# Virtual Private Networks

Jonathan Reed
jdreed@mit.edu
MIT IS&T VPN Release Team

## Overview

- Basic Networking Terms
- General Concepts
- How the VPN works
- Why it's useful
- What to watch out for
- Q&A

## Networking 101

- OSI seven layer model
  1. Physical (copper/fiber)
  2. Data Link (MAC/network card)
  3. Network (routing)
  4. Transport (protocol)
  5. Session (connections)
  6. Presentation (encoding/platform independence)
  7. Application (program itself)
  8. Money
  9. Politics
  10. Religion

## Layer 3 protocols

- IP - Internet Protocol (IPv4)
  - provides addressing for packets
  - provides fragmentation and reassembly of packets that exceed MTU (maximum transmission unit - how big a packet can be - typically 1500 bytes)
- ICMP - Internet Control Message Protocol
  - allows for reporting errors, congestion, timeouts, no such host, etc - used by “ping”

## Layer 4 protocols

- TCP - Transmission Control Protocol
  - allows for streams, reliability, full-duplex
  - packets guaranteed to be received in order sent
- UDP - User Datagram Protocol
  - connectionless
  - send a packet and hope for the best
  - packets can be received out of order (zephyr, streaming media)

## Packets

- IP packets
  - time to live
  - protocol
  - source address
  - destination address
  - checksum
  - data

## Packets, cont

- UDP: all of IP packet info, plus:
  - source and dest. ports
  - checksum
- TCP: all of IP packet info, plus:
  - source and dest. ports
  - sequence and acknowledgement numbers
  - checksum
  - window (amount of data that can be sent before the receipt is acknowledged)

## Services

- Services listen on a port (TCP or UDP)
- When a packet arrives, kernel checks port number
  - if a service is listening on that port (ie: sshd, apache), decode packet and send to service
  - if not, return ICMP “port unreachable” message

## Port Blocking/Firewalls

- kernel can check for user-specified info (port num, source addr, dest. addr, etc)
  - return ICMP unreachable (appears as if no service on that port)
  - drop packet on floor (wait for remote machine to timeout)
  - return other ICMP message (protocol not allowed, host doesn’t exist, prohibited, etc)

## NAT

- static NAT (what most people use)
  - 1 “real” IP address
  - 1 or more private IP addresses
- security only through obscurity
- can also act as firewall
- breaks any protocols that rely on knowing the source IP address (krb4, FTP, etc)

## Tunneling

- the idea that some element of data can be encapsulated in a larger element of data
- in simplest form, IP-IP tunnels
  - one IP packet wrapped in another IP packet
  - only works for IP networks (not Netware (IPX), AppleTalk, etc)
  - offers no more and no less security than a regular network

## Tunneling Abstraction

Regular Packet

To: www.google.com
From: mymachine.mit.edu
Port: 80
Data: “school” “science” +Cambridge

Tunneled Packet

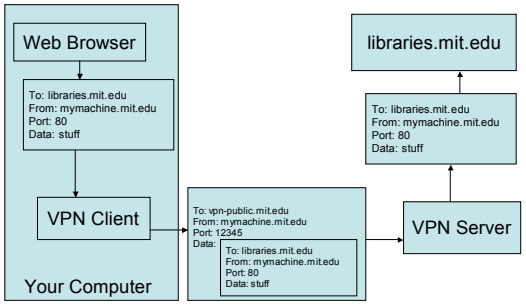
To: vpn-public.mit.edu
From: mymachine.mit.edu
Port: 12345
Data:

To: www.google.com
From: mymachine.mit.edu
Port: 80
Data: “school” “science” +Cambridge

## Layer 2 VPNs (L2TP, L2F, PPTP)

- Both based on the idea that layer 2 protocols can be encapsulated in IP
- Basically, extends PPP over the internet
- uses existing PPP authentication (which has vulnerabilities)
- PPTP encryption is also cryptographically weak (RC4)
- Requires routers to understand protocols

## VPN Abstraction




## Layer 3 VPN: IPSec

- IP Security: suite of protocols for securing IP packets
- Protect Data (ESP)
- Key exchange protocols (IKE)
- Authentication of packets (AH)
- can provide either authentication or authentication + encryption (most common use)
- backwards compatible with existing networks

## Negotiation

- authenticate nodes to each other
  - pre-shared keys
  - X.509 Certificates with RSA signature
- set up tunnel
  - establish “security associations”
  - assign client IP address for VPN
- renegotiate (“re-key”) periodically to ensure security

## Cisco’s Implementation

- pre-shared keys
  - In this case, the “group name”, and “group key” (stored in MITnet-VPN.pcf file)
- XAUTH - e**x**ternal **auth**entication
  - use other methods, such as RADIUS, to authenticate users
- Determined to be insecure, Cisco plans new version
  - man-in-the-middle attack

## Security

- Layer 3 and above security **between client and VPN server only**
- Security between VPN server and eventual dest. is user’s responsibility
- Once packet leaves server in W92, it’s no different than any packet from any other machine on MITnet

## How it bypasses port blocking

- The actual port numbers are encrypted, so firewalls don't see them
- Appears to just be one connection to one machine (the VPN server) on one specific port (or set of several ports)
- ISPs can disallow VPN connections by blocking those ports

## How it solves krb4 issues

- Assigns your machine a new IP address
- krb4 (and FTP, and other things) use that IP address
- local machine temporarily "forgets" about its other IP addresses (ie: address assigned by Comcast)

## How is it different from HTTPS or SSH?

- Those are layer 7 protocols.
  - Anyone can view those packets and tell what machine you're talking to and what the port numbers are, and what the protocol is, it's just the data that's encrypted
- VPN encrypts the entire packet
  - attackers monitoring the VPN stream can't tell what machine you're talking to or what protocol you're using

## When to use it?

- work around port-blocking (ie: Verizon SMTP)
- solve the krb4 NAT problem
- assign you a net-18 IP address (restricted library resources, etc)

## When not to use it

- on-campus
  - will hide you from those on your subnet, but nothing else
- to hide your websurfing
  - once the packet leaves W92, it's clear text
- to send passwords/credit card numbers in e-mail
  - once the packet leaves W92, it's clear text
- example: FileMaker databases

## What goes wrong

- krb4 tickets already obtained will no longer work
- established zephyr/discuss/kerberized IMAP sessions fail
  - Quit all applications before connecting
- your MTU changes (PPPoE/DSL customers might care)

## Known Issues

- You lose advanced IP/routing control
  - Cisco clients only - design choice
- various OS issues
  - VPN has to insert itself between kernel core and networking drivers
  - libpcap gets confused on Linux
  - OS X's KEM/configd doesn't work, if you use the Kicker, it won't notice (correct behavior)
  - MTU silently reset on OS X even when VPN not running

## Cisco VPN Client

- Available from http://web.mit.edu/software/
- Product Front Door: http://itinfo.mit.edu/product?name=vpn
- Supported on: Windows, Mac, Linux (Red Hat Enterprise)
- Linux client will work on other distributions (4.6 is LSB/FHS compliant)

## Open Source VPN Client

- http://www.unix-ag.uni-kl.de/~massar/vpnc/
- Still in testing, but functional on Linux (x86, ppc, arm), *BSD (x86), Solaris
- Runs entirely in "userland"
  - no kernel module required
  - uses native OS IPSec support
- Completely unsupported by IS&T

## ‘Native’ VPN clients

- XP and OS X
- will not work with MIT VPN server
- L2TP over IPSec or PPTP client only

## Wrap-Up

- Questions?
  - vpn-release@mit.edu
- More info:
  - http://vpn.shmoo.com
  - http://www.vpnc.org/ietf-ipsec/
  - Virtual Private Networks, 2nd Ed. (O’Reilly) (somewhat dated, but useful)