THE UNIVERSITY OF
MELBOURNE

COMP30023 – Computer Systems
2018 – Semester 1 - Week 7 – Lecture 1

Internet (Network) Layer –Subnets, NAT, & Fragmentation

Dr Lachlan Andrew

Recap

- Forwarding vs routing
- Flooding vs shortest path
- Bellman’s optimality principle
- Dijkstra’s algorithm to find the shortest path
  - Closed (decided) and open (undecided) sets
- BGP

Summary

- Network Address Translation (NAT)
- Fragmentation
  - IPv4
  - IPv6
- Subnets

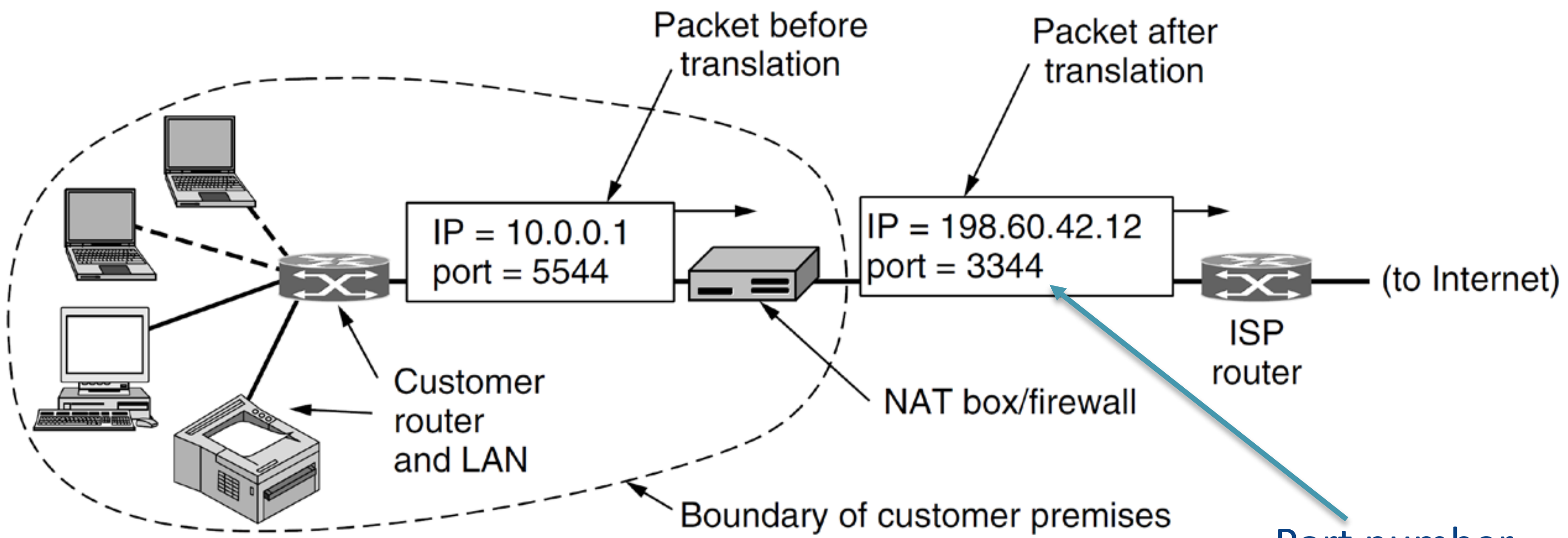
IPv4 address scarcity

- As IP addresses became scarce, methods for handling many more clients were developed
- Whilst IPv6 would solve the problem a stop gap was needed
- Private addresses
  - Many hosts in a company only need internal access
  - “Private” subnets 192.168.0.0/16, 172.16.0.0/12, 10.0.0/8
  - Can be reused: unique address within organisation, not globally
- Intention: “Application layer proxies” to access outside services
  - Instead…

Network Address Translation (NAT)

- Each customer/home is assigned one public IP address
  - Businesses might be issued a few
- Internally hosts/interfaces are issued Private IP addresses
  - Recall 10.0.0.0 - 10.255.255.255/8 (as an example)
- Internal IP addresses are used for communicating amongst hosts in the Local Area Network (LAN)
- They must never be used on the public internet
- When a packet is heading out of the network (to the ISP) the internal address is translated to the public IP address

Network Address Translation (NAT)



Port number also changes → NAPT

Network Address Translation (NAT)

- How NAT works:
  - Assumes TCP/UDP (some exceptions), in particular the locations of source and destination port fields
  - NAT box replaces TCP source address (10.x.y.z) with public IP address
  - TCP source port replaced with index of entry in NAT translation table
    - One of 65,536 entries (16 bits – same as TCP port field)
    - Each entry contains original IP address (private IP) and original source port number
  - IP and TCP checksums are recalculated
  - When a packet arrives from the internet at the NAT box it looks up the destination port from the TCP header in the translation table
    - Retrieves original source port and source IP address, updates headers and checksums and sends to the internal host

Network Address Translation (NAT)

- Criticisms of NAT
  - Breaks end-to-end connectivity: an interface in the private network can only receive packets once it has sent packets out and created a mapping (some exceptions)
  - Violates layer model by assuming nature of payload contents – initially only worked for TCP and UDP
  - Violates IP architectural model that states every interface on the internet has a unique IP address (millions of interfaces connecting to the internet have 10.0.0.1)
  - Changes internet from connectionless to pseudo-connection-oriented
    - NAT maintains connection state, if it crashes all connections are lost
  - Causes problems with FTP and other protocols that use multiple connections in a prescribed way
  - Limits number of outgoing connection

Network Address Translation (NAT)

- Despite criticisms, it is widely deployed, particularly in homes and small businesses
  - Carrier grade NAT: ISP only gives customers *private* addresses
- Significant security advantage
  - Since packets can only be received once an outgoing connection has been created, the internal network is greatly shielded from attacks from incoming unsolicited packets
  - NAT should not replace firewalls
- Likely to remain in use even after IPv6 is widely deployed and there is no longer a scarcity of IP addresses

Fragmentation

- Recall that IP packets have a maximum size of 65,535 - determined by the Total Length header field being 16 bits
- However, most network links cannot handle such large sizes
- All networks have a maximum size for packets, due to:
    - hardware
    - OS
    - protocols
    - standards compliance
    - desire to reduce transmissions due to errors
    - desire for efficiency in communication channel

Fragmentation

- Nature of layered protocol stack means lower layer potentially needs to be able to fragment larger packets
- More important: the most restrictive link on a packet's path may be on a link it is not connected to
  - Can't just pass info up the protocol stack in the sender

- Fragmentation (division of packets into fragments) allows network gateways to meet size constraints

Fragmentation

- Hosts want to transmit large packets, since it reduces workload for them
- Creates a problem when that packet transits other networks that may not support such a large packet size
- Common maximum sizes for different network technology
  - 1500 bytes for Ethernet (non-standard extension to 9000)
  - 2304 bytes for 802.11.
- For example, sending packets between two devices on the same WiFi network could use larger packets than sending between WiFi and Ethernet device

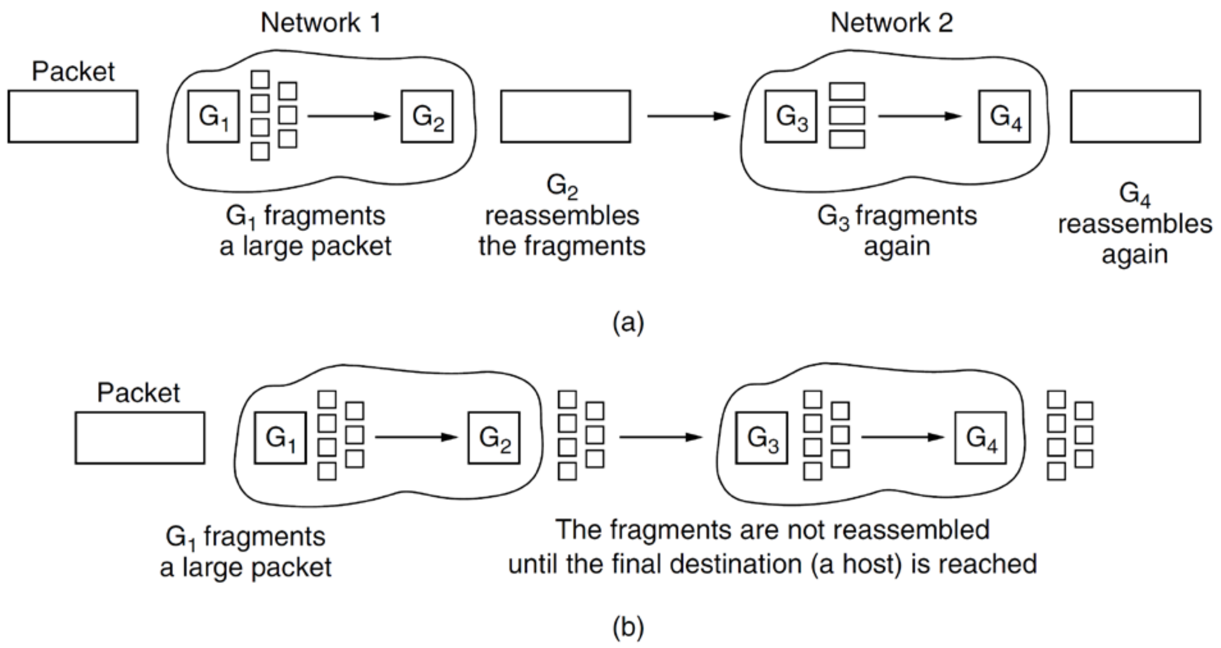
Fragmentation

- MTU – Maximum Transmission Unit
  - Maximum size for that network or protocol
- Path MTU
  - Maximum size for the path through the network
- Why not just set the Path MTU at the sender?
  - Connectionless network, with dynamic routing – both route and link MTU can change after the packet has been sent
- In keeping with design goals of TCP/IP (keep it simple) the easiest solution seemed to be to allow routers to break large packets into fragments to be sent individually along the network

Fragmentation

- Problem: breaking a large packet into smaller fragments is easy, putting them back together again is a much harder task
- Two approaches:
  - Transparent Fragmentation – reassembly is performed at next router; subsequent routers are unaware fragmentation has taken place
  - Nontransparent Fragmentation – reassembly is performed at the destination host

Fragmentation



Network 1
Network 2
Packet
G1
G2
G3
G4
G1 fragments a large packet
G2 reassembles the fragments
G3 fragments again
G4 reassembles again
(a)
Packet
G1
G2
G3
G4
G1 fragments a large packet
The fragments are not reassembled until the final destination (a host) is reached
(b)

a) Transparent

b) Nontransparent (used by IP)

Fragmentation and IP Headers

- Recall the following IP Headers
  - Identification – used to identify a packet
  - Flags (DF=Don’t Fragment and MF=More Fragments)
  - Fragment offset – offset in 8 byte blocks
    - 13 bits – max offset (2^13 – 1) * 8 = 65,528,
- If a packet is fragmented:
  - Identification stays the same for all fragments
  - MF=1 for all fragments, except the last
  - Fragment offset – appropriately set for each fragment
- Fragment offset allows the receiving host to reconstruct out-of-order fragments in a buffer – similar to TCP Segments

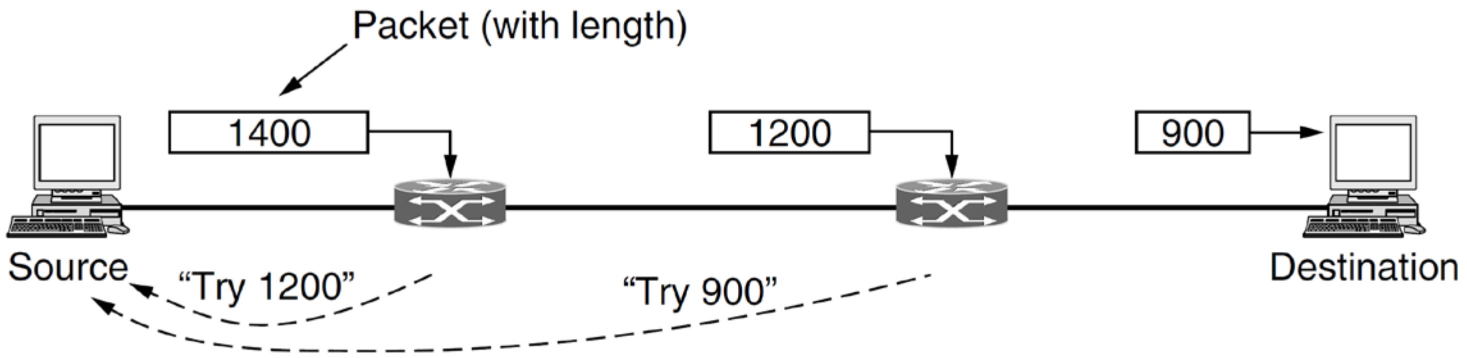
Fragmentation and IP Headers

- Fragment offset, and therefore fragmentation size, must be on an 8 byte boundary
  - Cannot send single byte fragments (except the last)
- If we have a payload of 1700 bytes. MTU=1500 bytes, ID=1:
  - 1st Fragment : ID = 1, DF=0, MF=1, FO=0
  - 2nd Fragment: ID = 1, DF=0, MF=0 , FO=185
    - (185*8=1480 + 20 byte header = 1500)

Fragmentation

- Simple approach, but some downsides:
  - Overhead from fragmentation (20 byte header for each fragment) is incurred from the point of fragmentation all the way to the host
  - If a single fragment is lost the entire packet has to be resent
  - Overhead on hosts in performing reassembly higher than expected
- Alternative approach is Path MTU discovery
  - Each packet is sent with the DF bit set – don’t fragment
  - If a router cannot handle the packet size it sends an ICMP (Internet Control Message Protocol) to the sender host telling it to fragment its packets to a smaller size

Path MTU Discovery



- May cause initial packets to be dropped, but host can learn optimal size quickly and reduce subsequent fragmentation
- Fragmentation may still have to occur between hosts, unless upper layers can be informed of the size restriction
  - This is one reason why TCP/IP are typically implemented together so they can share such information
  - UDP relies on PMTUs discovered by TCP

IPv4 vs. IPv6 Fragmentation

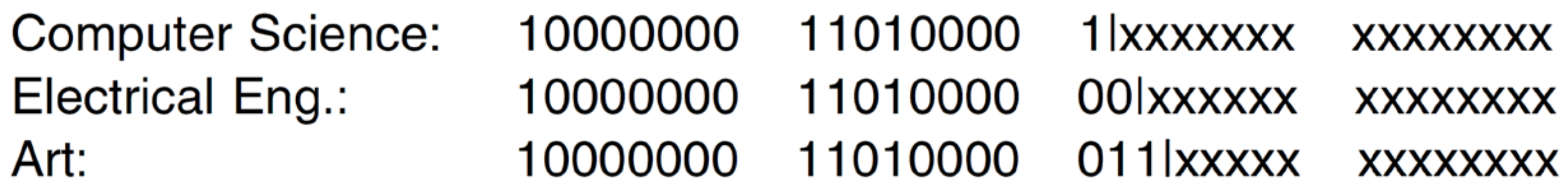
- IPv4 allows for either nontransparent fragmentation, or path MTU discovery
  - IPv4 minimum accept size 576 bytes
- IPv6 expects hosts to discover the optimal path MTU, routers will not perform fragmentation in IPv6
  - IPv6 minimum accept size 1280 bytes
- Caution: ICMP messages are sometimes dropped by networks, causing Path MTU discovery to fail, in such circumstances a connection will work for low volume, fails at high volume – if in doubt send at the minimum accept size
- https://labs.ripe.net/Members/gih/evaluating-ipv4-and-ipv6-packet-fragmentation/document_view_resolve

Subnets

- Recall: a prefix in IP addressing indicates different destination networks on the internet
- The same approach can be used internally within an organisation to maximise the use of their assigned IP prefix
- Subnetting allows networks to be split into several parts for internal use whilst acting like a single network for external use
- Subnet masks can be written using:
  - "dotted decimal"(e.g. 255.255.255.128) or
  - "slash"notation (e.g. /25)

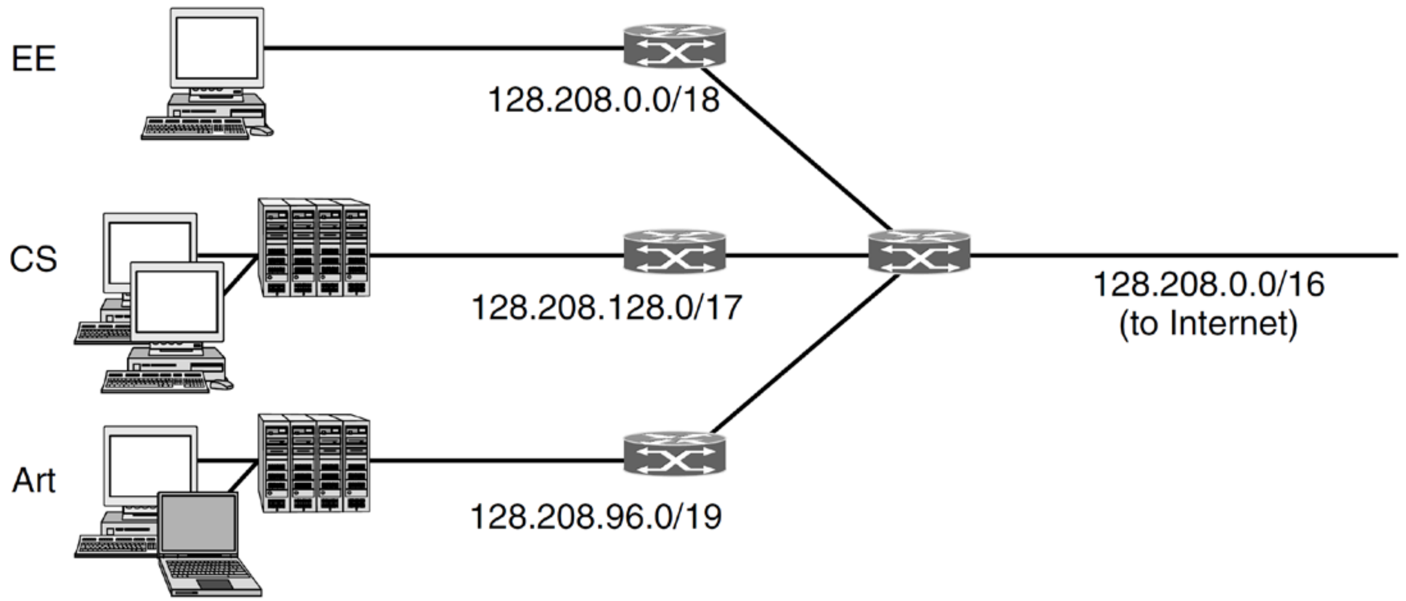
Subnets

- Example: A university with a /16 prefix could subnet its network as follows:
  - Computer Science /17 (half of allocation)
  - Electrical Engineering /18 (quarter of allocation)
  - Arts /19 (1/8 of allocation)
- Splits don't need to be even, but bits must be aligned to allow hosts portion to be used



Computer Science:	10000000	11010000	1|xxxxxxx	xxxxxxxx
Electrical Eng.:	10000000	11010000	00|xxxxxx	xxxxxxxx
Art:	10000000	11010000	011|xxxxx	xxxxxxxx

Subnets



- When a packet arrives from the internet, the router can use the subnet masks (bitwise AND) to find which subnet it should send the packet to, without knowing all hosts on the subnet

Subnets

Network	Prefix	Network Address (binary)
EE	128.208.0.0/18	10000000.11010000.00000000.00000000
CS	128.208.128.0/17	10000000.11010000.10000000.00000000
Arts	128.208.96.0/19	10000000.11010000.01100000.00000000

Network	Prefix	Subnet Mask	Binary Subnet Mask
EE	128.208.0.0/18	255.255.192.0	11111111.11111111.11000000.00000000
CS	128.208.128.0/17	255.255.128.0	11111111.11111111.10000000.00000000
Arts	128.208.96.0/19	255.255.224.0	11111111.11111111.11100000.00000000

- Example, packet comes in for 128.208.2.151, 10000000.11010000.00000010.10010111

Subnets

Network	Prefix	Network Address (binary)
EE	128.208.0.0/18	10000000.11010000.00000000.00000000
CS	128.208.128.0/17	10000000.11010000.10000000.00000000
Arts	128.208.96.0/19	10000000.11010000.01100000.00000000

Network	Prefix	Subnet Mask	Binary Subnet Mask
EE	128.208.0.0/18	255.255.192.0	11111111.11111111.11000000.00000000
CS	128.208.128.0/17	255.255.128.0	11111111.11111111.10000000.00000000
Arts	128.208.96.0/19	255.255.224.0	11111111.11111111.11100000.00000000

Incoming	10000000.11010000.00000010.10010111
AND CS Subnet Mask	11111111.11111111.10000000.00000000
Result	10000000.11010000.00000000.00000000
CS Network	10000000.11010000.10000000.00000000

Subnets

Network	Prefix	Network Address (binary)
EE	128.208.0.0/18	10000000.11010000.00000000.00000000
CS	128.208.128.0/17	10000000.11010000.10000000.00000000
Arts	128.208.96.0/19	10000000.11010000.01100000.00000000

Network	Prefix	Subnet Mask	Binary Subnet Mask
EE	128.208.0.0/18	255.255.192.0	11111111.11111111.11000000.00000000
CS	128.208.128.0/17	255.255.128.0	11111111.11111111.10000000.00000000
Arts	128.208.96.0/19	255.255.224.0	11111111.11111111.11100000.00000000

Incoming	10000000.11010000.00000010.10010111
AND Arts Subnet Mask	11111111.11111111.11100000.00000000
Result	10000000.11010000.00000000.00000000
Arts Network	10000000.11010000.01100000.00000000

Subnets

Network	Prefix	Network Address (binary)
EE	128.208.0.0/18	10000000.11010000.00000000.00000000
CS	128.208.128.0/17	10000000.11010000.10000000.00000000
Arts	128.208.96.0/19	10000000.11010000.01100000.00000000

Network	Prefix	Subnet Mask	Binary Subnet Mask
EE	128.208.0.0/18	255.255.192.0	11111111.11111111.11000000.00000000
CS	128.208.128.0/17	255.255.128.0	11111111.11111111.10000000.00000000
Arts	128.208.96.0/19	255.255.224.0	11111111.11111111.11100000.00000000

Incoming	10000000.11010000.00000010.10010111
AND EE Subnet Mask	11111111.11111111.11000000.00000000
Result	10000000.11010000.00000000.00000000
EE Network	10000000.11010000.00000000.00000000

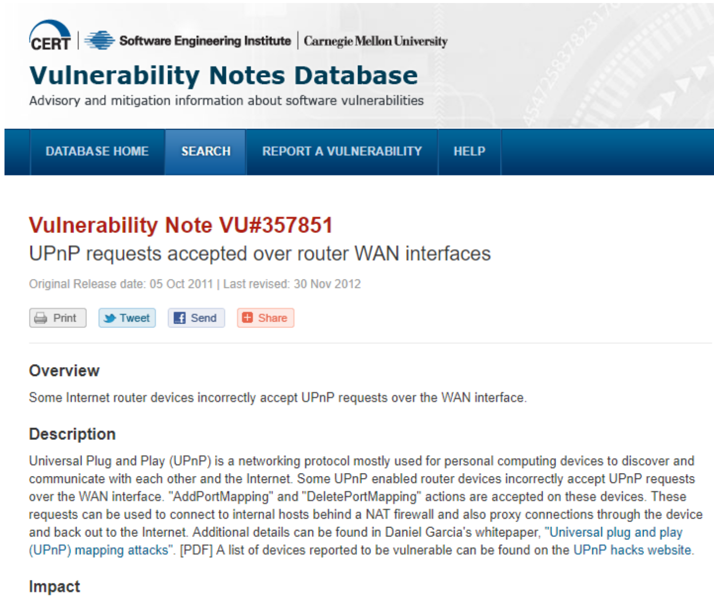
Match!

Subnets

- Future changes can be made without any external impact
  - No need to request additional IP address allocation
  - Routing on the internet does not change, only internally

And finally…

- A number of the security benefits of NAT have been undermined by functionality like Universal Plug and Play (UPnP)
- As internet usage became more extensive, the necessity for home machines to act as servers grew
  - Gaming, chat, media players, etc.
- UPnP implements the Internet Gateway Device Protocol that allows port mappings to be created in the NAT translation table to allow servers to be run on the internal network
- Bad implementations of UPnP have allowed attackers to alter translation tables from external IP addresses


CERT | Software Engineering Institute | Carnegie Mellon University
DATABASE HOME
SEARCH
REPORT A VULNERABILITY
HELP
Print
Tweet
Send
Share

Vulnerability Notes Database

Advisory and mitigation information about software vulnerabilities

Vulnerability Note VU#357851

UPnP requests accepted over router WAN interfaces

Original Release date: 05 Oct 2011 | Last revised: 30 Nov 2012

Overview

Some Internet router devices incorrectly accept UPnP requests over the WAN interface.

Description

Universal Plug and Play (UPnP) is a networking protocol mostly used for personal computing devices to discover and communicate with each other and the Internet. Some UPnP enabled router devices incorrectly accept UPnP requests over the WAN interface. "AddPortMapping" and "DeletePortMapping" actions are accepted on these devices. These requests can be used to connect to internal hosts behind a NAT firewall and also proxy connections through the device and back out to the Internet. Additional details can be found in Daniel Garcia's whitepaper, "Universal plug and play (UPnP) mapping attacks". [PDF] A list of devices reported to be vulnerable can be found on the UPnP hacks website.

Impact

Acknowledgement

- The slides were adapted by Lachlan Andrew from those prepared by Chris Culnane based on material developed previously by: Michael Kirley, Zoltan Somogyi, Rao Kotagiri, James Bailey and Chris Leckie.
- Some of the images included in the notes were supplied as part of the teaching resources accompanying the text books listed in lecture 1.