DHCP and the changing art of Network Security

~ or ~

Why DHCP deserves more love

James (njan) Eaton-Lee
http://www.jeremiad.org

11th December 2005

With thanks to all of those who offered time and effort in reviewing and proof-reading this paper and contributed their suggestions.

Introduction

Every network protocol is being subjected to scrutiny for security concerns – many of the protocols which drive the way in which we communicate, even the ones designed comparatively recently, were designed with efficiency and elegance in mind rather than security. Assumptions made during the construction of these protocols, mostly stemming from a design not motivated by security are either, depending upon your viewpoint, naïve and insecure, or optimistic and trusting. In many instances, the drive to build systems and protocols which paralleled systems in real life have led to insecurities which although generally accepted in conventional infrastructure and systems[1], are virtually unacceptable due to the ease of exploitation.

Either way, we can safely speculate that the world in which the designers of these protocols lived was fundamentally different to the world which we, as security professionals, inhabit now!

Mail and TCP/IP (or, more correctly, IP) are two of the main protagonists in this respect, and in the last few years numerous attempts have been made to secure both of them. Security concerns pertaining to specific abuse issues (such as spam) and more general ones (such as the interception and modification of data) have driven most of these, but recently considerations for these issues have started to become more prevalent, and there are several packages and sets of technologies which have been released, planned, and documented in the last few years to indicate that these are not issues thought about only by security-conscious Network Administrators.

Some examples of these include technical methods designed to verify the authenticity of mail (such as SPF), filtering based on the content in spam (such as packages which implement Bayesian spam filtering), and blacklisting, which are widely deployed to varying degrees of success. IPSec is one example of a security-measure designed to protect IP at the IP layer; many protocols incorporate to tls/ssl encryption standard in order to authenticate and protect data at a higher (session) layer (such as the https scheme[2]).

Consideration of Corporate/Desktop Infrastructure

The core of a corporate infrastructure is, these days, significantly more secure than when complex distributed networks with Single Sign-On started to become common, and as the vendor responsible for the majority of such corporate networks, Microsoft is to a great degree responsible for this. Active Directory is relatively secure by design both by nature of the safeguards which it ships with and the additional security features afforded by the 2003 release. Kerberos authentication, special permissions, delegation of administration, and smaller, implementational changes such as the new distinction between '*everyone'*, '*authenticated users'*, and '*anonymous users*' are a far from representative set of examples of this. Although Active Directory does not have a perfect security model and is *not* invulnerable to attack, there are comparatively few (if any) issues regularly reported with any of the Infrastructure components of the Windows product family.

The same is beginning to be true for corporate networks running on separate infrastructures such as Novell's eDirectory and as will surely be the case with RedHat

[1] http://www.unixwiz.net/advisories/unixwiz-2005-01.html
[2] http://en.wikipedia.org/wiki/Https

Directory Server (RHDS). But what has been done to address some of the most fundamental elements of a corporate infrastructure, such as TCP/IP and DHCP?

Consideration of DHCP Security within this context

Perhaps unsurprisingly, many of these issues have been thought about at Microsoft– papers such as '*Server and Domain Isolation Using IPsec and Group Policy*'[3] and technologies such as '*Network Based Quarantine Control*' [4]and '*Network Access Protection*' [5]indicate that security at the Network Layer is a consideration. The opening of the paper on IPSec discusses the risks posed to business networks, saying that:

> *"controlling physical access to a network can become impossible. Customers, vendors, and consultants may need to connect mobile devices to your network for valid business reasons."*

Given all of these considerations in addition to our consideration of the Defence in Depth principle and good security best practices, therefore, what happened to DHCP? BOOTP, DHCP's predecessor, was originally submitted as an RFC in September 1985 (RFC 951[6]), just 4 years younger than TCP, and even the latest definition of DHCP dates back to March 1997 (RFC 2131[7]); by the time Windows Vista is released and sees initial deployment, DHCP itself will be over a decade old, and yet the security issues which clients configured via DHCP face are still problems.

Centralised network management is not the vision it was in 1997 – even small companies routinely have their networks centrally (and even remotely) managed, and DHCP plays no small part in this picture. Although unglamorous and oft-ignored, managing a large, dynamic network of hosts without DHCP would be very difficult (essentially impossible, or at least highly impractical), particularly given the dynamic nature of today's networks.

For a client, DHCP is extremely important – for workstations which are dynamically configured, DHCP is a pre-requisite to virtually all communication with other hosts on the network, as without an IP address, none of the protocols which we consider essential to local and wide area networks (such as TCP, UDP, and ICMP or higher-level session and application protocols such as netbios, rpc, http, smb/cifs, etc.) will function. Furthermore, interception or misuse of DHCP has the potential, at the very least, to disrupt all of our network communications, and at most to allow an attacker to execute an attack against our network using a technology fundamental to (and therefore affecting) every other networked application which we use.

Malicious uses of DHCP

Using DHCP, an attacker who sets up a malicious DHCP server onsite can alter the IP addressing and subnetting, DNS, and routing on any host, allowing the attacker to manipulate this with potentially with no knowledge by (or visible disruption to) the user. At the very least, this would allow an intruder to cause massive disruption to a network as

[3] http://www.microsoft.com/technet/security/topics/architectureanddesign/ipsec/default.mspx
[4] http://www.microsoft.com/windowsserver2003/techinfo/overview/quarantine.mspx
[5] http://www.microsoft.com/windowsserver2003/technologies/networking/nap/default.mspx
[6]http://www.faqs.org/rfcs/rfc951.html
[7]http://www.faqs.org/rfcs/rfc2131.html

client computers had DHCP leases expire and were issued new leases by a rogue DHCP server, causing a total loss of connectivity.

This specific issue in itself is something which we can see has undergone consideration by Microsoft at one level – DHCP servers in a windows domain require ‘authorisation’, a process by which a DHCP server which is a member of the windows network is, essentially, told that it is not allowed to run until it is successfully registered with the central list of ‘authorised’ DHCP servers, preventing to some degree a malicious or incompetent administrator from causing this sort of disruption through use of a windows server which is a member of the domain on the network. This measure alone, however, does not (and necessarily can not) protect against a windows DHCP server (or any other, for that matter) which is not part of the domain, and is a basic security measure designed to guard against a casual attack or mistake at best, and relies upon the good intentions of the DHCP server process in obeying the instructions of the server telling it not to operate, and the good administration of the network, as a malicious administrator may well have access to authorise his own DHCP Server (or simply disjoin the server from the domain).

Not all issues with DHCP even arise from the risk posed by servers – another potential Denial of Service attack could result from clients. Most DHCP servers have no mechanism designed to protect against a client taking out multiple DHCP leases with faked MAC addresses. Any client doing this could effectively negate any other clients from joining onto the network by taking out a large number of leases, as there would be no available network addresses to allocate the new hosts. This is an attack which is also relatively simple to carry out (although slightly more complicated than the attack mentioned above).

In a demonstration in November of 2005 for an audience at the British Computing Society in Dundee based on a pre-publication version of this talk, the author chose to do this with no more than the 'ifconfig' command in linux, the Internet Systems Consortium DHCP Client (dhclient), and a simple bash script. The choice to use these was made ostensibly to demonstrate the ubiquitous and non-unique nature of the software required to abuse DHCP.

A combination of the bash $RANDOM function[8] and the 'bc' free software package[9] (or a command-line tool such as macchanger[10]) would enable simple automation of this in order to generate a theoretically enormous random number of mac addresses, although the author simply chose to hardcode a small number of fixed mac address changes for the purposes of simplifying the demonstration. All three of these tools (bash, ifconfig, and dhclient) are available as part of the standard desktop package for any linux distribution and could be utilised using a rogue laptop, PDA, or even smartphone running linux. There no doubt exist tools for doing this on other platforms also.

Outwith the scope of the fairly standard tools mentioned, there also exist tools designed specifically to abuse DHCP (for malicious or legitimate purposes) which a coordinated attacker to use, such as *yersinia*[11], a network tool/framework designed to take

[8] http://www.tldp.org/LDP/abs/html/abs-guide.html#RANDOMVAR
[9] http://www.gnu.org/software/bc/bc.html
[10] http://www.alobbs.com/modules.php?op=modload&name=macc&file=index
[11] http://yersinia.sourceforge.net/

advantage of weaknesses in several protocols and designed for penetration testing, which supports attacks against DHCP.

Denial of Service, although an issue which is of great importance to business and home uses alike, is not even the most important problem which we face with the security issues surrounding DHCP. The single greatest threat which DHCP poses is centralised around the role which it has as the ‘entry-point’ of a new (or dynamically configured) client to a network. Since TCP, UDP, and ICMP are all dependant upon IP, DHCP (as the mechanism used to configure IP addressing and other information on host machines) is pivotal to the correct functionality of all of the above. Incorrect information handed out by a rogue DHCP server, as well as disrupting connectivity, could also be used to maintain connectivity (either in a semi or fully functional state) whilst redirecting network traffic to an alternative target to be recorded, analysed, and viewed. This could allow an attacker, for instance, to cause clients to attempt to authenticate against a fake Domain Controller controlled by the attacker.

At the talk in Dundee, the author demonstrated this by setting up a simple website on a demonstration laptop, and redirected traffic from a client by using a fake DNS server configured to resolve the address of '*www.example.com*' to an instance of the apache web server setup to quietly *reverse proxy*[12] to the real webserver, giving the appearance (after a DHCP lease was acquired from the bogus server) that the site was still functioning in exactly the same manner, whilst the traffic (and logon credentials) were trivially intercepted and stored by the attacking laptop using ethereal[13]. The ease of extraction of the http basic authentication credentials using a display filter, and (aside from ethereal) the use of fairly standard software (apache2, bind9, and thttpd) served to demonstrate quite how easy this manner of attack is to accomplish.

How this is affected by kerberos

Kerberos, as the most widely deployed Single Sign On mechanism, implemented as part of any Active Directory Infrastructure (and frequently implemented on other platforms as well), goes reasonably far to negate the damage done by this by using public key cryptography to use ‘tickets’ to authenticate to computers which are part of the domain without actually sending the client’s authentication information over the network, reducing the likelihood of interception and authenticating the actual host to which the traffic is being sent/received. However, even these kerberized services are not invulnerable to man in the middle attacks by remote or local intruders, and given recent leaps in both cryptography and processor power, once-impractically breakable encryption used for network services (such as weaker variants of SSL, liberally touted as “used by credit card companies” by sites proclaiming their security) are, by cryptographic standards, relatively trivial to break.

Kerberos vulnerabilities aside, it is not good security practice (or good Defence in Depth) to rely on Kerberos as our sole means of protection against this problem, and Kerberos does not protect hosts and services which either do not require the use of Kerberos or do not support it (or prevent Denial of Service attacks). In addition, companies which run hosts on their network (or run networks) which do not depend on kerberos (either using standalone hosts, older versions of the windows operating system, or other platforms) are even more at risk; workstations or servers running windows, gnu/linux, osx,

[12] http://www.apacheweek.com/features/reverseproxies

[13] http://www.ethereal.com/

bsd, embedded/network devices, or any of the varieties of unix are all affected just as much by this issue as windows is.

In a Windows 2000 or 2003 domain, clients use kerberos to securely request a ticket from a Ticket Granting Server to give them a means to communicate with a domain server; non-domain clients (or older clients), however, use challenge/response authentication to verify their password with the target machine – this challenge/response authentication mechanism is significantly more vulnerable to MITM attacks than kerberos, and so non-domain clients and non-windows clients which are not integrated into the kerberos domain are even more vulnerable, especially if group policy allows NTLMv1 and LM authentication[14]. For networks with computers not using kerberos, either relying on conventional challenge/response authentication or without any single sign on mechanism, these 'redirection' issues which we face with rogue DHCP server attacks are even more significant.

Other Potential solutions to the problem

So what do we do? There are already several well-written articles about DHCP security in Windows online[15], and although they do address some of these issues (such as some of the Denial of Service and Lease Exhaustion issues with DHCP), they do not specifically offer any solutions to this issue for us. Looking at the issue on the surface, windows already has one mechanism in place (the list of authorised DHCP servers) which could form a part of securing DHCP by dropping all DHCP traffic to clients on the host itself using a local firewalling policy (or a simple modification to the DHCP client code) which does not come from these servers. Unfortunately, without any form of verification or keying, this method would still be subject to attacks from spoofed (forged) replies, and we would have no way of safely acquiring a DHCP lease for the first time on an untrusted network.

But none of this is groundbreaking - these are all issues which have been considered before. RFC 3118, "*Authentication for DHCP Messages*"[16], penned in June 2001 (although apparently originally devised in 1995) and edited by staff from Cisco Systems and the University of Maryland's Departmeant of Computer Science, makes many of these points in its introduction, and proposes a well-conceived method for securing DHCP by the use of keys used to authenticate incoming and outgoing messages on the part of servers and clients. As an entity relying on cryptography already, Active Directory already has the infrastructure in place which would easy allow key distribution and re-keying of clients; the only issue remaining would be that of initial clients prior to their entry into the domain; this issue is easily overcome by the use of one-time keys for our clients – if a mechanism exists to re-key a client which is using a weak key, it would be a simple matter to randomly generate a weak, shortened key for manual entry as part of the setup process for a machine being joined to a domain on startup. Authenticating network access in this manner would ensure that at no point in a machine's lifetime would a host so trivially be exposed to a malicious DHCP server, from startup through the lifetime of the software on the host.

[14] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/gp/576.asp

[15] http://www.windowsecurity.com/articles/DHCP-Security-Part1.html, http://www.windowsecurity.com/articles/DHCP-Security-Part2.html

[16] http://www.faqs.org/rfcs/rfc3118.html

Even given the pre-existing standard and these security considerations, however, securing DHCP appears not to have been given great consideration as part of a network security strategy focused on corporate networks. The RFC designed to secure DHCP has ostensibly been ignored, and there is seemingly no implementation of this standard in existence.

The situation is not entirely dire; there are several off-the-shelf packages designed to address this problem. Unfortunately, it is generally the case that such systems fall foul of one or two problems which precludes their being used as a suitable fix for this problem. Some systems secure the DHCP server against unauthorised clients, but fail to secure the client against unauthorised DHCP servers, which (as we discussed earlier) is one of the most damaging potential attacks in a well-secured network.

Some of them are only workarounds, relying on the authenticity of the hardware (MAC) address of a client device in order to verify it[17]. Although providing security against a casual intruder, this approach is significantly inferior to many others because of the (relative) ease of altering a system's MAC address in order to bypass this measure by posing as another system. Consider the example given earlier in which the author altered this in order to execute a DHCP lease exhaustion attack. In Windows 2000/XP, the system MAC address can often, NIC-dependently, be relatively simply modified with local administrator access by changing the '*Locally Administered Address*' parameter in the Advanced Properties for a network adapter. Failing this, it can more generally be easily altered by editing the registry[18] or using third party software.

This system (MAC address filtering) can also be accomplished using the Windows DHCP Server package (or DHCP server packages on other platforms) by using MAC address reservations and then exempting (in Windows, by adding an *Exclusion Range*) unreserved addresses – in this manner, any devices that do not use a MAC address with a reservation will not be issued an address.

Other systems, which do address the problem effectively, are often too complex to be realistically deployed in most environments, and as this is a common problem it requires a solution accessible to the majority of affected networks.

(Mostly Microsoft's view of) IPSec

Microsoft's Server and Domain Isolation, mentioned in the prelude to our exploration of the DHCP issue, is one such package. The introduction to this paper goes on to say that this the IPSec strategy:

> *"allows IT administrators to restrict TCP/IP communications of domain members that are trusted computers."*

and that, further,

> *"network traffic can be authenticated, or authenticated and encrypted, in a variety of customizable scenarios."*

[17] http://www.metainfo.com/index.cfm/page/safedhcp
[18] http://www.nthelp.com/NT6/change_mac_w2k.htm

IPSec (whether deployed as part of Microsoft's 'Isolation' solution or more generally) is one way of securing an old protocol lacking in security, IP, and it enables Network Administrators to do several things.

Firstly, and most obviously, it authenticates and optionally encrypts data, increasing the integrity of the data, mitigating some of the risk posed by Man In The Middle attacks from non-internal users, and – most importantly for some – rendering it significantly harder to intercept and read (“sniff”) in-transit.

Secondly, as part of this implementation, it provides a form of distributed protection to a network of Windows computers as part of the same domain or forest, enabling communication with network hosts to be tightly controlled. Implementing IPSec in this manner, a Network Administrator may restrict which domain hosts may communicate with other hosts on the same subnet, and create a strategy motivated by network topology, company requirements, and security considerations. This ‘distributed firewall’ provides a powerful way for companies to control where data goes, and IPSec itself inherently provides a means to control how the data gets there.

Thirdly, it prevents clients who are not domain members (standalone machines, contractors, and machines running other operating systems which do not support or on which it is difficult to configure the IPSec system) from connecting to services at a level lower than simply preventing them from accessing domain resources, increasing the level of security provided by the security features in an Active Directory Domain both through the technologies domain computers employ for authentication and by ensuring that hosts allowed to connect to network services are properly managed. This sort of protection greatly enhances a network's resilience against unknown threats and exploits.

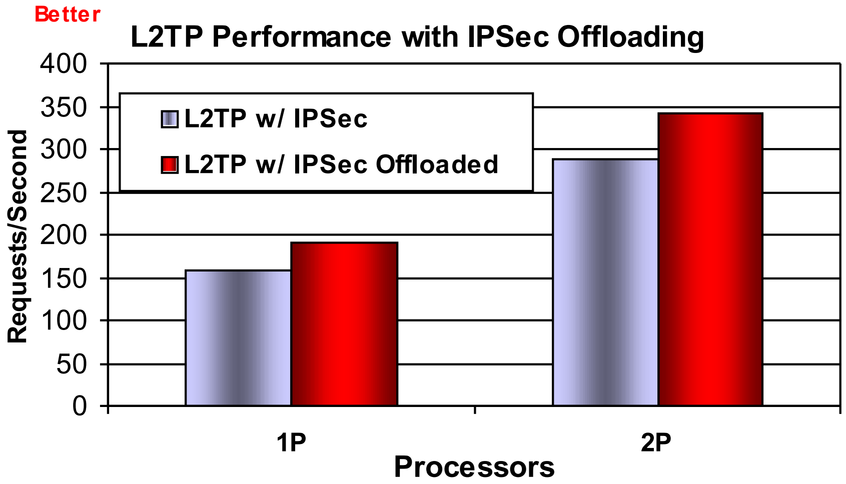
From the Introduction to Server and Domain Isolation:

"..server and domain isolation allows IT administrators to restrict TCP/IP communications of domain members that are trusted computers. These trusted computers can be configured to allow only incoming connections from other trusted computers, or a specific group of trusted computers. The access controls are centrally managed by using Active Directory® Group Policy to control network logon rights. Nearly all TCP/IP network connections are able to be secured without application changes, because Internet Protocol security (IPsec) works at the network layer below the application layer to provide authentication and per-packet, state-of-the-art security end-to-end between computers. Network traffic can be authenticated, or authenticated and encrypted, in a variety of customizable scenarios. The Group Policy and IPsec configurations are centrally managed in the Active Directory."

IPSec isolation of this type, however, has drawbacks; it is hard to diagnose and troubleshoot, and difficult to implement without a test environment and skilled IT staff (that is to say, difficult to implement for companies which are medium or smaller businesses and even some which aren't). IPSec also requires a considerably higher specification set of servers in order to fully encrypt traffic, due to the increased processor load where the encryption is not offloaded to a NIC or other piece of specialised hardware which performs this task in place of the system CPU. This is particularly the case for computers such as Domain Controllers which frequently carry out many transactions with client computers in parallel.

Most importantly for the purposes of this discussion, windows as a desktop operating system will not acquire IP addresses via DHCP over IPSec. Unfortunately, the implementation of IPSec in Windows 2000 and XP only allows IP addresses to be acquired over IPSec via DHCP using the L2TP protocol (ie. as part of a point-to-endpoint VPN configuration). It is generally in this configuration that DHCP over IPSec is considered[19], rather than the use of DHCP for configuration of a local network.

A complete discussion of this issue is outside the scope and intentions of this document, but although IPSec benchmarking statistics for Windows were initially difficult to find, *'Windows 2000 Performance: an Overview'*, published on technet in 2000, benchmarks the Windows 2000 and NT4 pptp and l2tp performance, and includes the following comparison of L2TP/IPSec performance with and without a NIC supporting Ipsec Offloading, which gives us approximate figures for the load of IPSec:



Copyright 2000 Microsoft Corporation, taken from '*Windows 2000 Performance – an Overview'*[20], for the purposes of commentary as fair use.

A benchmark in linuxjournal of the IPSec support in the 2.6-series linux kernel using the 3DES algorithm (a very processor-intensive block cipher for these purposes compared to faster, stronger candidates for the job, such as Rijndael/AES) gives slightly more conservative figures for throughput for IPSec Traffic:

Packet Size	Bandwidth without IPsec	Bandwidth with IPsec
1KB	10905KB	5157KB
2KB	10832KB	5222KB
4KB	10827KB	5305 KB
8KB	10811KB	5263KB
16KB	10814KB	5345KB
32KB	10729KB	5374KB

[19] http://www.faqs.org/rfcs/rfc3456.html

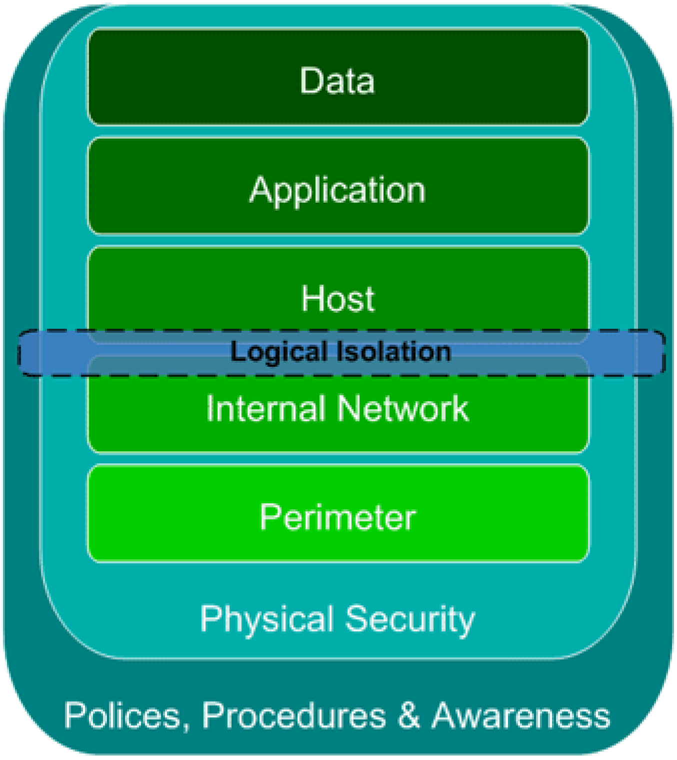
[20] http://www.microsoft.com/windows2000/server/evaluation/performance/overview.asp

Table 1. Results from Netio TCP Benchmark, Vincent Roy, '*Benchmarks for Native IPsec in the 2.6 Kernel*[21]*'* © 2004 Linux Journal, for the purposes of commentary as fair use.

This comparison is not designed to benchmark or compare IPSec performance in the two families of Operating System (and the two examples given almost certainly don't use the same algorithm, making these statistics irrelevant for benchmarking), but rather to demonstrate that IPSec slowdown on the same hardware is inevitable and that this is not a problem specific to either platform.

This issue aside, it is also inevitable that IPSec will also only provide security for encrypted and encapsulated traffic – unmanaged or incompatible computers will be excluded from the protection, potentially causing standards to have to be compromised as hosts on the network are obliged to accept non-IpSec communications.

Especially in conjunction with the problem mentioned above (that of unmanaged or older computers), IPSec has other requirements which require us not to rely on it as our sole network security measure. IPSec cannot prevent attacks which occur at a lower level than it exists at (IPSec cannot secure IP, only the TCP connections which sit above IP), and it does not protect against compromise by trusted hosts on the network; any host using IPSec which is broken into or used maliciously can still compromise another machine (such as an application server or domain controller) via IPSec. The following diagram positions IPSec with reference to five layers discussed as part of Microsoft's Defence in Depth strategy, and comes from the IPSec Server and Domain Isolation guide:



[21] http://www.linuxjournal.com/article/7840

Copyright March 2005 Microsoft Corporation, taken from '*Server and Domain Isolation using IPSec and Group Policy*'[22], for the purposes of commentary as fair use.

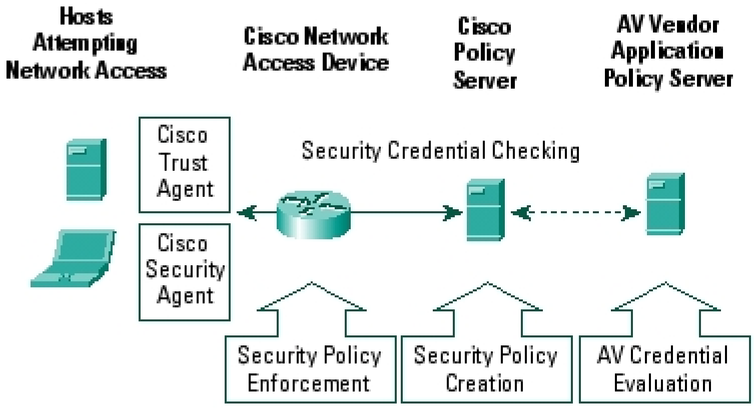
As we can see, this IPSec strategy does not prevent attacks via the Internal Network on hosts which accept non-IPSec traffic, and nor does it prevent attacks on the IP stack on the target machine, or issues with '*Internal Network*' technologies such as DHCP and ARP. IPSec also fails to provide a significant amount of protection for our Infrastructure against malicious clients which are domain members and thus negate the 'Isolation', as these are unable to be firewalled against clients which have legitimate uses for them (eg. Host□LDAP server or Host□HTTP application server). Unless our 'distributed firewall' policy is very complex, it is also quite possible that domain members will still be able to connect (via IPSec) to services (such as RDP) on infrastructure and application servers or other domain members which are unnecessary and present an additional Intrusion Vector.

These protocols which cannot be encrypted using IPSec in transport mode, such as ARP and DHCP, are our concern here. Attacks using ARP are a known occurrence[23], and there are several software packages for several operating systems which will both execute and detect such attacks, in addition to features built into modern switches in order to attempt to prevent attacks such as these.

Cisco NAC

Microsoft's solution to this aside, Cisco offer a package called NAC (Network Admission Control[24]) which:

"can provide network access to endpoint devices, such as PCs, PDAs, and servers that fully comply with established security policy. Cisco NAC allows noncompliant devices to be denied access, placed in a quarantined area, or given restricted access to computing resources.".


Hosts Attempting Network Access
Cisco Network Access Device
Cisco Policy Server
AV Vendor Application Policy Server
Cisco Trust Agent
Cisco Security Agent
Security Credential Checking
Security Policy Enforcement
Security Policy Creation
AV Credential Evaluation

Copyright © 1992--2004 Cisco Systems, Inc, taken from "*Cisco NAC - The Development of the Self-Defending Network*" for the purposes of commentary as fair use.

[22] http://www.microsoft.com/technet/security/topics/architectureanddesign/ipsec/default.mspx
[23]http://en.wikipedia.org/wiki/ARP_spoofing, http://node99.org/projects/arpspoof/arpspoof.pdf
[24]http://www.cisco.com/warp/public/cc/so/neso/sqso/csdni_wp.htm

Although extremely effective, the NAC system involves interaction of a number of different (Cisco) systems, and for an organisation which already had incompatible or non-Cisco network hardware in place, migration to this system would be more complicated than simply installing client software and configuring a management agent/server. As this is a software problem and not fundamentally an issue with our hardware, it seems unnecessary to (expensively) fix a software problem by replacing hardware which has no other faults.

The Cisco system is, in many ways, a similar system to Microsoft's *Server and Domain Isolation* strategy – it sets out a plan for a 'next generation' philosophy of network security, which raises the bar on what we can accomplish even on smaller networks. The differing approaches of the two systems, although not identical, are understandable given the two companies positions being thought of as, respectively, predominantly a hardware and software vendor. In October of 2004, the two vendors even agreed to work together on these approaches[25], although this author has yet to see any concrete results of this.

IPv6

IPv6, the successor to the present version of the Internet Protocol (IPv4), reformulates the way in which ARP and DHCP work. Unfortunately, no inherent security has been added onto the protocols which provide the functionality presently provided by DHCP and ARP with IPv4. IPv6 introduces a feature known as *Stateless Autoconfiguration*,[26] which although not as extensible and powerful as DHCP, fulfils the same purpose in main environments. The RFC has the following to say about Stateless Autoconfiguration:

> *IPv6 defines both a stateful and stateless address autoconfiguration mechanism. Stateless autoconfiguration requires no manual configuration of hosts, minimal (if any) configuration of routers, and no additional servers. The stateless mechanism allows a host to generate its own addresses using a combination of locally available information and information advertised by routers. Routers advertise prefixes that identify the subnet(s) associated with a link, while hosts generate an "interface token" that uniquely identifies an interface on a subnet. An address is formed by combining the two. In the absence of routers, a host can only generate link-local addresses. However, link-local addresses are sufficient for allowing communication among nodes attached to the same link.*

Stateless autoconfiguration incorporates no security mechanisms, and as such is vulnerable to spoofing attacks in the same way that DHCP in IPv4 is. The alternative to this *stateless* configuration mechanism (the *stateful* configuration mechanism) is DHCPv6[27]. A paper by Sean Convery and Darrin Miller at Cisco Systems entitled "*IPv6 and IPv4 Threat Comparison and Best-Practice Evaluation (v1.0)*"[28] states that:

> "*Although DHCPv6 is investigating security options, the protocol is too new to be considered in this paper. At a minimum the approaches used for protecting DHCP in IPv4 networks should be implemented for IPv6.*"

[25] http://www.theregister.co.uk/2004/10/18/cisco_dewormer_alliance/
[26] http://www.faqs.org/rfcs/rfc1971.html, http://www.faqs.org/rfcs/rfc2462.html
[27] http://www.faqs.org/rfcs/rfc3315.html
[28] http://www.cisco.com/security_services/ciag/documents/v6-v4-threats.pdf

RFC3315, published in 2003, does codify some of these security options, and has an optional Authentication scheme which may be used by clients. Page 65 of this RFC describes this behaviour as follows:

```
 Client behavior, if no Advertise messages include authentication
information or pass the validation test, is controlled by local
policy on the client.  According to client policy, the client MAY
choose to respond to an Advertise message that has not been
authenticated.
```

The uptake of this scheme remains to be seen, but at least if implemented properly (and with the proper degree of vendor support from our directory services, desktop platform, and infrastructure vendors!), this should provide a strong option for medium and larger enterprises for whom these issues present notable security risks to implement.

In Conclusion

Although IPv4 is dying, it is extremely safe to say that it is dying slowly, and it's death pangs will be long and complex. For the time being, DHCPv4 security concerns are still important. At the very least, the mistakes which have been made in deploying and supporting DHCP in IPv4 networks are important, and without an acknowledgement of these mistakes, it seems likely that the '*optional*' keying mechanism in IPv6 may befall the same state as our old friend, RFC3118.

As a final thought, going back to the software solution for this, the RFC designed for this purpose would not even, technically, need to form a part of a keyed DHCP system integrated with Active Directory and the Windows Domain – The DHCP Specification allows for Vendor Extensions to DHCP (RFC 2132 - "*DHCP Options and BOOTP Vendor Extensions*"[29]) with option code 43.

Use of this field to authenticate servers for clients would even provide a mechanism by which the *same* DHCP server could service both authenticating and non-authenticating clients, as the Vendor Extensions should be dropped gracefully by clients which did not talk the authentication scheme; non-domain windows clients and non-windows clients would therefore be just as home with this system, unlike the full authentication scheme proposed by RFC 3118, which sadly lacks backwards compatibility.

As a few conclusive bullet points summarising the author's points:

- DHCP is a widely-deployed but insecure, unauthenticated configuration mechanism designed with no security safeguards.
- Safeguarding the lower levels of our networks is increasingly important given the threats posed by misconfigured workstations, malicious employees, contractors, and wireless networks.
- DHCP (although insecure) is very rarely considered as a security risk on networks, in spite of the wide array of intrusion vectors it opens into and disruptions it poses to our client and server systems.

[29] http://www.faqs.org/rfcs/rfc2132.html

- Systems which address DHCP are either plainly inadequate, do not specifically address the DHCP problem (IPSec), or are too high-end for the needs of most networks (Cisco NAC).
- More holistic, practical security measures for DHCP have been considered in the past, but there are presently virtually no measures integrated into the operating system and very few third party mechanisms accessible for businesses designed to secure it.
- DHCP deserves more love!