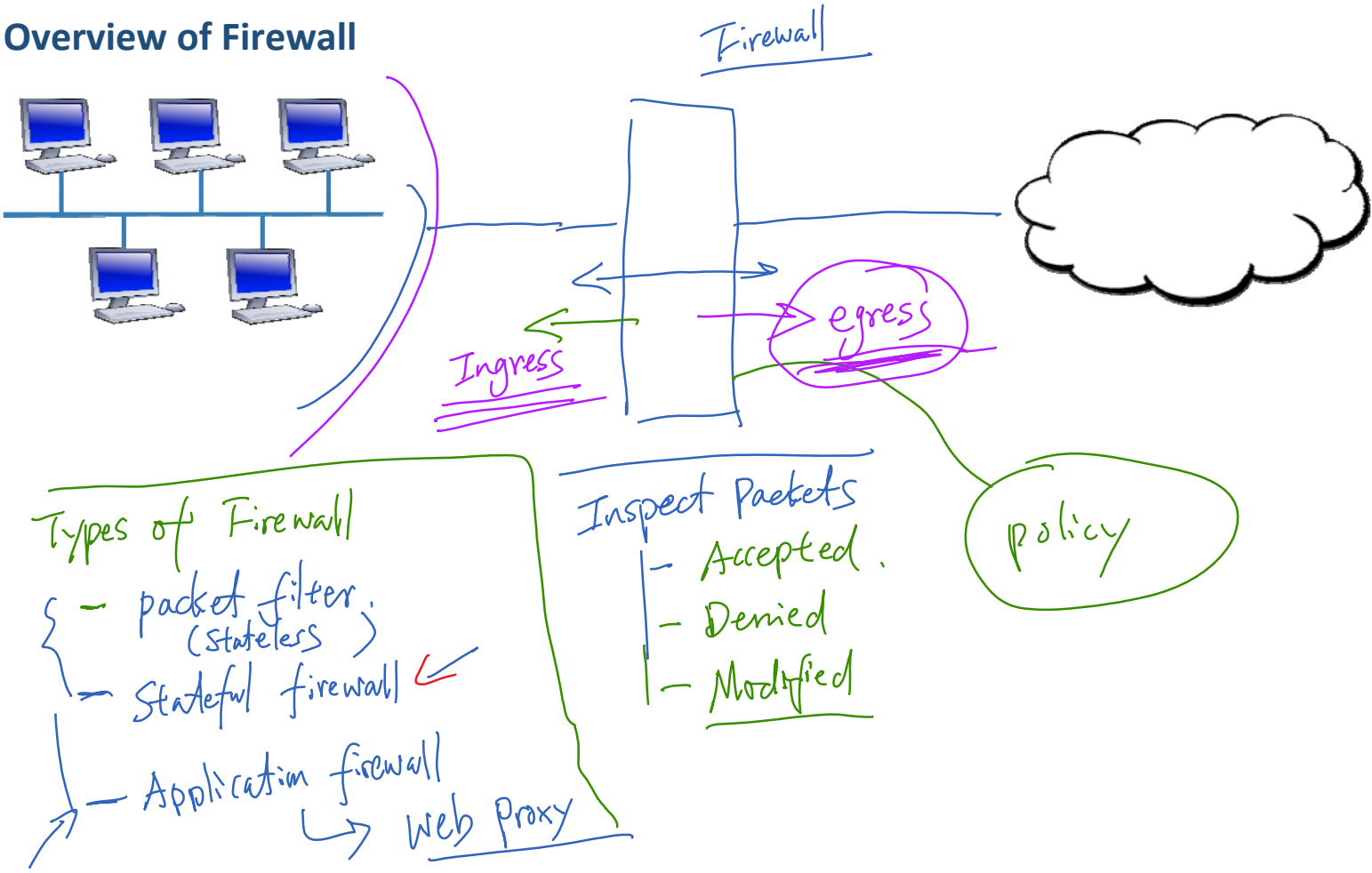
Overview of How Firewall Works

Overview of Firewall

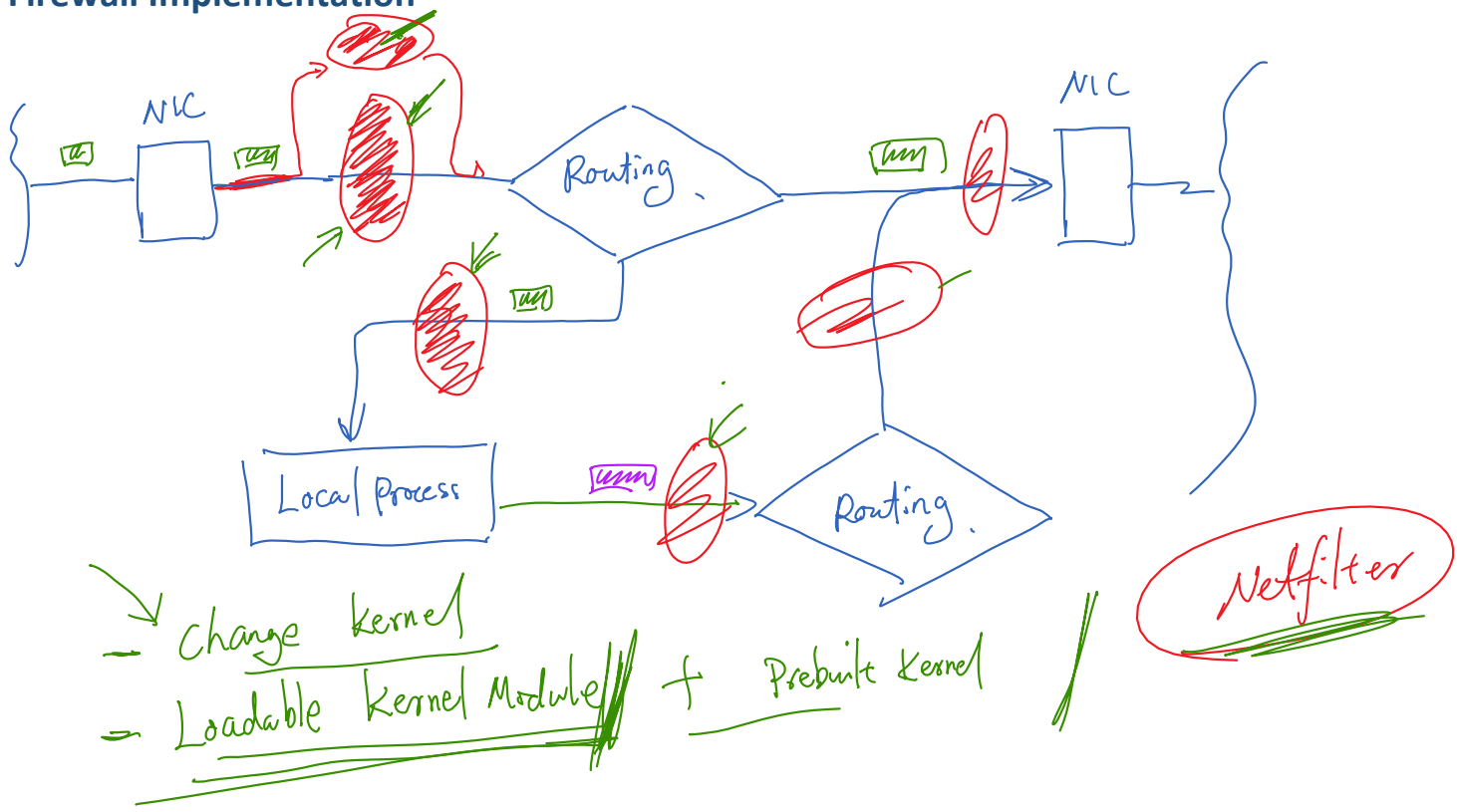


SYRACUSE UNIVERSITY
ENGINEERING & COMPUTER SCIENCE

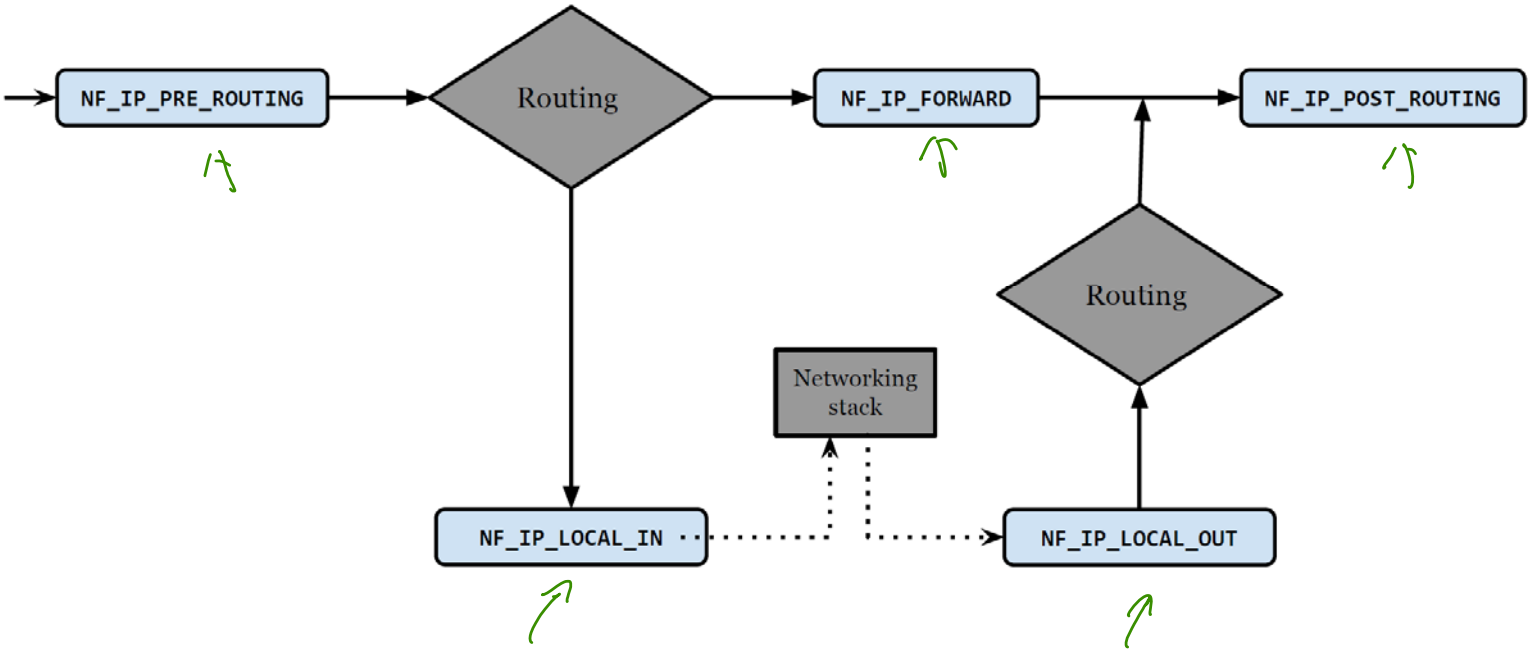
Linux Firewall Implementation

SYRACUSE UNIVERSITY
ENGINEERING & COMPUTER SCIENCE

Linux Firewall Implementation



Netfilter Hooks



Netfilter: Implement a Simple Firewall (`minifirewall`)

❖ Hooking filter code to one of the netfilter hooks

```c
static struct nf_hook_ops telnetFilterHook;

int setUpFilter(void) {
    printk(KERN_INFO "Registering a Telnet filter.\n");
    telnetFilterHook.hook = telnetFilter;
    telnetFilterHook.hooknum = NF_INET_POST_ROUTING;
    telnetFilterHook.pf = PF_INET;
    telnetFilterHook.priority = NF_IP_PRI_FIRST;
    nf_register_hook(&telnetFilterHook);
    return 0;
}

void removeFilter(void) {
    printk(KERN_INFO "Telnet filter is being removed.\n");
    nf_unregister_hook(&telnetFilterHook);
}

module_init(setUpFilter);
module_exit(removeFilter);
```

❖ Implementation of the filter

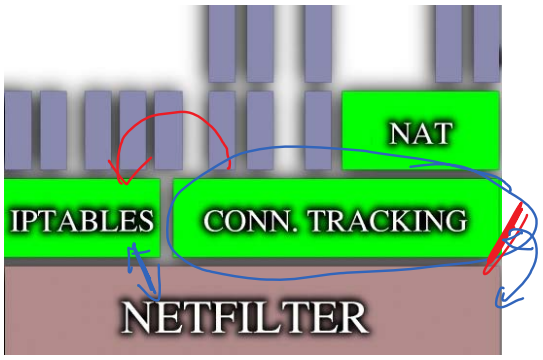
```c
unsigned int telnetFilter(unsigned int hooknum, struct sk_buff *skb,
        const struct net_device *in, const struct net_device *out,
        int (*okfn)(struct sk_buff *)) {
    struct iphdr *iph;
    struct tcphdr *tcph;

    iph = ip_hdr(skb);
    tcph = (void *)iph+iph->ihl*4;

    if (iph->protocol == IPPROTO_TCP && tcph->dest == htons(23)) {
        printk(KERN_INFO "Dropping telnet packet to %d.%d.%d.%d\n",
                ((unsigned char *)&iph->daddr)[0],
                ((unsigned char *)&iph->daddr)[1],
                ((unsigned char *)&iph->daddr)[2],
                ((unsigned char *)&iph->daddr)[3]);
        return NF_DROP;
    } else {
        return NF_ACCEPT;
    }
}
```

SYRACUSE UNIVERSITY
ENGINEERING & COMPUTER SCIENCE

Iptables and UFW

SYRACUSE UNIVERSITY
ENGINEERING & COMPUTER SCIENCE

Iptables and Uncomplicated Firewall (UFW)



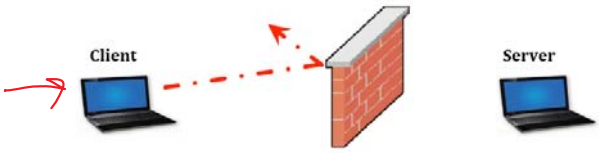
iptables — { firewall impl.
user-level prog.

→ several tables
- filter : filtering
- nat : modification : src/dest addr.
- ---
- mangle : modify contents

UFW: Using UFW to Set up Firewall Rules

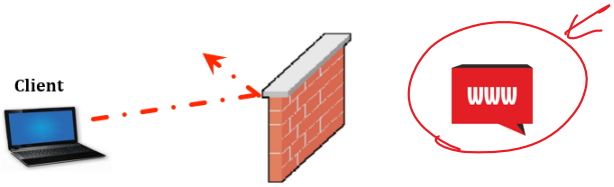
```
ufw <action> <direction> <service>
ufw (allow | deny) (in | out) from (src) to (dest) port (portNo)
```

#1 *"Prevent client machine from telnetting to any external machine"*




```
sudo ufw deny out from Client_IP to any port 23
```

#2 *"Prevent client machine from accessing a website"*

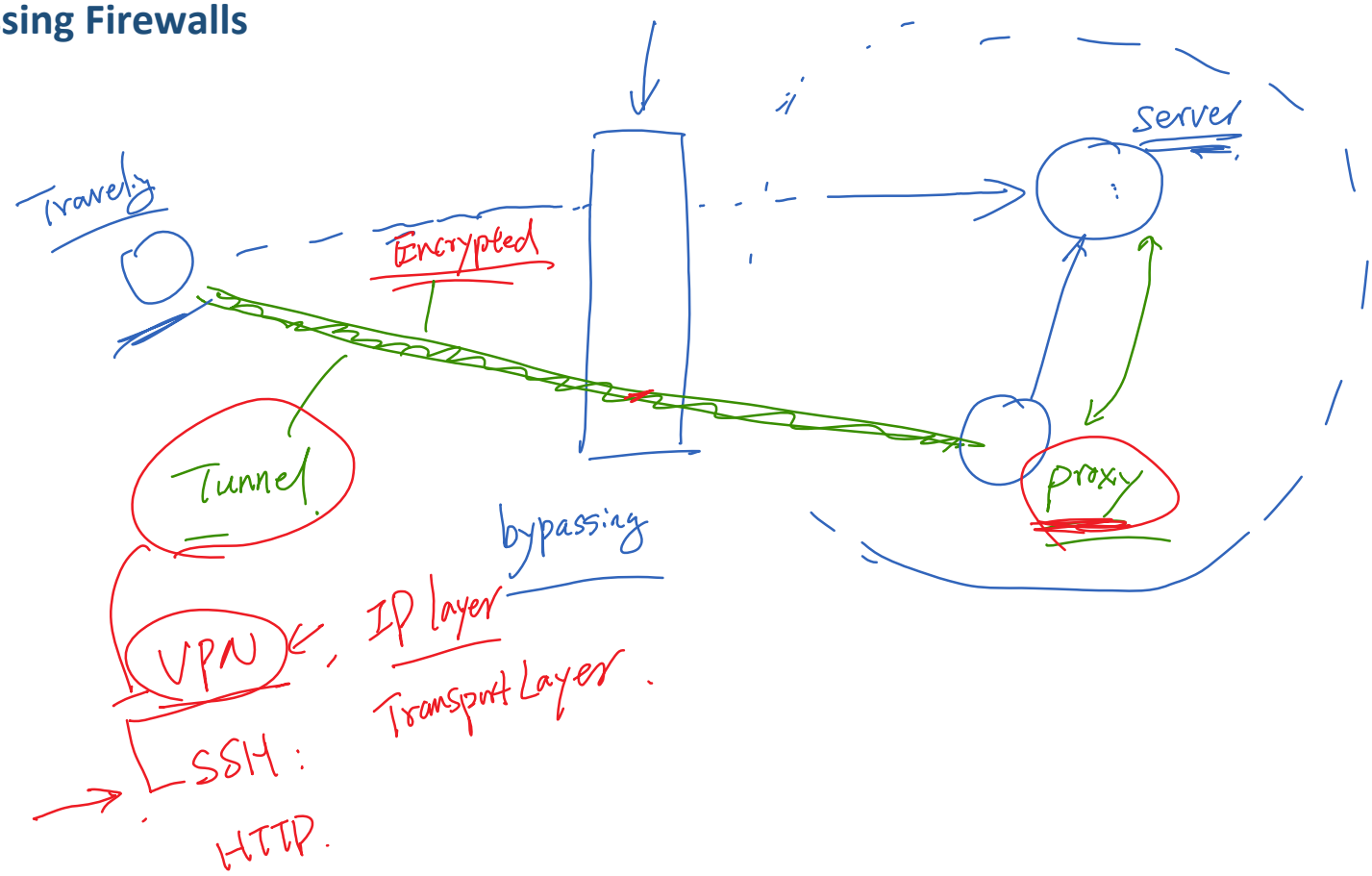


SYRACUSE UNIVERSITY
ENGINEERING & COMPUTER SCIENCE

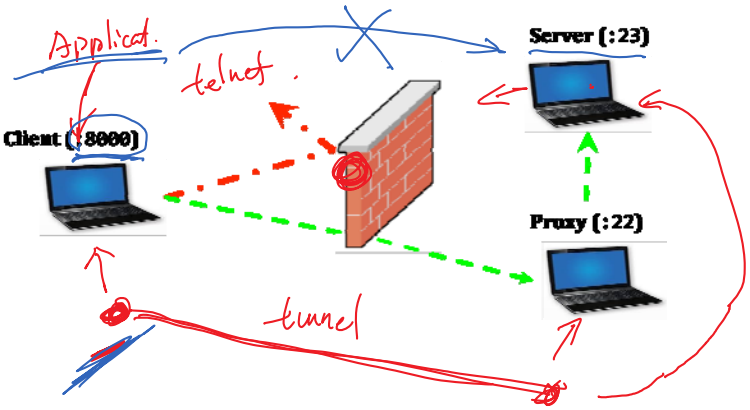
Bypassing Firewall Using SSH Tunnel

SYRACUSE UNIVERSITY
ENGINEERING & COMPUTER SCIENCE

Bypassing Firewalls

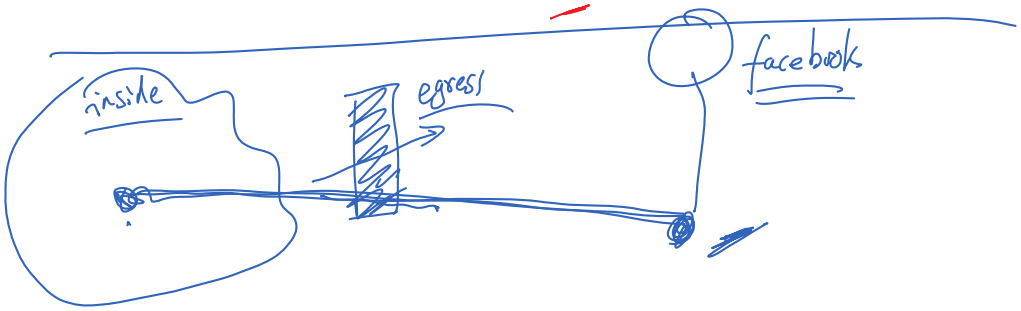


SSH Tunnel: Static Port Forwarding

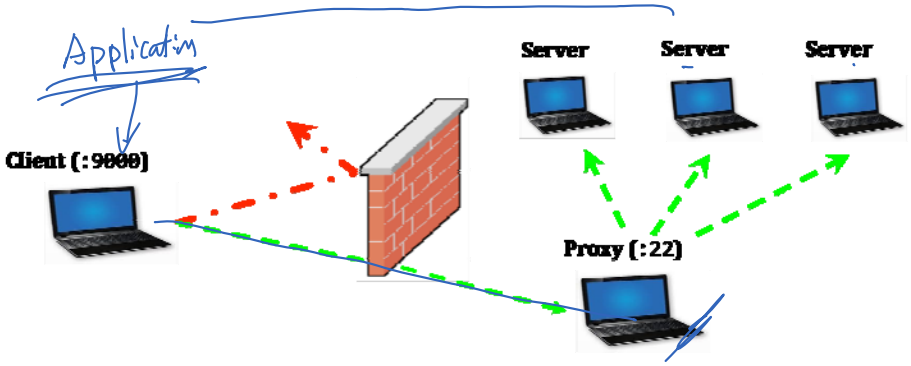


- ssh -L 8000 : server : 23 proxy

- telnet localhost 8000



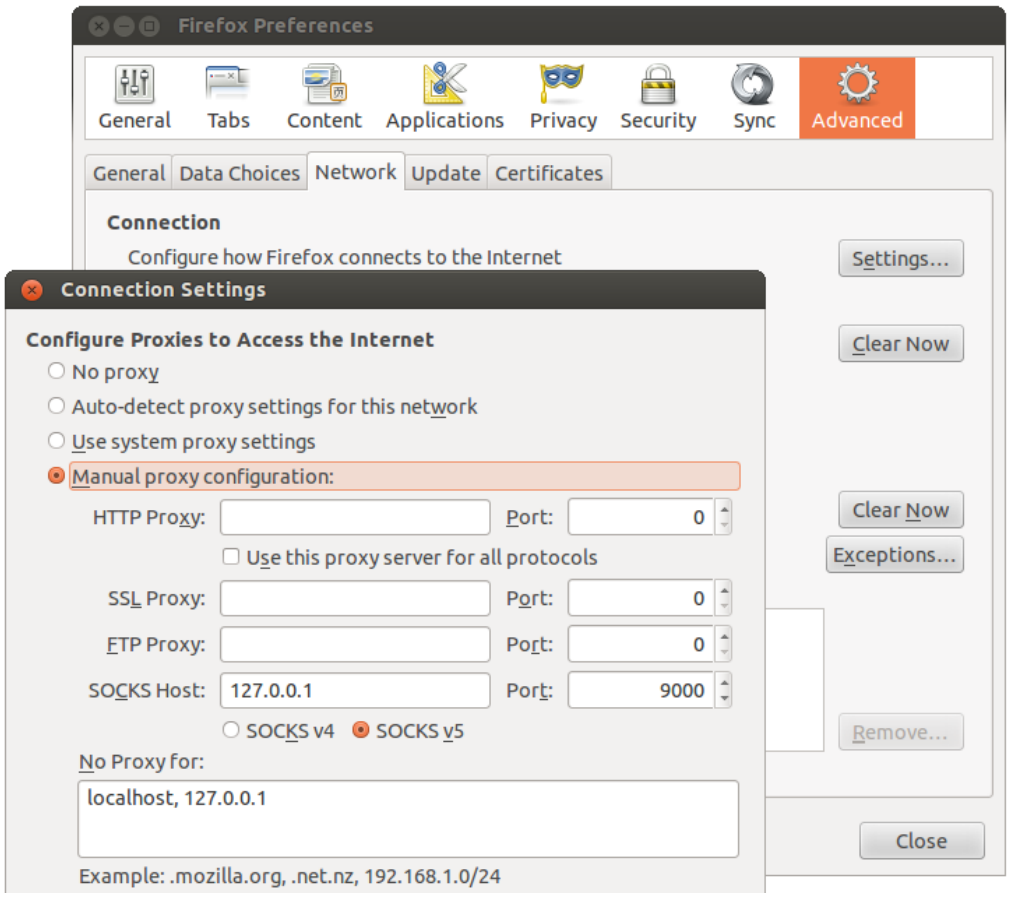
SSH Tunnel: Dynamic Port Forwarding



8000 : server : 23
target

- ssh -D 9000 -C proxy

Configuring Browser to use Dynamic Port Forwarding

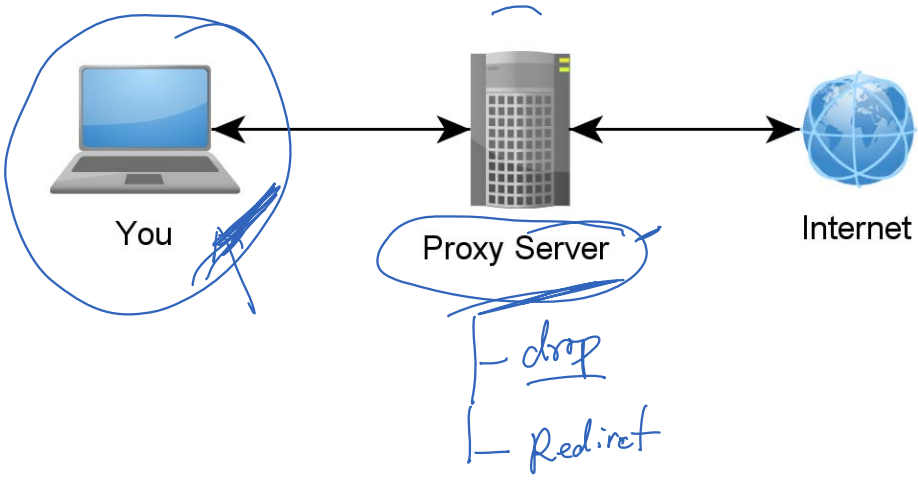


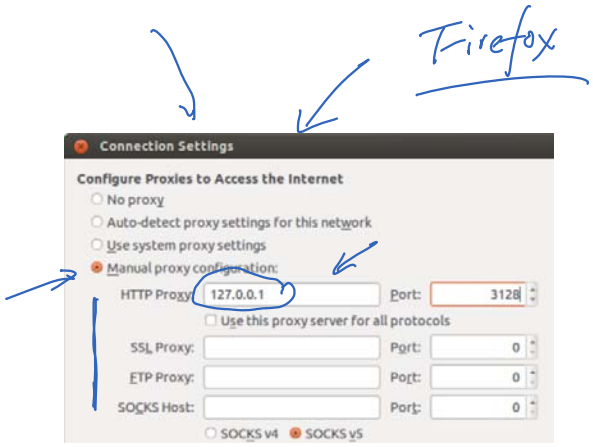
SYRACUSE UNIVERSITY
ENGINEERING & COMPUTER SCIENCE

Web Proxy

SYRACUSE UNIVERSITY
ENGINEERING
& COMPUTER
SCIENCE

Web Proxy: Application Firewall

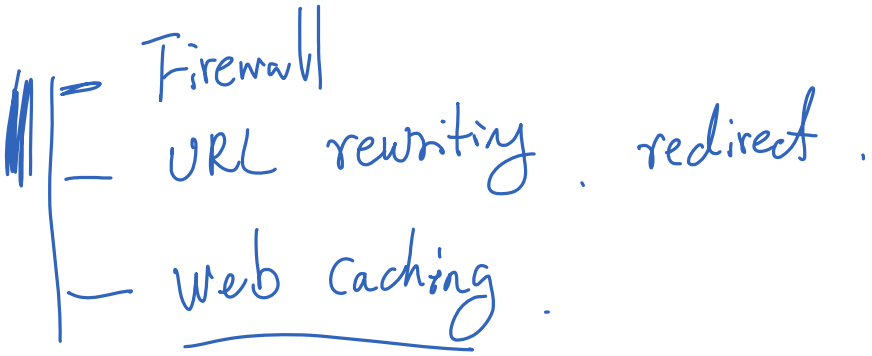




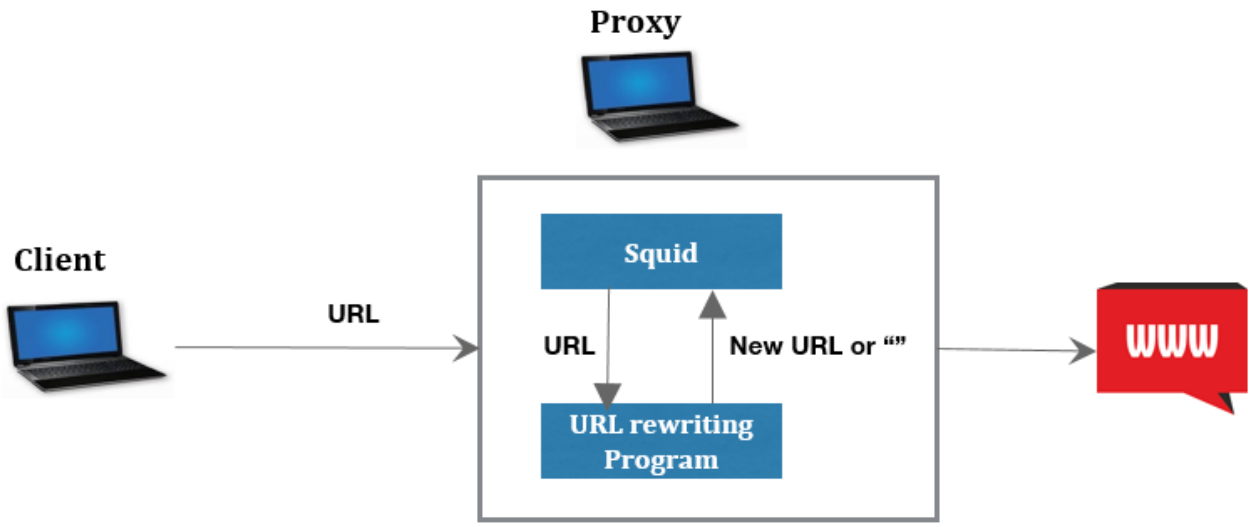
Web Proxy: Squid

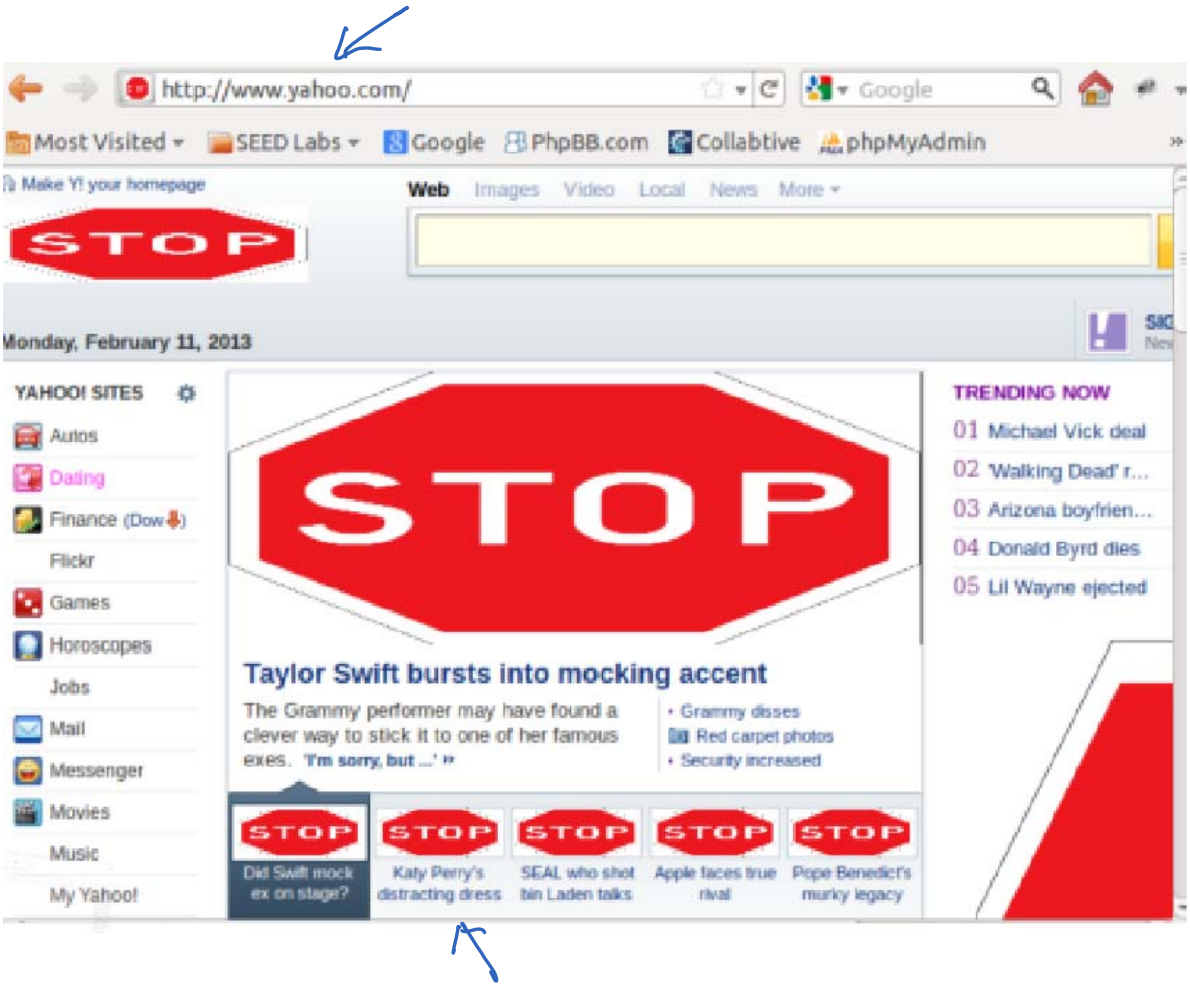
What is Squid?

Squid is a fully-featured HTTP/1.0 proxy which is almost (but not quite - we're getting there!) a fully-featured HTTP/1.1 proxy. Squid offers a rich access control, authorization and logging environment to develop web proxy and content serving applications. Squid offers a rich set of traffic optimization options, most of which are enabled by default for simpler installation and high performance.



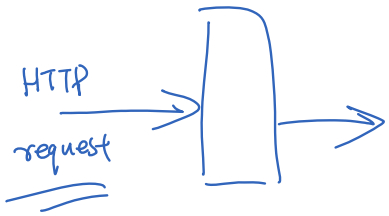
Squid: Redirect Traffic





Squid: URL Rewriting Code

```
#!/usr/bin/perl -w
use strict;
use warnings;
# Forces a flush after every write or print on the STDOUT
select STDOUT; $| = 1;
# Get the input line by line from the standard input.
# Each line contains an URL and some other information.
while (<>) {
    my @parts = split;
    my $url = $parts[0];
    # If you copy and paste this code from this PDF file,
    # the ~(tilde) character may not be copied correctly.
    # Remove it, and then type the character manually.
    if ($url =~ /\.(jpg|bmp|gif|jpeg)/) {

        # URL Rewriting
        print "http://mars.syr.edu/html/seed/stopsign.png\n";
    }
    else {
        # No Rewriting.
        print "\n";
    }
}
```



SYRACUSE UNIVERSITY
ENGINEERING & COMPUTER SCIENCE

Summary

SYRACUSE UNIVERSITY
ENGINEERING & COMPUTER SCIENCE

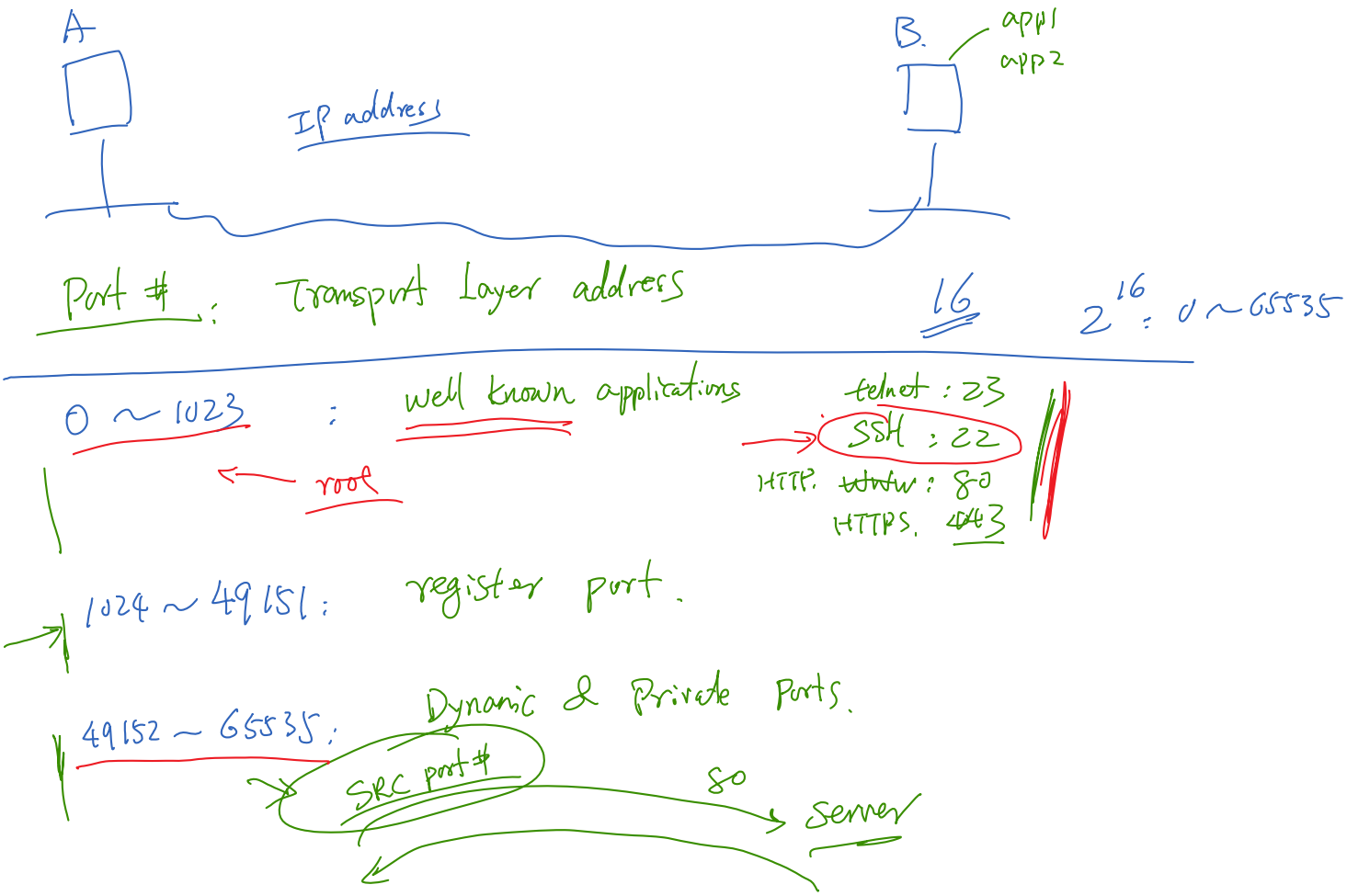
Summary

- Concepts of firewall
- Firewall implementation (simple packet filter)
- Netfilter and iptables
- Evading firewall using SSH tunnel
- Web proxy firewall

SYRACUSE UNIVERSITY
ENGINEERING & COMPUTER SCIENCE

UDP Overview

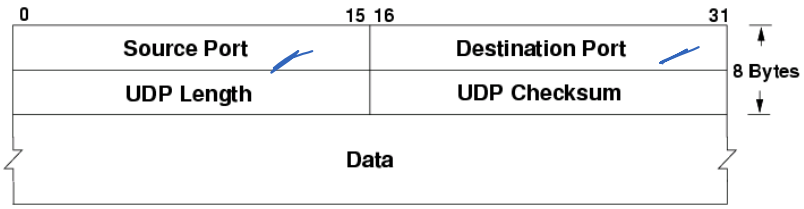
Transport Layer and Port Numbers



SYRACUSE UNIVERSITY
ENGINEERING & COMPUTER SCIENCE

UDP Header and Protocol

UDP Header and Protocol



TCP UDP.

UDP:
light-weight

TCP

IP: best effort delivery
packet loss
1 2 3
3 1

UDP Client/Server Programs

❖ UDP client

a. Create socket:

```
sockfd = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP)
```

b. Send data:

```
sendto(sockfd, buffer, …, (struct sockaddr *)&servaddr …)
```

c. Receive data:

```
recvfrom(sockfd, rec_buffer, …, &from_addr, ...);
```

❖ UDP server

a. Create socket:

```
sockfd = socket(AF_INET, SOCK_DGRAM,  IPPROTO_UDP)
```

b. Bind the socket to a port:

```
bind(sockfd, &si_me, ...)
```

c. Receive data:

```
recvfrom(sockfd, rec_buffer, …, &from_addr, ...)
```

SYRACUSE UNIVERSITY
ENGINEERING & COMPUTER SCIENCE

UDP Applications

UDP Applications

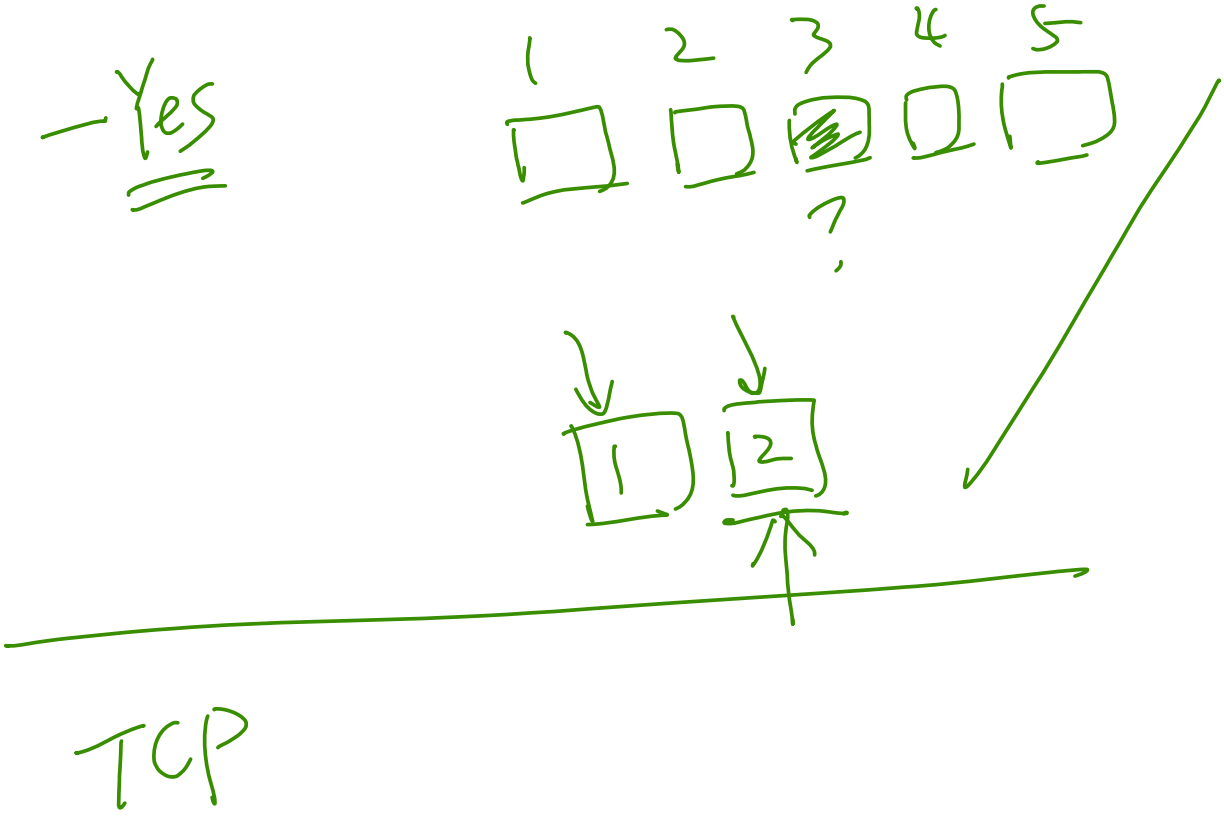
- DNS Protocol
- Video/Audio Streaming
- Real-Time Applications

– { real time : UDP
  video : TCP.

UDP
- Light weight
- packet loss

Question

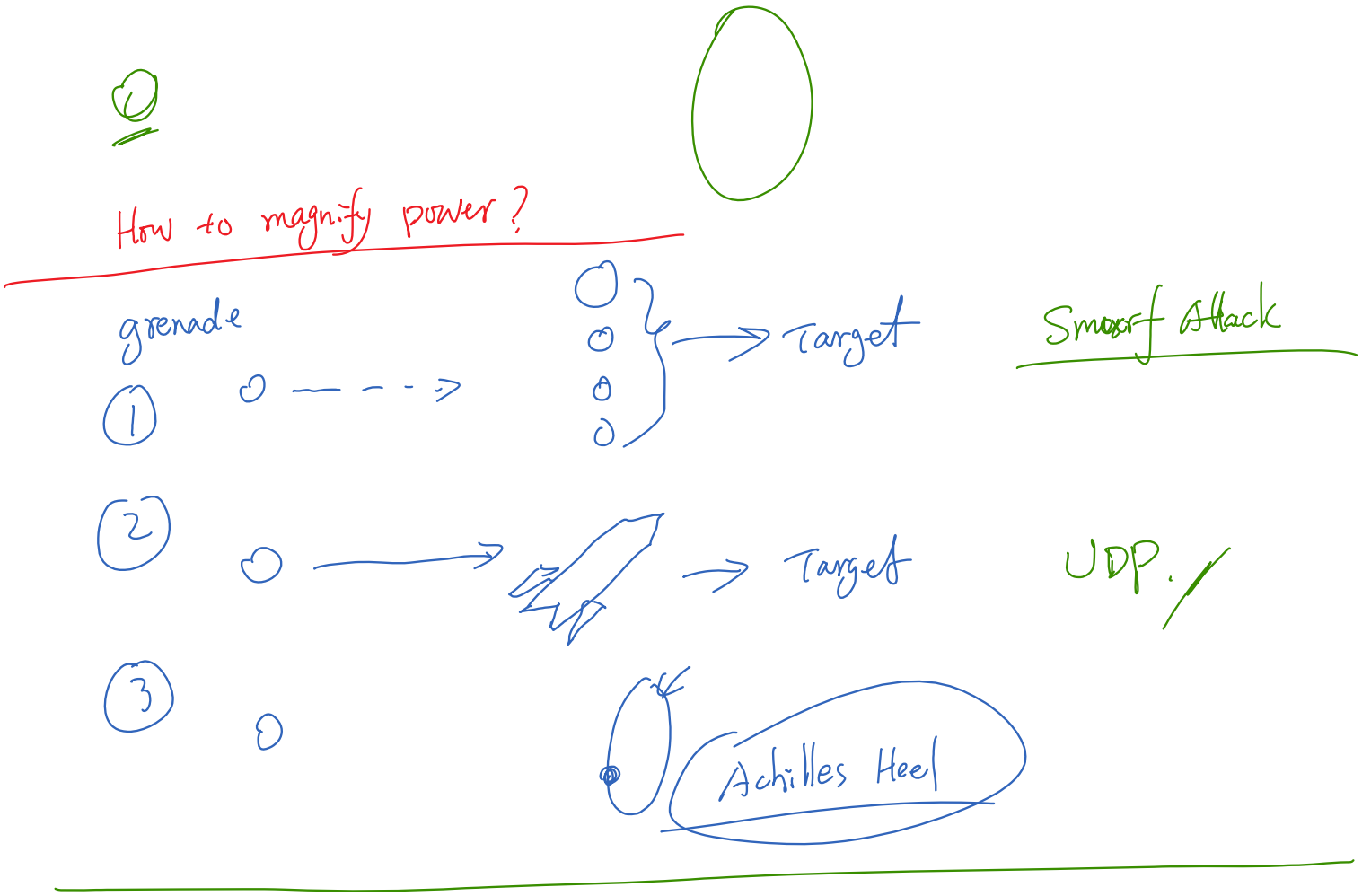
UDP does not preserve order and does not handle packet loss. If an application does care about packet loss and order, can it still use UDP? Please explain.

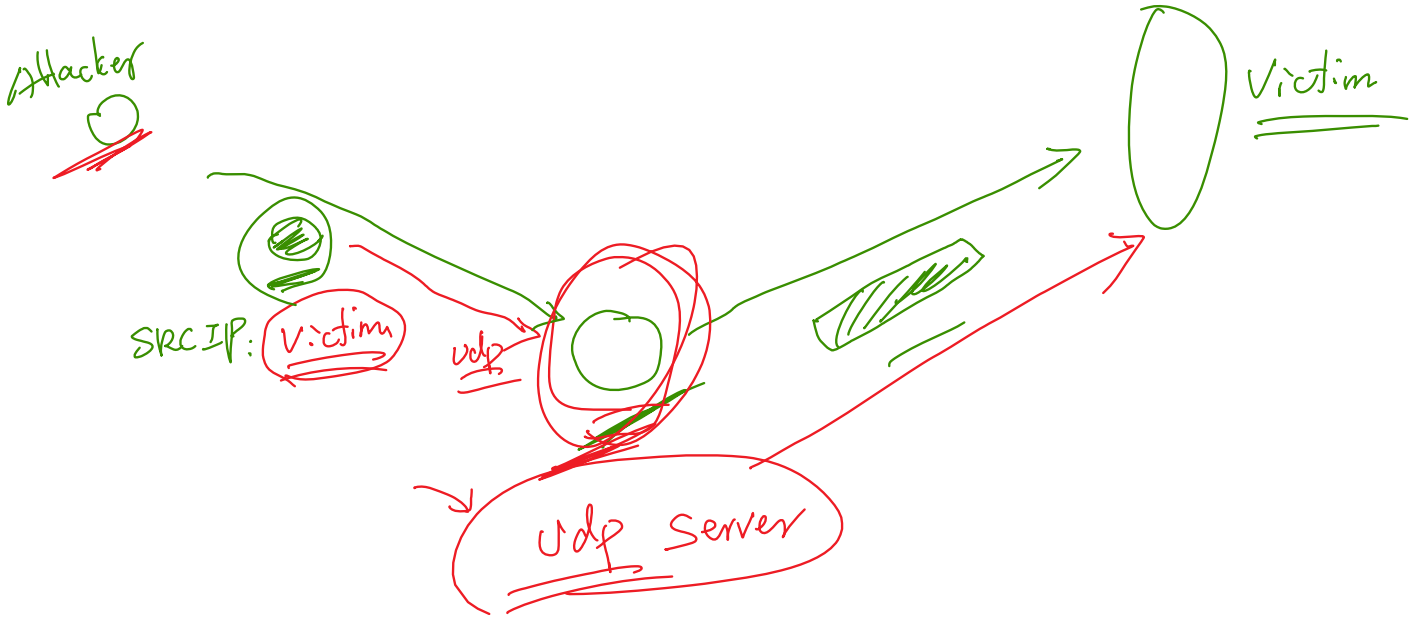


SYRACUSE UNIVERSITY
ENGINEERING & COMPUTER SCIENCE

Attacks on UDP

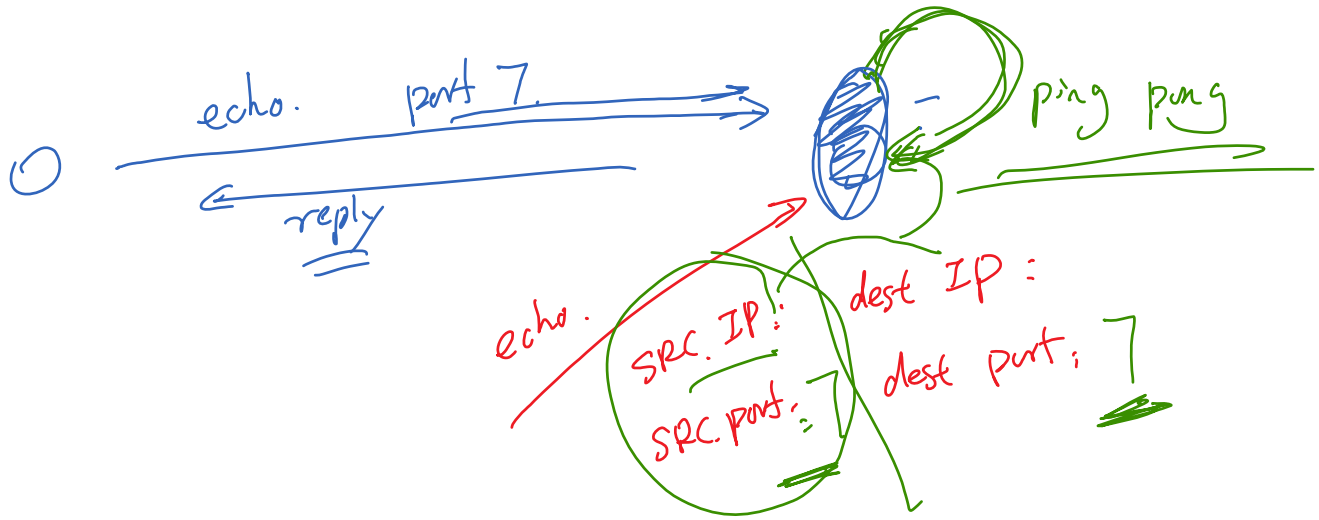
Causing Great Damage Using a Grenade





Turn a Grenade Into a Missile

Target Achilles Heel: UDP Ping-Pong



SYRACUSE UNIVERSITY
ENGINEERING & COMPUTER SCIENCE

Summary

SYRACUSE UNIVERSITY
ENGINEERING & COMPUTER SCIENCE

Summary

- Transport layer
- Port number
- UDP protocol and header
- UDP applications
- Attacking strategy: magnify power
- Attack on or using UDP

SYRACUSE UNIVERSITY
ENGINEERING & COMPUTER SCIENCE