/write data from/to *KServer*.

# Implementation (Illustrations)

In this section, Illustrations of working of *KServer/KClient: Remote Clipboard Client-Server Console*, are provided. They are given as under:



```

C:\WINDOWS\system32\cmd.exe
F:\Progs\UC++\KServer\Debug>kserver 127.0.0.1 4242
KServer v1.0 - Server side loader
Copyright . Kirti Chawla . 2008

KServer: Press 'CTRL+C' to exit ...

KServer: KClient connected (at ip-addr: 127.0.0.1, port:1429)
KServer: Command recieved from KClient: _WRITE_
KServer: Stream size: 63 Bytes


F:\Progs\UC++\KServer\Debug>

C:\WINDOWS\system32\cmd.exe
F:\Progs\UC++\KClient\Debug>kclient 127.0.0.1 4242 w test.txt
KClient v1.0 - Client side loader
Copyright . Kirti Chawla . 2008

KClient: <File: test.txt> sent to KServer (at ip-addr: 127.0.0.1, port: 4242)
KClient: <Bytes sent: 63> to KServer
KClient: Connection closed with KServer

F:\Progs\UC++\KClient\Debug>_
  
```

Figure 1. KServer/KClient: remote WRITE command (i.e. writing onto clipboard remotely)



```

Untitled - Notepad
File Edit Format View Help
KClient can write into clipboard buffer from a remote machine
  
```

Figure 2. Notepad contents of file test.txt retrieved using CTRL+V

```

C:\WINDOWS\system32\cmd.exe - kserver 127.0.0.1 4242
F:\Progs\UC++\KServer\Debug>kserver 127.0.0.1 4242
KServer v1.0 - Server side loader
Copyright . Kirti Chawla . 2008

KServer: Press 'CTRL+C' to exit ...
KServer: KClient connected (at ip-addr: 127.0.0.1, port:1430)
KServer: Command recieved from KClient: _READ_
KServer: Stream size: 0 Bytes
KServer: Read from Clipboard done
KServer: Press 'CTRL+C' to exit ...

C:\WINDOWS\system32\cmd.exe
F:\Progs\UC++\KClient\Debug>kclient 127.0.0.1 4242 r output.txt
KClient v1.0 - Client side loader
Copyright . Kirti Chawla . 2008

KClient: (File: output.txt) recieved from KServer (at ip-addr: 127.0.0.1, port:
4242)
KClient: (Bytes recieved: 1024) from KServer
KClient: Connection closed with KServer

F:\Progs\UC++\KClient\Debug>_

```

Figure 3. KServer/KClient: remote READ command (i.e. reading from clipboard remotely)

```

Untitled - Notepad
File Edit Format View Help
kclient can also read data from a remote machine

```

Figure 4. Notepad contents of test message stored in clipboard buffer using CTRL+C

```

output.txt - Notepad
File Edit Format View Help
kclient can also read data from a remote machine

```

Figure 5. Notepad contents of output.txt read using KClient from KServer

# Suggestive Improvements

There can be a number of improvements that can be made to the existing code-base, a few are mentioned here:

1. Provide support for *KServer* as a stealth process, such that it does not appear in Windows task-list.
2. Provide support for *KServer* as a Windows service.
3. Provide support for strong encryption (AES-192, AES-256, 3DES ...) for *KServer/KClient* data-transfer.
4. Provide support for other forms of data (image/audio/video ...) read/write to clipboard.
5. Provide support for interconnect between numerous *KServer(s)* to form a stealth network.
6. Provide support for remote safe-wipe clipboard buffer.
7. Provide a web-based prototype.

...

# Notes

1. This experimental tool only demonstrates remote read/write of text data.
2. There are inherent bugs in this tool.
3. This tool is for demonstrative purpose only and comes as is (with no warranty whatsoever) and in no circumstances author is liable for incorrect or malicious use of the tool on user's part.