Abstract- Many enterprise sites utilise 802.11b/g technology to create an untrusted access network sitting outside their protected institutional IP network, with internal access allowed only through an IP-layer virtual private network (VPN) gateway. Often such networks do not implement link layer security, because of the known weaknesses of the IEEE's wired equivalent privacy (WEP). This results in a wireless network on which arbitrary people can establish themselves as hosts with arbitrary IP addresses. Although the enterprise IP network is protected by the VPN gateway, users of the wireless network can become victims of unscrupulous (or accidental) interception of their IP communication. Common Windows laptop (mis-)configurations often try and establish communications through a default gateway on the 192.168/16 network. Anyone could configure another host as this default gateway on the enterprise 802.11b/g network and thus hijack a visitor's network connection without the visitor even realising. In this paper we test and confirm the plausibility of this attack in a University wireless LAN and present results from real world data, confirming the existence of users failing to reconfigure their visiting host and attempting to connect via possible malicious gateways. We then suggest possible mitigation techniques.

I. INTRODUCTION

Although the IEEE's 802.11b and 802.11g wireless LAN technologies have become popular and flexible methods of providing network access around enterprise sites, they bring a number of security and performance issues. A fundamental security issue is that in basic form, 802.11b/g (a term we will use to refer to either 802.11b or 802.11g) sends LAN frames openly through the air, able to be intercepted by anyone within radio range of an 802.11b/g access point. On the performance front, 802.11b networks have been shown to suffer dramatic throughput degradation in the presence of low bandwidth, high frame-per-second traffic [1].

The IEEE initially specified an encryption protocol, known as Wired Equivalent Privacy (WEP), to combat the security problem. However, WEP has two fundamental weaknesses relevant to enterprise (and in particular, university campus) environments. First, vulnerabilities were discovered in WEP's encryption system [2], effectively rendering WEP pointless, if its secret encryption keys were not changed frequently. Second, WEP required every client to share the same encryption key and keep that key secret. This assumption is trivially proven untenable in university campus environments – once the key is handed out to a few thousand students it should be assumed to be no longer secret.

Because of these WEP issues, many institutions chose to implement their network security using IP-layer virtual private network (VPN) technology. The wireless network is considered to be 'outside' the institution's own IP network, accessible only by people who can authenticate with the VPN gateway. Consequently there is nothing preventing any individual (within wireless range) from attaching to the 802.11b/g network and interacting, at the link layer level, with other clients on the network. Generally, this is not going to be a problem for regular users who successfully create VPN connections using the wireless network. However, we have identified and evaluated a possible vulnerability for people first connecting their laptops to such a wireless network.

The vulnerability arises from the way in which many people use their wireless-equipped laptop at a number of locations during the day. In the most simple case a nomadic user shifts their laptop from home, to work and back home again each day. For the sake of discussion we will assume work is a university with a campus-wide 802.11b/g network. At each site, the laptop's owner may have statically configured the necessary IP addresses (at minimum, local host address, netmask and gateway) or specify that the laptop use DHCP to auto configure necessary IP settings. Unless both home and university networks support the use of DHCP, it is quite likely a laptop will initially appear on the university wireless network using its statically configured home network IP host and gateway addresses.

A key marker of such a laptop, is the sight of address resolution protocol (ARP) packets on the wireless LAN seeking mappings for common 'private gateway' addresses, such as 192.168.0.1 or 192.168.1.1. These addresses are commonly used by home broadband routers and gateways and so tend to be configured into laptops when used at home. Preliminary inspection of our own university wireless network, revealed a scattering of hosts exhibiting precisely this behaviour. (ARP requests are typically broadcast at the link layer, so they are visible at all 802.11b/g access points making up the bridged ethernet service of a typical campus wireless network.)

The possible security problem arises as follows: Most people don't usually notice a misconfiguration of their
laptop's network interface, unless they're unable to 'get on the network'. If the university's wireless network does not itself use 192.168/16 private address space [3], the misconfigured laptop will be 'off the air', eventually triggering manual reconfiguration by the user. However, consider a third party attached to the wireless network, who decides to pretend (by replying to the ARP requests) that they are indeed a gateway at 192.168.0.1 (or 192.168.1.1, perhaps even concurrently). This third party, equipped with their own separate IP connectivity to the Internet (or possibly back into the university network), could ensure the laptop immediately receives connectivity via the original gateway address. The laptop would thus begin sending and receiving all its traffic via this third party node, rather than the university network's own router. The potential for mischief becomes fairly clear, as the laptop's owner never realises their traffic is being redirected and intercepted. (In the extreme case, the laptop owner forgets to initiate the VPN connection, thus sending all their data through the third party node unencrypted.)

Figure 1 shows an example of this scenario. A hypothetical host 192.168.0.99 wirelessly attaches to the university network, ARP's for its previous home gateway at 192.168.0.1 and is provided unofficial network connectivity by a (possibly un-) friendly host claiming to be 192.168.0.1 on the wireless network.

It is important to note that the host masquerading as 192.168.0.1 does not need to be physically on-campus. They need only be within wireless range of a campus 802.11b/g access point.

It is possible for this redirection to occur accidentally, as a side-effect of the 'internet connection sharing' (ICS) functionality built into Microsoft Windows 2000 and XP. ICS implements network address port translation, (NAPT) to map private IP address space into regular routable IP address space. Through human error in misconfiguration, a laptop simultaneously connected to the university's wired and wireless networks can be told to "share" the wired network connection onto the wireless network. This not only opens an instant side-effect of the 'internet connection sharing' (ICS) functionality built into Microsoft Windows 2000 and XP. ICS implements network address port translation, (NAPT) to map private IP address space into regular routable IP address space. Through human error in misconfiguration, a laptop simultaneously connected to the university's wired and wireless networks can be told to "share" the wired network connection onto the wireless network. This not only opens an instant interception attack upon our network. Results from tests are presented in section IV, before we note some possible mitigation techniques in section V and conclude in section VI.

II. Experimental Background

Recently our university undertook the roll out of 802.11b (and later 802.11b/g) wireless infrastructure to cover large geographical portions of our multi-campus university. This roll out was to allow flexible access to staff and students of network resources.

Both staff and students are able to access the university network, with either private laptops brought onto campus or using a pool of laptops provided for short term loan by the library.

A. Normal network access

Our university wireless network utilises approximately 300 Access Points (APs) and spans multiple campuses. As well as access from any AP, access to the “wireless subnet” can be achieved from multiple public access ethernet ports in one campus library. While wireless coverage is still available within this library, the physical access points are provided for laptops lacking 802.11b/g network interfaces.

Normal network access involves associating to an access point or in the case of wired access, simply plugging in. DHCP then provides IP connectivity to the wireless subnet and the VPN client/server solution from a commercial vendor.
allows secure access to the wider university network and by extension, the Internet.

No layer 2 security (such as 802.11i [4] or Cisco's LEAP [5]) is implemented.

III. EXPERIMENTAL METHODOLOGY

To test for the aforementioned issue on our network, without actually intercepting the traffic of staff and students, we undertook a short controlled test. A desktop workstation with two network interfaces was configured using the FreeBSD[6] operating system. One network interface was an 802.11b connection onto the wireless network, the other a connection onto the campus wired ethernet network. The machine was configured to allow NAPT access from the wireless interface to the wired interface. The (uncommonly used) private IP address of 10.99.88.1 was configured as the wireless address and DHCP was used to gain IP address details from the campus network on the wired interface.

A laptop enabled with an 802.11b card was configured with the IP address of 10.99.88.100/24 and had its default gateway set to the address of the gateway machine, 10.99.88.1. The laptop was able to gain access to the campus network and the Internet via this connection.

On the gateway machine all traffic to and from the laptop could be seen at the packet level (using tcpdump [7]). Tools such as dsniff [8] could have easily allowed unencrypted application layer data, such as mail passwords and visited URLs, to be extracted.

After confirming the problem could be successfully exploited on our network, we undertook to verify whether the vulnerability had any operational relevance.

IV. RESULTS

We used network packet capture to infer the number of individuals who were potential unwitting victims of this interception technique. A FreeBSD host was connected to the wireless network and recorded all broadcast traffic (again using tcpdump) from February 16, 2005 to May 2, 2005, collecting almost one gigabyte of raw packet data.

A. General usage patterns

During the collection period, there were 1,871 individual MAC addresses used to access the wireless network. Figure 3 shows the general usage patterns of the wireless network based on ARPs for the network's legitimate default gateway (136.186.49.1). Usage rises sharply in the morning, holding relatively steady into the day and dropping linearly until the early morning.

B. Discovering Potential Victims

Figure 4 shows the ARP requests for private address space. This traffic is primarily hosts checking for collisions on their own previously configured IP address, or seeking out their previously configured default gateway. (The distribution of least significant octet values in the ARP requests showed clustering around addresses of the form 192.168.0.1, 192.168.0.2 and 192.168.0.254. The former and latter are common default gateways, whereas 192.168.0.2 is likely to be the address most commonly assigned to the laptops when they were last connected to their home network.)

Interestingly, the ARP traffic for private addresses is strongly clustered around a few hours in the middle of each day. This suggests that interception of unsuspecting laptop users is most profitably performed during the day, so need not be pursued out of hours.

To gain a further insight into how many individuals could fall victim to the interception we've described, we count the number of people who first attempt to ARP for private address space and then shortly after reconfigure and request a DHCP address from the wireless network. We treat this behavior as indicative of someone who really does want to use IP networking services from their laptop, and reconfigures their wireless network interface when they discover lack of network connectivity.

We define the “Time to Reconfigure” (TTR) as the interval between first ARPing for a private address and then issuing a broadcast DHCP query. We correlate ARP and DHCP requests through the client laptop's source ethernet MAC address.

---

![Figure 3](image3.jpg) Wireless network daily usage pattern, based on ARP requests

![Figure 4](image4.jpg) Attempts to ARP for private address space
Our measured TTR values should also be considered a lower bound as it is possible (arguably most likely) that these individuals are also repeat offenders. Our analysis currently only calculates TTR the first time a MAC address is seen going through the ARP/DHCP reconfiguration process.

During the TTR analysis, 174 individuals were seen. Eight outliers have been removed from the data set. In these cases, hosts attempted to ARP for private address space but failed to reconfigure and request DHCP information. Days later they returned with DHCP enabled, giving TTR results measured in days. Their initial behaviour suggests they did not wish IP connectivity and therefore do not count as potential victims.

Similarly, hosts with TTR under 10 seconds represent probable, but inexplicable, automatic reconfigurations and hence will most likely not be vulnerable to interception, 67 reconfigurations fell into this category and were discarded.

We feel the remaining 99 instances, (0.05% of the 1,871 total MAC addresses seen) would have been a situation where a user accessed an illegitimate default gateway and been open to communication interception and interference.

Figure 5 Shows 70% of users reconfigured within 600 seconds (10 minutes) of attempting to access private address space. If malicious configuration of the requested gateway occurs within 30 seconds (a time easily achievable when assuming even manual configuration) 86% of our 99 users could have been entrapped.

Of interest was the relatively even distribution of reconfiguration events throughout the semester, revealed by Figure 6. No large spikes of misconfiguration events were seen at the beginning of the semester, as may have been predicted.

V. MITIGATION

A number of steps might be taken to reduce the impact of the interception technique we have outlined, despite the limited physical layer and link layer security of typical campus 802.11b/g deployments. We believe the following steps can be achieved at low cost, using easily obtained PC hardware and open source operating systems (such as FreeBSD or a Linux variant). Our focus is to ensure that typical private gateway addresses (192.168.0.1, etc) are not mapped to rogue or unofficial devices on the campus network.

One approach would be for the entire campus wireless network to itself utilise the 192.168/16 private address space, thereby ensuring that the 'real' campus router can be located at 192.168.0.1 (and equivalents). The use of private address space between the host and the official VPN gateway does not impact on the host's ability to communicate with the outside world, since the VPN service itself typically overlays an additional, publicly routable IP address to each VPN client.

If the campus wireless network is built around publicly routable IP address space, (or private space outside the 192.168/16 range) then we must proactively intercept misconfigured laptops ourselves and/or constantly monitor the network for other hosts masquerading as 192.168.0.1 (etc).

There are two approaches here, both utilising a low cost 'mitigation' PC permanently connected to the campus wireless network. First, our mitigation PC should itself intercept the traffic of misconfigured laptops by pretending to be the router at 192.168.0.1. However, rather than routing subsequent IP packets to the real campus network, all except web traffic would be blackholed (dropped). Outbound web traffic would be redirected to a local web server that returns a warning page, letting the laptop's owner know they need to reconfigure their machine for proper and secure network access. In this manner the laptop's owner is both protected and informed.

Second, the mitigation PC should also be programmed to regularly issue its own ARP requests for well-known private gateway addresses in the 192.168/16 space, seeking out hosts that may be innocent (or not so innocently) misconfigured. (We recommend using an open-source unix-variant for the mitigation host. For example, under FreeBSD it is trivial to alias multiple IP addresses on to a single network interface, allowing the mitigation PC to concurrently intercept a number of possible private gateway addresses).

VI. CONCLUSION

Given the inability of Wired Equivalent Privacy (WEP) security to provide many institutions with a proper form of
wireless security, many move to a IP layer only security model for their wireless networks. This leaves their clients vulnerable to an interception attack that arises from the way in which many people use their wireless-equipped laptops. When a laptop associates with a campus network after having previously been configured to use a private address space gateway (for example, from their home LAN), it is possible for a third party to intercept the user's network communications by pretending to be the sought-after private gateway.

We proved that this attack is technically possible, and showed that it is more than theoretically interesting. A small but finite community of wireless network users (99 clients over three and half months) was shown to exist who would be vulnerable to the interception attack. The typical vulnerable client took between 10 and 300 seconds to realise their laptop was misconfigured and to then establish 'correct' IP connectivity with the legitimate gateway on the campus wireless network. This period provides ample time for a third party to intercept and hijack the IP connectivity of such users.

Finally, we suggest some relatively simple mitigation techniques, including proactively intercepting such misconfigured laptops with a 'good' gateway that redirects all subsequent web traffic to a page advising the laptop's owner how to correctly reconfigure their machine. Such a gateway/webserver can be trivially established using cheap PC hardware and an open-source unix-compatible operating systems such as FreeBSD or Linux.

REFERENCES