Protocols….

Motivation

• We’ve seen a number of “demonstrating” protocols
  – IP
  – TCP
  – UDP
  – HTTP
  – RPC
• Want to now take a quick look at a number of other “lesser
well known” protocols
  – What’s common
  – What’s new
• Tunneling
• Translating
• Booting

Sticks. Shaking.
Tunneling

- Encapsulate one protocol in another
  - Make IP look like IPV6, or secure IP, or..

- May rely on PROXY architecture
  - A network agent that translates from the tunneled protocol to the tunneling protocol.

- Permits broad, but gradual, deployment of new services
  - Build tomorrow’s services on today’s.
  - Anticipates that a given service may one day be “built into” the network.

Example: IPV6 Tunnel
Example: VPN Tunnel

Mobile IP

• Problem: How to enable a node to move from one IP subnet to another.
SSH Tunnel (Port Forwarding)

- Transport packets destined for one endpoint to another
  - Eg, “please revector all packets sent to my ip address, port 1029, to 128.95.1.4:3000”

Translation

- Mechanically “mutate” the packets on ingress/egress
- Requires some sort of real or apparent proxy
Network Address Translation
RFC 1631

• Problem: How to give everybody their own network
  – Subnets (Class A, B, C, D…)
  – IPV6
  – NATted Networks
• Idea: Burn 1 IP address and not a whole subnet
  – Assign network ingress/egress a routable network address
  – Assign hosts a “non routable” network address
  – Translate IP packets as they come and go

Challenges with NAT

• A non-routable address is not very routable

VOIP, Web servers, etc require special “hacks”
10., 192., 172., 169!!
Protocol “Leakage”

- Translators only work well if they translate everything that needs to be translated.
- What if the data portion of the conversation “reveals” something about the part that is being translated
  - Active vs. Passive mode FTP
- Forces us to make smarter and smarter (“statefuller”) NATters

Bootstrapping

- Problem: How to assign a protocol “identifier” to a dumb host?
  - IP address, host name, etc.
- Assume
  - Some unique ID (enet address)
  - Broadcast
DHCP

1. What is DHCP?

DHCP stands for "Dynamic Host Configuration Protocol".

2. What is DHCP’s purpose?

DHCP’s purpose is to enable individual computers on an IP network to extract their configurations from a server (the ‘DHCP server’) or servers, in particular, servers that have no exact information about the individual computers until they request the information. The overall purpose of this is to reduce the work necessary to administer a large IP network. The most significant piece of information distributed in this manner is the IP address.

3. Who Created It? How Was It Created?

DHCP was created by the Dynamic Host Configuration Working Group of the Internet Engineering Task Force (IETF, a volunteer organization which defines protocols for use on the Internet). As such, its definition is recorded in an Internet RFC and the Internet Activity Board (IAB) is assigning its status as to Internet Standardization. As of this writing (June 1999), DHCP is an Internet Draft Standard Protocol and is Elective. BOOTP is an Internet Draft Standard Protocol and is Recommended. For more information on Internet standardization, see RFC2309 (May 1998).

4. How is it different than BOOTP or RARP?

DHCP is based on BOOTP and maintains some backward compatibility. The main difference is that BOOTP was designed for manual pre-configuration of the host information in a server database, while DHCP allows for dynamic allocation of network addresses and configurations to newly attached hosts. Additionally, DHCP allows for recovery and reallocation of network addresses through a leasign mechanism.

DHCP In Action

![DHCP Diagram]

1. デバイスは起動時にDHCPサーバーに対して問い合わせる
2. DHCPサーバーはデバイスの情報を返す
3. (その他のネットワークマネージメント情報もDHCPから取得）

DHCPサーバー

IPアドレス
192.168.0.1
サブネットマスク
255.255.255.0
defalutゲートウェイ
192.168.0.30

デバイス

IPアドレス
192.168.0.1
サブネットマスク
255.255.255.0
gateウェイ
192.168.0.30

IPアドレス
192.168.0.2
サブネットマスク
255.255.255.0
gateウェイ
192.168.0.30

IPアドレス
192.168.0.3
サブネットマスク
255.255.255.0
gateウェイ
192.168.0.30
DHCP+NAT

- One gives you an address that’s potentially always changing
- Another conceals your internal addressing structure from the outside world
- Where is this great?
- Where is this lousy?

Summary

- Protocols we already know lend themselves well to new protocols
- New protocols may directly leverage existing ones
- Or they may support them
- Interactions which often “feel” like bugs are often indicators of a more general property.