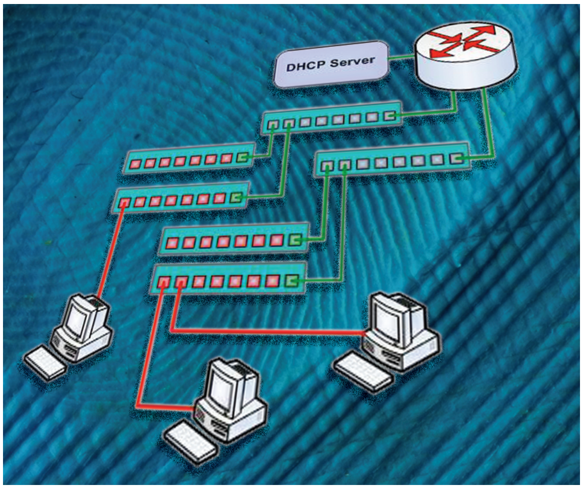
Next Generation DHCP Deployments

Dave Hull and George F. Willard III

As device mobility has transformed a novelty into a user expectation, the need for managed dynamic network configuration in campus and wireless environments has grown exponentially. User mobility and ease of end-user device network configuration have become key requirements when designing multi-user accessible networks. Additional challenges in this environment include maintaining security access controls, usage tracking, billing, and end-user support.

To support these requirements, the Dynamic Host Configuration Protocol (DHCP) provides the foundation of network auto-configuration, but it must also extend beyond the demands of traditional DHCP servers to meet the security and functionality requirements of next generation deployments. Before DHCP services can be properly utilized, weaknesses of the protocol and server implementation should be understood to ensure proper deployment and integration with other services. Non-traditional uses of DHCP to detect client operating systems, emerging extensions to DHCP, and security measures that can be taken to protect DHCP deployments will also be explored.



Driving Factors

Information security officers and systems and network administrators all want to control which devices are allowed onto their networks, but this is not always possible. Internet service providers would be overwhelmed if they had to pre-approve every device on their networks, especially when many corner coffee shops offer wireless access to their customers.

Large universities are another environment where pre-approving all network devices is a Herculean task. University networks are usually well connected to the Internet and are therefore popular targets for hackers, but many systems in these environments are owned by students more concerned with their mp3 collections than their operating systems or anti-virus definitions.

Several commercial products attempt to address these issues, but they often fall short of providing a comprehensive solution that scales reasonably. Captive portals, where users are redirected to a login page while their systems are scanned for vulnerabilities, often require gateways to be installed in front of every Ethernet (layer 2) segment of the network. While this approach provides a level of intra-subnet isolation, it still allows inter-subnet communication and thus infection in the case of a worm or virus outbreak. In response to default client firewalls like Windows XP service pack 2, many of these systems have resorted to a client agent that allows for end-user device scanning for proper configuration and patches. Installation of software on an end-user system for which you do not have administrative control can quickly generate a large number of end-user support issues.

Operating System Fingerprinting

Increasingly, security and network administrators are relying on device registration systems for access control as well as intrusion detection and prevention systems to maintain network integrity. Additionally, administrators are actively scanning hosts looking for vulnerable systems so they can proactively instruct the owners of those systems to apply the appropriate patches.

Remote OS fingerprinting is useful for filtering out false positives from network scans and IDS logs in these environments. If a remote system is shown to be running Linux, a Microsoft SQL Server vulnerability reported for that host may be a false positive.

Current fingerprinting techniques fall into two categories: active and passive. Fyodor's NMap Security Scanner demonstrated in Figure 1 falls into the former category. NMap examines various elements of TCP/IP packets to determine the remote host's operating system. Essentially, utilities like NMap do a "call and response" with a remote host, sometimes sending specially crafted packets to the remote system and examining the packets that come back from the host. Different operating systems respond differently to these packets, and these responses contain enough information to make a determination about which operating system is running on the remote host.

Although active TCP/IP stack fingerprinting is highly effective, it does have one major disadvantage. Active fingerprinting requires communication with the remote host and, because this may include the use of unusual or malformed packets, an intrusion detection system or personal firewall on the remote system may detect these odd packets and sound an alarm or discard them.

Passive fingerprinting, on the other hand, can be undetectable. Tools like p0f [1] also analyze TCP/IP headers to determine a remote host's OS. However, passive tools eavesdrop on the conversations of others and determine the remote operating systems without participating in the conversation.

University of Kansas

The University of Kansas data network is large and diverse with centralized management to the edge and distributed management of the end nodes. Every fall, thousands of students move into residence halls where they have the option of plugging their computers (often unpatched and unprotected) into the University's network. If and when students do connect to the campus network, they find their access to the campus network and the Internet limited.

When a student in a residence hall opens a Web browser, he is automatically redirected to a Web site where he must register his computer and purchase Internet access before he can access the KU network and the Internet as shown in Figure 2.

As part of the registration process, ResNet customers' computers are checked for the latest operating system updates, University-supported anti-virus software and up-to-date virus definition files. Each of these operations is handled by RINGS (Resnet Integrated Next-Generation System) [2], an internally developed J2EE-based registration system that contains functionality similar to the popular NetReg [3] system. Additionally, the MAC addresses of the students' computers are recorded in a directory server along with the DHCP-assigned IP addresses and other customer device information. Recording the customers' operating systems and detailed network location information allows for rapid problem detection and resolution regarding client mis-configuration and security incident handling.

DHCP OS Fingerprinting

In large deployments and transient networks, a better passive fingerprinting technique that provides a high degree of accuracy is needed. Enter the software engineering team at the KU Networking and Telecommunications Services (NTS), a division of Information Services. While developing the University's Advanced Network Services Registry (ANSR) system [4], a suite of directory-enabled applications for network management, the team discovered a new technique for passive OS fingerprinting.

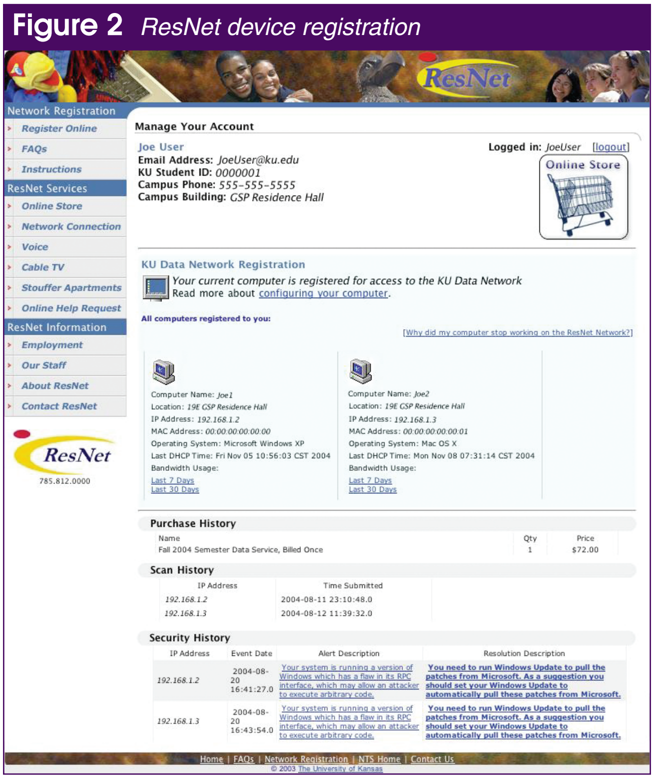
Figure 1 *NMAP fingerprinting*

```
root@localhost root# nmap -sT -O -P0 192.168.2.2

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Warning:  OS detection will be MUCH less reliable because we did not find at least 1 open
and 1 closed TCP port
Interesting ports on (192.168.2.2):
(The 1600 ports scanned but not shown below are in state: filtered)
Port       State       Service
22/tcp     open        ssh
Remote operating system guess: AIX 4.3.2.0-4.3.3.0 on an IBM RS/*

Nmap run completed -- 1 IP address (1 host up) scanned in 580 seconds
```

Figure 2 *ResNet device registration*



Manage Your Account

Joe User

Logged in: *JoeUser* [logout]

Email Address: *JoeUser@ku.edu*
KU Student ID: *0000001*
Campus Phone: *555-555-5555*
Campus Building: *GSP Residence Hall*

KU Data Network Registration

Your current computer is registered for access to the KU Data Network
Read more about configuring your computer.

All computers registered to you:

[Why did my computer stop working on the ResNet Network?]

Computer Name: *Joe1* Location: *19E GSP Residence Hall* IP Address: *192.168.1.2* MAC Address: *00:00:00:00:00:00* Operating System: Microsoft Windows XP Last DHCP Time: Fri Nov 05 10:56:03 CST 2004 Bandwidth Usage: Last 7 Days Last 30 Days	Computer Name: *Joe2* Location: *19E GSP Residence Hall* IP Address: *192.168.1.3* MAC Address: *00:00:00:00:00:01* Operating System: Mac OS X Last DHCP Time: Mon Nov 08 07:31:14 CST 2004 Bandwidth Usage: Last 7 Days Last 30 Days

Purchase History

Name	Qty	Price
Fall 2004 Semester Data Service, Billed Once	1	$72.00

Scan History

IP Address	Time Submitted
192.168.1.2	2004-08-11 23:10:48.0
192.168.1.3	2004-08-12 11:39:32.0

Security History

IP Address	Event Date	Alert Description	Resolution Description
192.168.1.2	2004-08-20 16:41:27.0	Your system is running a version of Windows which has a flaw in its RPC interface, which may allow an attacker to execute arbitrary code.	**You need to run Windows Update to pull the patches from Microsoft. As a suggestion you should set your Windows Update to automatically pull these patches from Microsoft.**
192.168.1.3	2004-08-20 16:43:54.0	Your system is running a version of Windows which has a flaw in its RPC interface, which may allow an attacker to execute arbitrary code.	**You need to run Windows Update to pull the patches from Microsoft. As a suggestion you should set your Windows Update to automatically pull these patches from Microsoft.**

Home | FAQs | Network Registration | NTS Home | Contact Us
© 2003 The University of Kansas

The technique developed at KU has been dubbed "DHCP Fingerprint", and it relies on DHCP Options as defined in RFC 2132 [5]. According to the RFC, "configuration parameters and other control information are carried in tagged data items that are stored in the 'options' field of the DHCP message." Each operating system requests a different ordered set of options, thus the operating system may be determined by examining the options requested by the client.

KU's NTS software engineering team uses this passive OS-fingerprinting technique to determine the operating systems of ResNet customers. This information is then recorded in the ANSR directory server where it can be used by RINGS and other ANSR applications, such as those that restrict rogue wireless access points or broadband routers from accessing the network — even in cases where students are savvy enough to clone the MAC addresses of legitimately registered devices.

Besides providing information to KU's integrated systems, the operating system information may also be used for planning and analysis. Figure 3 shows the current ResNet operating system distribution. This graph clearly indicates that our ResNet student consultants should have Microsoft Windows XP training and experience.

Figure 3 *ResNet operating system distribution*

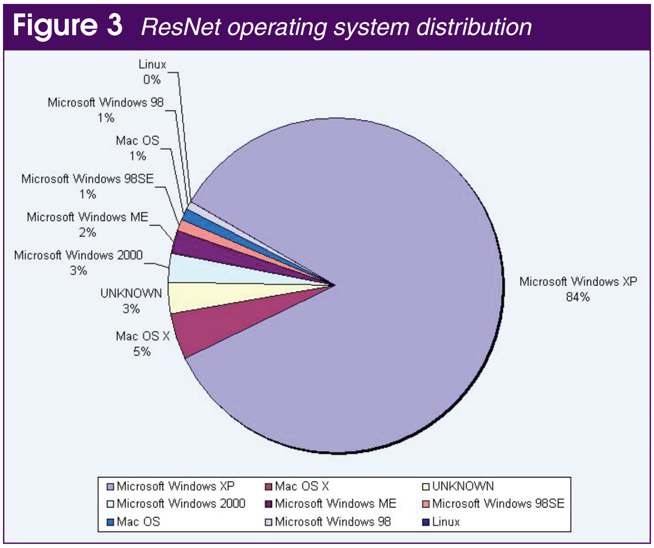
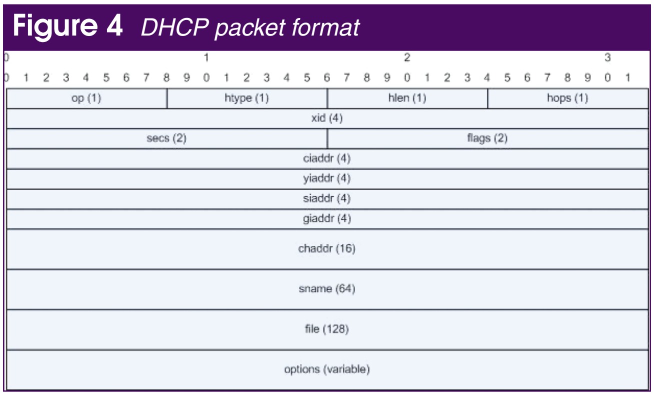


Figure 4 *DHCP packet format*



op (1) | htype (1) | hlen (1) | hops (1)
xid (4)
secs (2) | flags (2)
ciaddr (4)
yiaddr (4)
siaddr (4)
giaddr (4)
chaddr (16)
sname (64)
file (128)
options (variable)

How DHCP OS Fingerprinting Works

DHCP options are used by DHCP clients to request additional information from the DHCP server. Each option is identified by a unique code or "tag octet". Figure 4 shows the format for a DHCP message.

Options include things like the SMTP server option (code 69), domain name server option (code 6), or NetBIOS over TCP/IP name server option (code 44). Option code 55 contains the parameter request list or the list of requested options themselves. According to RFC 2132, the list may be in the client's order of preference. Option 43 is reserved for vendor-specific information and may contain additional vendor-defined options. Options 128 through 254 are reserved for vendor- or site-specific purposes.

By design, a freshly booted DHCP client broadcasts a DHCPDISCOVER message on the network. In most cases, the client broadcasts include requests for optional information via option 55. After the client receives one or more DHCPOFFER messages, it responds by broadcasting a DHCPREQUEST message that includes an option indicating which server's offer is being accepted. The DHCPREQUEST message may contain optional parameters, and they may be the same or different from the optional parameters included in the DHCPDISCOVER message.

A single DHCPDISCOVER or DHCPREQUEST message contains enough information within option 55 alone to accurately fingerprint a remote operating system. Contrast this with the traditional TCP/IP stack fingerprinting methods used by NMap or p0f where multiple packets are needed to conclusively identify the remote host.

ANSRdhcp

One of the first features added to KU's ANSRdhcp server was the recording of a DHCP client's last requested options. Recording the last requested options in a human-readable format has several advantages. With the delegated administration of the device-centric data stored in the ANSR directory server, it would be inconvenient at best for a systems administrator to obtain server-side logging information or DHCP packet captures to determine which options a device is requesting.

Systems administrators on KU's campus can log in to the ANSR system and look at the options their DHCP clients are requesting from the central campus ANSRdhcp server. Administrators may also add the desired responses to these options to ANSR, and the DHCP server will pass those responses on to the clients the next time they send a DHCPDISCOVER or DHCPREQUEST message. The requested options are converted from byte form to the human-readable directory attribute name. This provides a "fill in the blank" approach for administrators to ensure that their DHCP clients receive the settings that they are requesting. The hierarchical structure of the directory service is used to leverage inheritance of settings, so that common settings can be applied at a parent entry instead of every client entry.

There are a number of devices on the market that claim they will only work with a proprietary DHCP server when, in reality, all that is needed is a way to instruct a DHCP server to provide the proper response to the device's request for vendor-specific information. This is a critical capability that ANSR system possesses, along with the ability to override handling of any option at any level, even on a per-device basis to accommodate client-specific needs.

Figure 5 *Sample entries from fingerprints.txt*

```
1,3,6,12,15,28,44,Linksys WRT54G
1,28,2,3,15,6,12,40,41,42,Linux
1,3,6,15,112,113,78,79,95,-4,Mac OS X
1,15,3,6,44,46,47,31,33,249,43,Microsoft Windows XP
1,15,3,6,44,46,47,31,33,43,77,Microsoft Windows ME
1,15,3,6,44,46,47,31,33,43,Microsoft Windows 2000
Pro
```

During the development of the ANSR system, the software engineering team noticed that client devices were requesting different options from the DHCP server with enough variation in ordering and requested options to be used to fingerprint devices using the DHCP server.

DHCP Listener

To test this idea, the software team initially developed a standalone Java application called DHCPListener [6] that runs on a host and listens for broadcast traffic to port 67, the default DHCP server port. As packets come in, the application parses them and examines the DHCP options information. Option 53 contains the DHCP message type. If the value of option 53 is 1 or 3, the packet contains DHCPDISCOVER or DHCPREQUEST information, respectively, and the parsing continues; otherwise the packet is ignored.

Following the extraction of the message type, DHCPListener attempts to extract additional options, if they are present, including the vendor class identifier that may reveal some basic information about the remote host. For example, Windows 2000 and XP report MSFT 5.0, while Windows 98 reports MSFT 98 instead.

Option 55, however, contains the real meat of the fingerprinting technique. To date, the team has collected unique signatures of 19 distinct devices and operating systems. When DHCPListener parses option 55's contents, it takes the list of options and uses them to retrieve the device or operating system from a hash map that is loaded from a text file when the application starts. Figure 5 shows some of the entries from the fingerprints.txt file.

DHCPListener generates XML markup as shown in Figure 6. The XML output may be validated against a DTD to easily be processed by other programs and systems, or processed by a style sheet for tabular display. For unknown operating systems, the operatingSystem tags will contain a braced list of requested options. For example, the list might resemble `{1,66,6,3,67,12,-106}`. If the DHCPListener operator knows the operating system for the device in question, he may simply add it to the fingerprints.txt file and restart the listener. In this case, the device is actually a Cisco 2900 Catalyst XL switch using DHCP to obtain its own IP address and network information.

Figure 6 *Example DHCPListener output*

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dhcpMessageList SYSTEM
 "dhcpMessageList.dtd">
<dhcpMessageList>
        <dhcpMessage>
                <messageType>
                        DISCOVER
                </messageType>
                <macAddress>
                        00:02:d3:02:67:ea
                </macAddress>
                <requestedIP>
                        null
                </requestedIP>
                <systemName>
                        netbotz0267EA^@
                </systemName>
                <operatingSystem>
                        NetBotz
                </operatingSystem>
                <vendorClassID>
                        NBTZ 1.3
                </vendorClassID>
                <date>
                        Wed Oct 06 11:41:29 CDT 2004
                </date>
        </dhcpMessage>
</dhcpMessageList>
```

Some devices have been found to give out no information via the optional parameters list. In these cases, DHCPListener fills the operating system tag with the word “NONE”. In most situations, the device in question has turned out to be a Linksys broadband router.

ANSRdhcp Fingerprint Integration

The fingerprints.txt file used by DHCPListener is for demonstration purposes, but the concept may be easily extended into production DHCP servers. Instead of relying upon a file, the ANSRdhcp server obtains DHCP fingerprints, as well as other server configuration options, directly from the directory server that hosts the networked device information. The dhcpDeviceFingerprint attribute contains a description and the requested options list in human-readable format. A visual LDAP browser, like the Softerra LDAP Administrator [7] tool, may be used to manage DHCP fingerprints as well as all other directory-hosted information as shown in Figure 7.

DHCP Relay Agent Information Option

When a DHCP client broadcasts a discover message on a subnet, and a DHCP server is listening on the same subnet, there is minimal information that the server can use to decide how to service the request. This information includes the MAC address of the client, optional client-provided identification information (including a client identifier and vendor class identifier), and the options that the client is requesting from the server. The server makes a decision on how to service the client from this information.

When a DHCP relay is used to forward the request to the DHCP server, the relay address “giaddr” may also be used to determine how to service the request. This is implemented by using either a DHCP relay server or more typically via IP helper functionality on Cisco routers. In most cases, the relay address acts as a subnet selector to determine which pool or set of static assignments is appropriate. This may also be taken a step further into the concept of having a device registered to a home subnet, and as a registered device, allowed to obtain a lease from “roaming” pools on other subnets. Using this technique, however, still does not provide access control granularity beyond the subnet where the device is attached.

The DHCP relay agent information option as defined in RFC 3046 provides additional information to the DHCP server to make fine-grained decisions about host configuration. The DHCP relay agent information option, also known as option 82, is injected into the DHCP request packets by the network device to which the client is connected, typically a layer-2 Ethernet switch. Figure 8 shows an outline of the Cisco relay agent information option format. Additional information appended to the request includes the VLAN number, interface index, and the MAC address of the network device itself. From this information, the server can determine that the request came from a specific switch on a specific port and knows the active VLAN.

A common use of the relay agent information option is to limit the number of unique MAC addresses that may DHCP boot on a given network port. More sophisticated implementations may track the switch and port information and map it to circuit or jack locations so that hosts may be located quickly for security or help-desk reasons.

Figure 9 shows how inbound discover packets are processed, and the information that is recorded from the request directly into the device’s directory server entry. Note that the IP address of the network switch must be resolved via the MAC address. To accommodate this, the edge network switches are also registered devices in the directory. Figure 10 shows an example device entry that contains the resulting decoded information.

This additional network information can also be used to make detailed decisions. For example, you may want to provide different information via DHCP to a test-bench switch on the same subnet as other switches. The downside to utilizing this option is that the relay agent information option provides a great deal of information about the layer-2 physical topology to a potential attacker. An attacker could passively build a complete map of the edge switch topology, including the locations of each DHCP client. After this information is injected into the client’s DHCP request, it is then broadcast visible on the subnet unless additional precautions are taken.

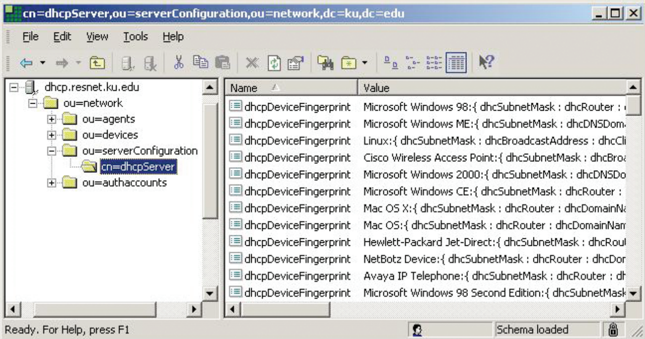
DHCP Snooping

To prevent the broadcast visibility of DHCP requests and to enforce additional DHCP controls, Cisco utilizes a configuration feature called DHCP snooping. Three major benefits of DHCP snooping include elimination of DHCP broadcast request visibility, prevention of rogue DHCP servers, and protection against DHCP server address exhaustion attacks.

DHCP snooping defines a trust model where a port on a switch is either trusted or not trusted. A trusted port may answer DHCP requests. If your DHCP server is located on the same subnet, it must be connected to a trusted port. If however, your DHCP server is on another subnet and accessible via DHCP relay, the uplink port of the switch must be trusted. This allows for the DHCP requests to reach the server. If the port is not trusted, DHCP serving is not allowed from the port; however, client DHCP requests may be serviced. Figure 11 shows an example of KU’s DHCP snooping network topology for the ResNet network.

The configuration needed to enable DHCP snooping is relatively simple. Figure 12 shows the configuration for an example edge switch. Interface Fa0/48 is used as the uplink port, and thus must have

Figure 7 *ANSRdhcp server configuration*



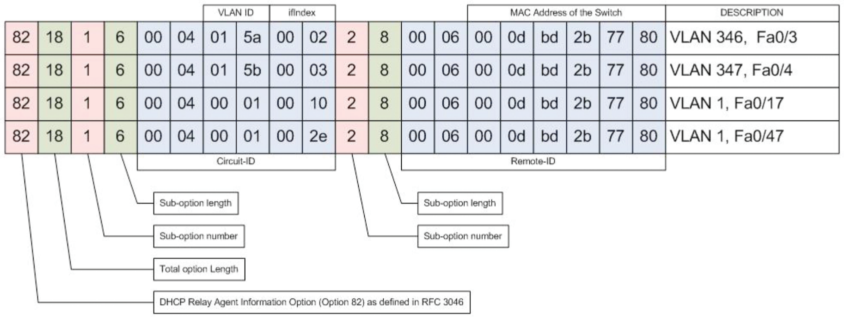
cn=dhcpServer,ou=serverConfiguration,ou=network,dc=ku,dc=edu

- dhcp.resnet.ku.edu
  - ou=network
    - ou=agents
    - ou=devices
    - ou=serverConfiguration
      - cn=dhcpServer
    - ou=authaccounts

Name	Value
dhcpDeviceFingerprint	Microsoft Windows 98:{ dhcSubnetMask : dhcRouter :
dhcpDeviceFingerprint	Microsoft Windows ME:{ dhcSubnetMask : dhcDNSDom
dhcpDeviceFingerprint	Linux:{ dhcSubnetMask : dhcBroadcastAddress : dhcCli
dhcpDeviceFingerprint	Cisco Wireless Access Point:{ dhcSubnetMask : dhcBroa
dhcpDeviceFingerprint	Microsoft Windows 2000:{ dhcSubnetMask : dhcDNSDo
dhcpDeviceFingerprint	Microsoft Windows CE:{ dhcSubnetMask : dhcRouter :
dhcpDeviceFingerprint	Mac OS X:{ dhcSubnetMask : dhcRouter : dhcDomainN
dhcpDeviceFingerprint	Mac OS:{ dhcSubnetMask : dhcRouter : dhcDomainNam
dhcpDeviceFingerprint	Hewlett-Packard Jet-Direct:{ dhcSubnetMask : dhcRou
dhcpDeviceFingerprint	NetBotz Device:{ dhcSubnetMask : dhcRouter : dhcDor
dhcpDeviceFingerprint	Avaya IP Telephone:{ dhcSubnetMask : dhcRouter : dh
dhcpDeviceFingerprint	Microsoft Windows 98 Second Edition:{ dhcSubnetMask

Ready. For Help, press F1 | Schema loaded

Figure 8 *Cisco relay agent information option*



						VLAN ID		ifIndex						MAC Address of the Switch						DESCRIPTION
82	18	1	6	00	04	01	5a	00	02	2	8	00	06	00	0d	bd	2b	77	80	VLAN 346, Fa0/3
82	18	1	6	00	04	01	5b	00	03	2	8	00	06	00	0d	bd	2b	77	80	VLAN 347, Fa0/4
82	18	1	6	00	04	00	01	00	10	2	8	00	06	00	0d	bd	2b	77	80	VLAN 1, Fa0/17
82	18	1	6	00	04	00	01	00	2e	2	8	00	06	00	0d	bd	2b	77	80	VLAN 1, Fa0/47
				Circuit-ID								Remote-ID								

ip dhcp snooping trust enabled. All other 47 client connection ports are not trusted and are also configured with a limit rate of 30 DHCP packets per second to suppress DHCP server denial of service attacks. Note that you may need to upgrade your Cisco IOS image on these switches to support these relatively new features. The IOS revision that KU used is also shown in Figure 12.

The router configuration in Figure 13 shows the configuration options that must be added to each VLAN that uses the DHCP snooping model. First, the address of the DHCP server is specified as the ip helper-address. Second, the ip dhcp relay information trusted option must be set. This ensures that DHCP traffic will be relayed to only the DHCP server and that the server specified may receive the additional relay agent information option.

Because a port must be trusted to be a DHCP server, the threat of rogue DHCP servers is eliminated. KU had previously experienced numerous ResNet customer service outages due to wireless broadband routers being connected to their internal network interfaces instead of the WAN ports. These improperly connected broadband routers then handed out RFC 1918 [8] local addresses to other clients instead of proper ResNet addresses. This feature of DHCP snooping alone has saved countless hours of customer service headaches.

Cisco's DHCP snooping implementation also allows for limiting the rate of DHCP requests per port, per second. This greatly reduces the amount of traffic generated by a DHCP server address exhaustion attack, preventing denial of service against the DHCP server. In combination with a packet throttle built into the DHCP server, DHCP snooping has prevented DHCP service outages.

DHCP Lease Query

A DHCP server contains authoritative information regarding the mapping of MAC addresses to IP addresses. This information can be critical for tracking network usage in a layer-3 routed environment where a client may roam across subnets. The IETF draft for DHCP lease query, draft-ietf-dhc-leasequery-07.txt, defines how DHCP lease query performs the lookup and has already been successfully pre-standard implemented by some vendors, including Ellacoya Networks[9].

Other uses for this information mapping include passive authentication through layer-3 gateways such that only hosts registered in

Figure 9 *ANSRdhcp request recording logic*

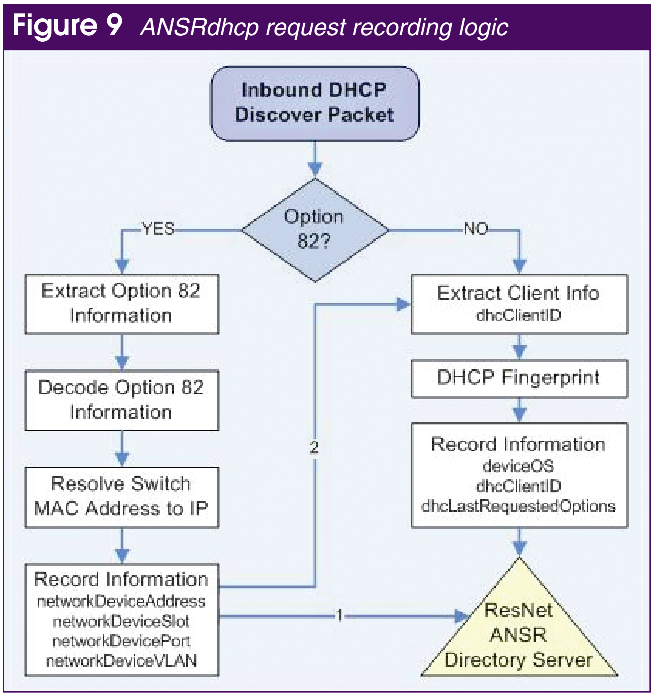
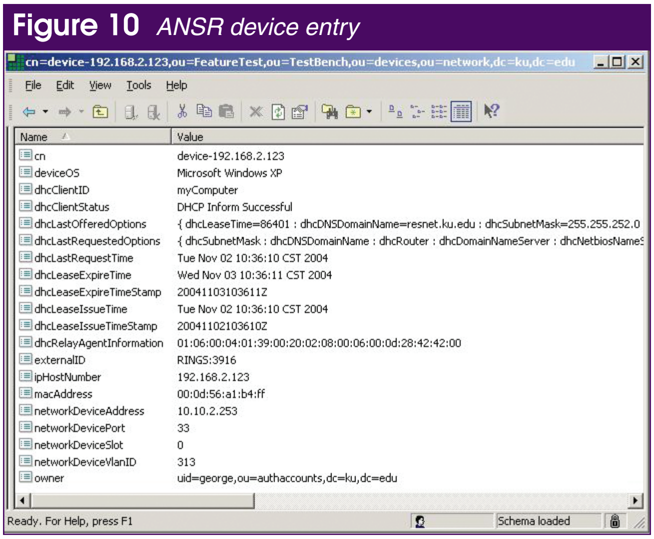


Figure 10 *ANSR device entry*



cn=device-192.168.2.123,ou=FeatureTest,ou=TestBench,ou=devices,ou=network,dc=ku,dc=edu

Name	Value
cn	device-192.168.2.123
deviceOS	Microsoft Windows XP
dhcClientID	myComputer
dhcClientStatus	DHCP Inform Successful
dhcLastOfferedOptions	{ dhcLeaseTime=86401 : dhcDNSDomainName=resnet.ku.edu : dhcSubnetMask=255.255.252.0
dhcLastRequestedOptions	{ dhcSubnetMask : dhcDNSDomainName : dhcRouter : dhcDomainNameServer : dhcNetbiosName
dhcLastRequestTime	Tue Nov 02 10:36:10 CST 2004
dhcLeaseExpireTime	Wed Nov 03 10:36:11 CST 2004
dhcLeaseExpireTimeStamp	20041103103611Z
dhcLeaseIssueTime	Tue Nov 02 10:36:10 CST 2004
dhcLeaseIssueTimeStamp	20041102103610Z
dhcRelayAgentInformation	01:06:00:04:01:39:00:20:02:08:00:06:00:0d:28:42:42:00
externalID	RINGS:3916
ipHostNumber	192.168.2.123
macAddress	00:0d:56:a1:b4:ff
networkDeviceAddress	10.10.2.253
networkDevicePort	33
networkDeviceSlot	0
networkDeviceVlanID	313
owner	uid=george,ou=authaccounts,dc=ku,dc=edu

Figure 11 *DHCP snooping network topology*

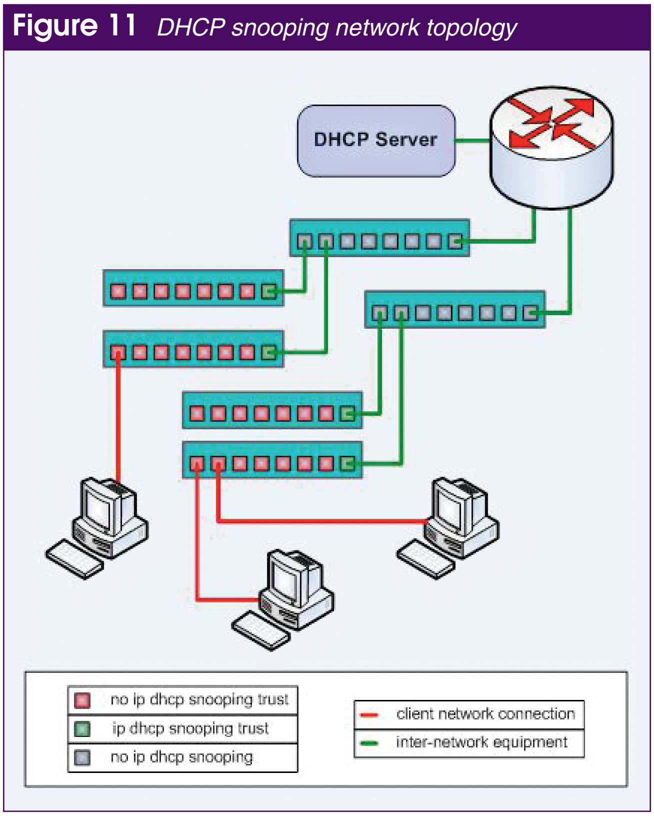


Figure 12 *Edge network switch configuration*

Cisco Catalyst 2950G 48
C2950-I6K2L2Q4-M, Version 12.1(22)EA1

```
ip dhcp snooping vlan 123
ip dhcp snooping

interface FastEthernet0/1-47
 switchport access vlan 123
 switchport mode access
 ip dhcp snooping limit rate 30

interface FastEthernet0/48
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 123
 switchport mode trunk
 ip dhcp snooping trust
```

the DHCP server may pass traffic through a layer-3 boundary, perhaps through a border router to the Internet. This also allows for Internet usage billing and policy application at a granular level. For example, some hosts in the DHCP server may be allowed to pass peer-to-peer traffic while others may not.

Gateway devices often maintain the policy information and grouping constructs for policy application. The gateway device utilizes unicast DHCP lease query messages to resolve layer-2 MAC addresses to layer-3 IP address mapping information.

When using a DHCP server with lease query support, the server should be configured to answer lease queries only from trusted sources (i.e., the gateway devices). Otherwise, an attacker could mine the complete layer-2 to layer-3 map of your network through remote unicast packets. This is the equivalent of remotely obtaining a complete ARP table.

Figure 14 shows the flow of inbound DHCP packets through the lease query packet filter. If the lease query request originated from an authorized host, as specified in the directory server configuration dhcpAuthorizedLeaseQueryHost attribute, then the lease query will be processed. If the requester is not authorized, the packet is discarded. The ANSR directory server is then searched by IP address for corresponding registered device entry. If the entry is found, a lease query response is generated containing the device's MAC address and lease time remaining. Otherwise, if the host is not found, a response is generated that does not contain any device-specific information indicating that a device registered to the requested IP address was not found.

The network location of your DHCP server is also critical to a secure DHCP server deployment. If the DHCP server is located on the same subnet it services, the server may be more vulnerable to local subnet ARP style redirect and spoofing attacks to obtain this information. For this reason, it is good practice to place the DHCP server on a separate segment where it does not answer local DHCP requests; only those obtained via specified trusted relay addresses are answered.

Conclusion

Systems administrators in large environments have welcomed DHCP and with good reason; it has cured countless headaches caused by having to resolve IP conflicts and has enabled the era of mobile computing.

However, as with any technology, DHCP has brought new security challenges. Unforeseen uses of DHCP, including passively fingerprinting remote operating systems and passively mapping the entire network, require DHCP administrators to carefully consider their implementations. Hardware vendors should include features like Cisco's DHCP snooping in order to mitigate information leaks and denial of service attacks in next generation DHCP deployments.

References

1. Zalewski, Michal. p0f v2 is a passive OS fingerprinting tool. More information is available on the Internet at: `http://lcamtuf.coredump.cx/p0f.shtml`
2. The University of Kansas. ResNet Integrated Next Generation System (RINGS) is a J2EE network device registration and access control system used in the residence hall system at KU. RINGS was jointly developed by KU's Networking and Telecommunications Services and ResNet departments.
3. Valian, Peter and Watson, Todd K. 2000. NetReg: An Automated DHCP Registration System. *Sys Admin* 9(12):26-32. Published on the Internet at: `http://www.netreg.org/SysAdmin`
4. The University of Kansas. Advanced Network Services Registry (ANSR) is an LDAP-based application developed by KU's Networking and Telecommunications Services. More information is available on the Internet at: `http://www.nts.ku.edu/ \ services/data/networkmgmt/ansr/documentation/index.jsp`
5. Alexander, Steve and Droms, Ralph. March, 1997. DHCP Options and BOOTP Vendor Extensions. Published on the Internet at: `http://www.ietf.org/rfc/rfc2132.txt`
6. The University of Kansas. Source code available at: `http://www.nts.ku.edu/downloads`
7. Softerra, LLC. More information available on the Internet at: `http://www.ldapadministrator.com`
8. Rekhter, Yakov, et al. February, 1996. Address Allocation for Private Internets. Published on the Internet at: `http://www.ietf.org/rfc/rfc1918.txt`
9. Ellacoya Networks. More information available on the Internet at: `http://www.ellacoya.com`

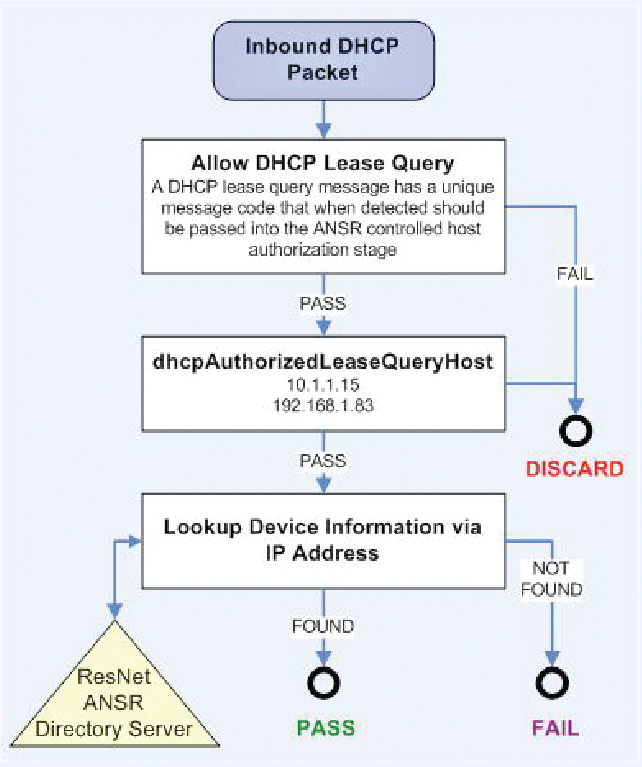
Dave Hull and George F. Willard III are employed by the Networking and Telecommunications Services division of Information Services at the University of Kansas. As the DBA and systems administrator of the software engineering team, Dave leverages data architectures and open source software running on the Linux operating system to support campus-wide networking services and initiatives. George, as the chief systems architect and manager of software engineering, is leading the development and implementation of the directory enabled network, wireless network security, and network middleware integration initiatives. Dave and George can be contacted at `dphull@ku.edu` *and* `gfwillar@ku.edu`*, respectively.*

Figure 13 *Router configuration*

Cisco Catalyst 6509

```
interface Vlan123
description DHCP Snooping Example
ip helper-address 10.1.1.2
ip dhcp relay information trusted
```

Figure 14 *ANSRdhcp lease query support*


Inbound DHCP Packet
Allow DHCP Lease Query
A DHCP lease query message has a unique message code that when detected should be passed into the ANSR controlled host authorization stage
FAIL
PASS
dhcpAuthorizedLeaseQueryHost
10.1.1.15
192.168.1.83
DISCARD
PASS
Lookup Device Information via IP Address
NOT FOUND
FOUND
ResNet ANSR Directory Server
PASS
FAIL