

Countering Rogues in Wireless Networks *

Austin Godber and Partha Dasgupta
Department of Computer Science and Engineering
Arizona State University
Tempe, AZ
{godber, partha}@asu.edu

Abstract

Wired networks are prone to the same attacks as wireless ones, including sniffing, spoofing and Man-in-the-middle attacks (MITM). In this paper we show how wireless networks are particularly vulnerable to a simple MITM that can make even rudimentary web surfing dangerous. We describe how we performed the attack and its ramifications. We argue why it is essential to have a VPN tunnel from the client to some trusted host (not access point) in order to avoid being compromised.

1 Introduction

With the explosive popularity of wireless networks, based on the 802.11b and its descendant technologies, have come great benefits and great risks. The benefits include reduced cost of deploying additional networked devices due to the inexpensive network devices and no longer requiring expensive wiring. These networks also have the ability to extend institution networks over many miles at a reasonable price. These benefits have made it enormously popular within practically every computing community, be it corporate, educational, or private home users.

The installed base of 802.11b devices that flourished from the availability of inexpensive consumer grade equipment ensures that the risks of 802.11b will continue to pose a threat for a very long time. So, despite the development of new networking technologies that can perhaps solve these problems, other solutions that work with the existing technologies, should be explored.

At first glance, a wireless network seems to have tremendous and obvious risks. The risks to WiFi clients include eavesdropping with the intent of learning what the client is doing or with the intent of intercepting authentication in-

formation. In addition to eavesdropping, wireless networks are prone to jamming, spoofing, rogue access points, and possible Man-in-the-middle attacks.

However, at a second look, it seems that the same problems are also present in wired networks. For example, a rogue access point might be seen as a threat to the network administrator, but not a threat to the clients on the network. Hence, wireless security problems can be solved by all the known solutions to the security problems in wired networks.

The above claim, however, is not true. We have discovered a particular Man-in-the-middle attack that is invitingly easy to implement on wireless network, and is not so easy to do on wired network. In addition, we show why eavesdropping is much more of a threat in wireless networks. The solution to these problems is the use of encryption. However, in this paper we show why encryption between a wireless client and a base station is not effective (even if proper, strong encryption is used).

Since this new class of threats targets the client, it falls outside the common logic for combating Rogue APs. The common concern for Rogue APs is that they provide unauthorized access to internal corporate networks; and if an AP is not connected to the internal network, it is not a threat.

1.1 Privacy in wireless and wired networks

In this section, we first discuss eavesdropping. In a wired network eavesdropping is possible, but in most cases it is not practical. In most corporate networks, clients are connected to switches and hence the traffic between the client and the network is not readily visible to other clients. The switches feed traffic into administered routers, which eventually lead to border routers into some backbone network. Here, to snoop, the routers have to be reprogrammed (difficult) and the amount of traffic flowing through them makes the overhead of sniffing and filtering prohibitive.

*This work is partially supported by DARPA/Spawar, AFOSR and NSF.

For home users or small businesses, Internet service is provided by dial-up, cable-modem, or DSL. All of these are not prone to sniffing by end users. Again, the traffic terminates within the ISP's administered networks and most ISPs have other, more significant tasks to perform than to sniff data traffic.

Wireless networks allow clients to sniff other people's packets. Outsiders can sniff traffic on corporate networks. It seems an encrypted channel between the client and access point solves the problem. Turns out, as described later in the paper, it does not. We need an encrypted channel between a client and an authenticated trusted server on a wired network to achieve the same level of privacy that unencrypted wired networks provide.

1.2 Man-in-the-middle

The Man-in-the-middle (MITM) attack is possible in both wired and wireless networks. In a wired network, one either needs to spoof DNS requests or ARP requests or compromise a valid gateway machine to obtain access to the clients traffic. In a wireless network, since there is no authentication of the network or the client is haphazardly using an untrusted hotspot, the MITM is relatively simple. The attacker connects (using wireless) to a wireless network. He then provides service to other clients with another access point that has the same SSID as the host network. Clients associate with the attackers "rogue" access point and traffic is routed through the attackers router. The attacker can not only sniff, but can actually change the traffic, insert viruses into downloaded files, change web pages, and use known vulnerabilities in browser scripting to attack the client machines when they visit well known web pages. This attack is a particularly nefarious and easy to perform, even on sophisticated users. In addition, the encryption between the access point (rogue) and client does not protect the client.

Most web access is not encrypted as there seems to be no private information worth protecting (except the information about which sites were visited). However the MITM attack on public web pages creates a new vulnerability. Using a public wireless network to access cnn.com, can cause a client to be exposed to attacks. Downloading a program from a known download site can be terribly dangerous.

Thus, we argue that wireless networks are inherently at higher risk than their wired counterparts. Our solution to this problem is to require the wireless client to VPN all traffic (not just "sensitive" or corporate traffic) to a trusted wired network. By utilizing a secure VPN solution, the client is no longer at risk from malicious network attacks on the wireless segment. Also, the termination point of the VPN must be carefully chosen and authenticated.

1.3 802.11b Wireless Network Configurations

We can classify wireless networks into two different classes of wireless network scenarios: large institutional networks and small individual wireless hotspots. Even though the risks are generally the same over both deployments, it may be useful to consider them separately.

1.3.1 Rogue Access Points

A rogue access point is an access point deployed on a large centrally administered network outside the administrative controls established for the authorized wireless access points. Without mutual authentication to the network, clients could inadvertently connect to one of these Rogue APs and thus be at risk.

There are precautions that can be taken by the network administrators to detect and prevent these Rogue APs, however, only securing this network will not resolve a clients vulnerabilities in other networks. A client compromised elsewhere could then return to the secured institutional wireless network and create additional an additional threat.

1.3.2 Hostile Hotspots

A Hostile Hotspot is a wireless hotspot, or a public wireless Internet point of presence where the owner or administrator of that hotspot has malicious intentions and tampers with the traffic it handles. Casual use of such hotspots puts clients and their home networks at risk. These networks are the real risk to wireless users whose home network has deployed an effective local security solution.

2 Related Work

Much work has gone into securing wireless networks and wireless clients. The majority of the work has focused on developing methods to prevent unauthorized access to the wireless network. Such efforts would include improving media access control and detecting and preventing rogue access points. We shall outline some of these solutions.

2.1 WEP and MAC Filtering

The first attempts at providing some security to WiFi networks were WEP and MAC Address Filtering. WEP utilizes the RC4 stream cipher and manual key distribution to provide data confidentiality and integrity. WEP's weaknesses have long been legendary [3, 11]. It is argued that it is better than nothing and at least prevents casual attacks. However, in the attack scenarios we present here it provides no protection what so ever.

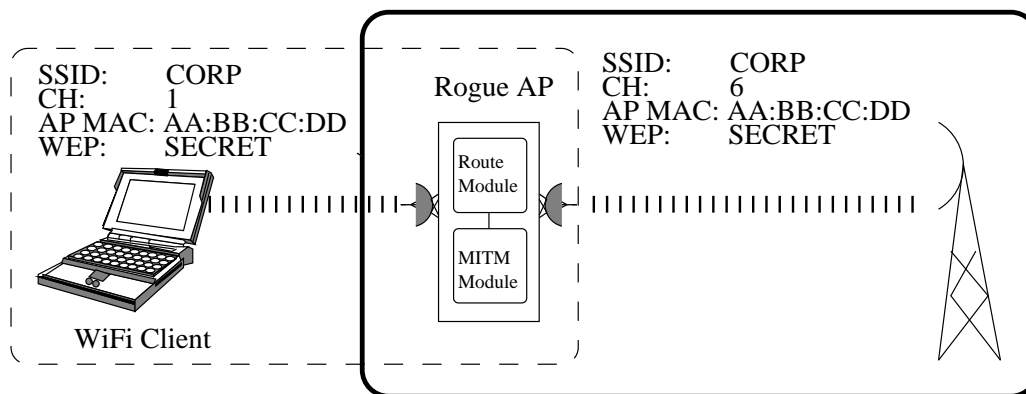


Figure 1. Example MITM Configuration.

MAC Address filtering is the attempt to restrict access to the wireless network by only allowing certain MAC addresses to connect. Since MAC addresses can be changed from their factory default and valid MACs can be sniffed from the network it accomplishes nothing more than perhaps keeping honest people honest.

2.2 802.1x

Soon after the massive adoption of 802.11b, a new and improved security mechanism was introduced, 802.1x[5]. This mechanism made modifications to the clients, APs and added an authentication server that would allow clients to authenticate to the network. The protocol was made to be general enough so that many different authentication protocols could be utilized as seen fit by those deploying the network.

This newer solution is not without its flaws either [9]; in fact, it suffers from the same fundamental flaw that 802.11b suffers from: there is no authentication of the network. Without this mutual authentication, there is no guarantee that the client connects to the desired network and thus cannot trust the AP it connects to.

802.1x and TKIP, which amounts to an improved version of WEP, have been packaged into a new security solution called WiFi Protected Access (WPA). This interim solution addresses client access to the network and WEP's previous vulnerabilities. TKIP still relies on a pre shared key, thus is still vulnerable to MITM attack from valid network clients. The latest standard, 802.11i, also leverages these solutions but is supposed to add secure deauthentication and disassociation among other things. This standard is still in the draft phase and is expected out in the end of 2003.

2.3 Detecting Rogue Access Points

There are recommended standard practices for deploying a wireless network infrastructure in an institutional setting. Among the recommended practices is monitoring both your wired and wireless networks for indications of Rogue Access Points. Good record keeping and doing radio site audits will help detect these rogues. These techniques rely on monitoring 802.11b Sequence Control numbers. Depending on your deployment scenario, monitoring the traffic on the wired LAN can also aid in detection of Rogue APs.

2.4 Other Security Solutions

There have been other attempts to address these security problems. Some from Academia [1, 2, 4] and some from the open source community[13, 10].

These and many more solutions arose out of the particular needs of the groups designing them, all with the intent of improving security. However, in the case of highly mobile networks where the clients have a large degree of network promiscuity, a partial fix, or fix at home, will not solve the problem. A client from an entity that has deployed and ultra secure local solution will not benefit from that solution when away from the home network.

3 Vulnerabilities

As mentioned earlier wireless networks suffer from the same vulnerabilities as wired networks. Both can experience the same hodgepodge of IP based and higher vulnerabilities. The difference begins at the Data Link Layer and the inherent broadcast nature of the wireless physical layer, which doesn't benefit from the restricted physical access of traditional wired networks.

3.1 Mutual Authentication

The first fundamental difference is that wireless networks are a broadcast media with no effective mechanism of media access control. This design and several solutions target their effort at authenticating the user without guaranteeing that the client is connecting to the network he desires.

Technically, wired networks also do not provide mutual authentication, however, wired networks (not being broadcast through the air) benefit from the physical security of the network jacks that connect to it.

3.2 Network Promiscuity

This important difference is where trouble begins and the new class of vulnerabilities become apparent. In the past, network connectivity was static. A computer sat on a desk at one location connected either to a corporate network or to an ISP where accountability and reliability of the provider was important to that provider. The provider had an incentive to protect both his own network and the client computers that utilized it. The incentive to protect the client computer comes from knowing that if it is compromised, that would increase the chance of the provider being compromised.

With wireless networking, things have changed. The inherent mobility of wireless clients and lack of network authentication have given rise to an age of high risk network usage, a type of *network promiscuity* if you will. Mobility implies that a computer will move between administrative domains. Each of these domains will have different goals and levels of administrative competence. In the wired scenario, the network provider had incentive to protect the client thus his network, this is no longer necessarily the case. Since a computer will cross domains there may now be incentive for a domain administrator to interfere with a client computer's operation with the intent of compromising another administrative domain. With the lack of network authentication the risk is greater than a few malicious administrators. Valid network clients masquerading as valid network access points can also be in a position to cause harm.

4 Proof of Concept Experiment

To further illustrate the unique vulnerabilities encountered due to the lack of mutual authentication and network promiscuity that arises with mobile computing we implemented a proof of concept software download MITM attack. Unlike the attacks in the previous section this attack is intended to take advantage of the roaming between administrative domains. Once a wireless client is compromised

by installing trojaned software, it brings that threat to any other network it encounters.

The scenario we shall consider is a Rogue Access Point in a corporate or university setting where a WEP key is established and only verified MAC addresses are permitted. This Rogue AP could be created by a valid user, using the authentication information he was given for his personal use. It could also be created by an outside attacker who has retrieved the WEP key via Aircsnort and a MAC address that he has observed by sniffing network traffic. Please note that the attack could be modified to fit several different scenarios.

The attacker will first authenticate to the existing network as a valid client with one WiFi card. A second WiFi card will be used to create the Rogue AP. It will emulate a valid AP as best it can. He can use the same SSID and require the same WEP key.

As clients connect, some will doubtlessly accidentally connect to the Rogue AP. If the attacker wants to target a specific wireless client he can do so. If the attacker knows the target clients MAC address he could force the clients disassociation from the legitimate AP until the client associates with the Rogue AP. Now, he will have complete control over the traffic of any client that has associated with him.

4.1 Experiment Setup

We perform this experiment with two Sony Vaio Z1 laptops running the Linux operating system. The unsuspecting client will be configured to connect to the corporate network with SSID "CORP" and have the WEP key entered into his machine (Figure 1).

The gateway machine has a D Link DWL-650 PCMCIA WiFi card and a Netgear MA101 USB WiFi card. The Netgear card is configured to be a client on the "CORP" network with the WEP key "SECRET" and use the Linux Atmel driver[12].

The D Link card is configured with the Linux hostap driver[8] to operate in Master mode, or to behave like an Access Point, with SSID "CORP". It also uses the "SECRET" WEP key.

After the proper configuration of the wireless interfaces an ARP proxy bridge was established between the two interfaces using parprouted[6]. After setting the appropriate routes, the gateway machine was ready to begin transparently bridging traffic. Appendix A contains a script that can be used for configuring the interfaces and bridge.

With the bridge enabled it is time to target a download. We set up a sample target download web page which contained a downloadable binary, a link to that downloadable binary and an MD5SUM of that binary. This download

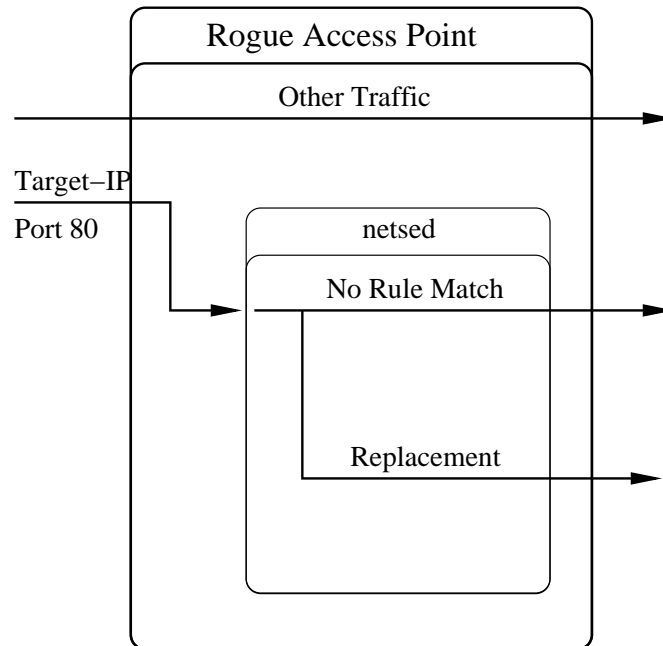


Figure 2. Software Download MITM Detail.

scenario is relatively common, where the MD5SUM is intended to verify that package was downloaded properly.

Since the clients traffic is already passing through the gateway machine it makes our job much easier. To accomplish this in a wired network is possible via ARP spoofing, DNS spoofing, or by compromising a legitimate gateway machine. Since we have already established ourselves as a legitimate gateway, all that is required is that we redirect the clients traffic destined to the Target website through our user space proxy. In this case, the redirection is handled via Netfilter in Linux. The following iptables command will accomplish this:

```
# iptables -t nat -A PREROUTING \
  -p tcp -d Target-IP --dport 80 \
  -j DNAT --to Gateway-IP:10101
```

This command redirects any TCP traffic destined for Target-IP on port 80 to Gateway-IP on port 10101. In this case, listening on port 10101 of the gateway is a program called netsed[16].

Netsed proxies this HTTP traffic to the destined host while watching for strings specified in the rules given to it. Upon a match, netsed will replace the identified string with the replacement string it was given. This is a simple search and replace. This netsed command was issued:

```
# netsed tcp 10101 Target-IP 80 \
  s/href=file.tgz/href= \
  http:%2f%2fGateway-IP%2ffile.tgz \
  s/REALMD5SUM/FAKEMD5SUM
```

This command tells netsed to listen on port 10101 (where Netfilter will direct all port 80 traffic for the Target-IP) and replace every occurrence of the legitimate link with the fake link and every occurrence of the real MD5SUM with the fake MD5SUM (the %2f is ASCII hex for the / character and will be properly interpreted by the web server). The net effect of doing these replacements is to replace the valid HTML link with a link to a trojaned version of the software desired by the client. It also manages to replace the MD5SUMs so the client is assured that the download has completed safely.

4.2 Experiment Conclusions

This particular implementation is only one particular way of accomplishing this attack. In fact it is even a rather naive attack, in that it reveals the real download IP to the client. In addition to that netsed will not match strings that cross packet boundaries. These, and other problems, could easily be addressed by someone with malicious intent. In fact, there are **many** variations on this attack. This approach could be used to do all sorts of nasty things to the client but we expect that this particular attack sufficiently illustrates the risks.

5 Solution

The previous example should convince the reader that even casual web browsing over a wireless link is suscepti-

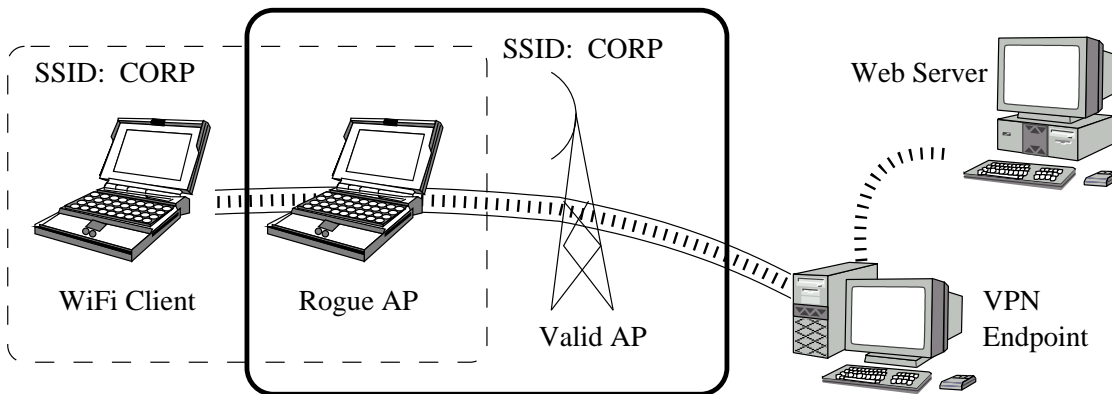


Figure 3. Example VPN Proxy Configuration in a Compromised Wireless Network.

ble to tampering of considerable consequence. The solution to this problem is to require **all** traffic to pass through a VPN to a trusted, secure, wired network. If, however, this example was not sufficient to convince the reader that **all** traffic should pass through the VPN then consider this scenario.

5.1 CNN - “Trustworthy” Websites

Consider a wireless client who thinks his web browsing habits are safe because he only visits large legitimate websites, like CNN. He doesn’t expect the administrators of that site to attempt to compromise his computer. This user may be a little behind on browser or email client updates. He doesn’t worry though, since CNN wouldn’t include a malicious javascript or similar client software exploit. On an unprotected wireless segment, the trust he places in the website provider is irrelevant, since, as our example shows, anyone could insert malicious code into any web content requested.

5.2 VPN Requirements

When considering a solution to this problem one must recall the nature of network promiscuity. A solution that is local to one network will not protect the client reliably. The risk of a client using a Hostile Hotspot and bringing trouble back home still remains. By tunneling all traffic through a secure VPN connection to a trusted network network promiscuity is no longer a concern.

The VPN must satisfy the following requirements:

1. Provided by trustworthy entity
2. Authentication information preestablished
3. VPN endpoint in secure wired network
4. Must handle all client traffic

The VPN endpoint could be provided by the client’s home corporation, home ISP, or perhaps a trusted third party. The important fact is, that arrangements for the VPN (secret exchange or certificate issuance) must take place out of band or on a secure network and not in a situation where the initial transaction would be vulnerable.

5.2.1 Hotspot Provider

This being said, it is not sufficient or at least not practical for a hotspot provider to provide this service. Since in the course of a mobile client’s life it will pass through many networks where the ownership is not immediately apparent. The client cannot be guaranteed to know who to perform an out of band protocol with.

One might ask, isn’t it sufficient for the client to interact with the service provider if that provider has a valid, signed SSL certificate from a legitimate certificate authority? Wouldn’t this enable a secure browser based transaction to facilitate a VPN connection? We would not consider this to be the case. Without knowing the WiFi provider’s reputation the valid certificate is a guarantee of nothing more than that provider having given the certificate authority several hundred dollars. Under certain circumstances, this is a small price to pay.

5.3 VPN Selection

There are many VPN solutions, both free and commercial. The selection of a specific VPN solution will depend upon the users requirements. For testing purposes we have utilized a PPP through SSH VPN as described in *Building Linux Virtual Private Networks*[7]. This of course has drawbacks since any UDP traffic is subject to unnecessary retransmission by TCP.

So, by requiring all traffic to VPN through a preestablished VPN provider we have prevented the possibility of a

vast array of attacks on the least secure segment of network the client's traffic is likely to encounter. In addition to this added security, the client's traffic can also be anonymized for privacy reasons at the VPN endpoint.

6 Conclusion

With this understanding of the fundamental differences between wired and wireless networks our task of working toward a secured wireless infrastructure in the future should be made easier. The idea of network promiscuity is with us today and with the increased popularity of light weight computing devices such as PDAs and IP enabled cell phones it is likely to be here with us tomorrow.

It is also important to keep an eye on network bridge technologies when considering oneself with network promiscuity. Even if WiFi evolves successfully into a secure technology does that device participate in other local networks, like Bluetooth[14]? Perhaps this scenario isn't very realistic yet, but in the long term, such things may become a threat in the future.

Future work will likely include a thorough evaluation of VPN technologies to determine their strengths and weaknesses with this application in mind. As well as improving techniques of detecting and countering attacks similar to the ones discussed here[15].

References

- [1] P. Bahl, S. Venkatachary, and A. Balachandran. Secure wireless internet access in public places. In *Proceedings of the IEEE ICC 2001*, June 2001.
- [2] D. B. Faria and D. R. Cheriton. Dos and authentication in wireless public access networks. In *ACM Wireless Security Workshop (WiSe'02)*, 2002.
- [3] S. Fluhrer, I. Mantin, and A. Shamir. Weaknesses in the key scheduling algorithm of RC4. *Lecture Notes in Computer Science*, 2259:1-??, 2001.
- [4] A. Godber and P. Dasgupta. Secure wireless gateway. In *ACM Wireless Security Workshop (WiSe'02)*, 2002.
- [5] IEEE 802.1 Working Group. 802.1x - port based network access control.
- [6] V. Ivaschenko. parprouted: Proxyarp routing daemon.
- [7] O. Kolesnikov and B. Hatch. *Building Linux Virtual Private Networks*. New Riders, Indianapolis, Indiana, 2002.
- [8] J. Malinen. Host ap driver for intersil prism2/2.5/3.
- [9] A. Mishra and W. A. Arbaugh. An initial security analysis of the ieee 802.1x standard.
- [10] NoCatNet. Nocat open authentication package.
- [11] A. Stubblefield, J. Ioannidis, and A. Rubin. Using the fluhrer, mantin, and shamir attack to break wep, 2001.
- [12] The Atmelwlandriver Team. Opensource linux driver for Atmel AT76C503A-based wireless devices.
- [13] WAVEsec Team. Wavelan security using ipsec.
- [14] O. Whitehouse. Redfang - the bluetooth hunter.
- [15] J. Wright. Detecting wireless lan mac address spoofing.
- [16] M. Zalewski. netsed.

A Bridge Configuration

```
#!/bin/sh

# This script, of course, assumes that
# you have already properly configured
# the wireless NICs with iwconfig
# eth1 - associated to CORP
#           in Managed Mode
# wlan0 - is in Master mode
#           with essid CORP

# Turn on IP Forwarding
echo "Turning on IP Forwarding ..."
echo 1 > /proc/sys/net/ipv4/ip_forward

# Give the NICs IPs
ifconfig wlan0 10.6.6.2 \
    netmask 255.255.255.0 \
    broadcast 10.6.6.255
ifconfig eth1 10.6.6.3 \
    netmask 255.255.255.0 \
    broadcast 10.6.6.255

# Create the bridge
parprouted wlan0 eth1

# Set some routes
route add -host 10.6.6.1 dev eth1
route add -host 10.6.6.7 dev wlan0
route add default gw 10.6.6.1

echo "Bridge enabled"
```