

Penetration Test Team

Related terms:

[Penetration Testing](#), [Backdoors](#), [Law Enforcement](#), [Commercial Tool](#), [Penetration Test Project](#), [Project Manager](#), [Test Team Member](#)

[View all Topics](#)

Pentest Project Management

Thomas Wilhelm, in [Professional Penetration Testing \(Second Edition\)](#), 2013

Future Project Priority Identification

A successful [penetration test](#) team will inevitably have too much business. When that happens, prioritization of projects must be carefully performed. The project manager will require input from numerous personnel before being able to prioritize projects, but there are some things that must be considered regardless:

- Overall security risk to the client
- Cost of each project
- Financial gain of each project
- Length of time needed for each project
- Skills needed to successfully complete each project
- Staff/resource availability (yes, even engineers take vacations)
- Project sponsor/requestor

All those factors influence project prioritization and should be considered before assignment. By identifying all factors involved in future projects, the project manager can arrange projects that maximize the use of resources and time.

Corporate organizational structure can influence the roles and responsibilities of professional penetration test team members. By understanding the advantages and disadvantages of each organization, the project managers can plan strategies to im-

prove the success of their projects. Regardless of which organizational structure the pentest team works under, the team must have the support of upper management, a team champion. The team must also have a strong project managerial presence and skilled [penetration test engineers](#) who are given ample opportunity to participate in training.

Even with the right combination of organizational design, team support, the right staff, and sufficient training, the project manager must address areas within a project that are unique to penetration testing. All phases of a project include challenges that must be overcome and opportunities to improve the long-term success of the team and its members.

[> Read full chapter](#)

Low tech vulnerabilities

Jack Wiles, ... Sean Lowther, in [Low Tech Hacking](#), 2012

Check all locks for proper operation

On every one of my [team's penetration tests](#), we found at least one lock (either interior or exterior) in the building that wasn't functioning properly. This provided us with easy access to buildings and rooms that we shouldn't have been able to get into so easily. If employees are trained for just a few minutes on how to check to see if the locks on the doors that they use every day are working properly, this vulnerability can be all but eliminated. Building maintenance teams should also take a close look at all locks at least twice each year. Slightly misaligned strikes on the doorframes are the most common problem that we find. This is a serious problem, in that it defeats the purpose of the dead bolt feature of the lock. It takes me less than a second with my trusty fingernail file to see if a particular lock (bolt) has this problem. If it does, I'll know (and have the door opened) in a few seconds.

[> Read full chapter](#)

Wireless Penetration Testing Using a Bootable Linux Distribution

Chris Hurley, ... Brian Baker, in [WarDriving and Wireless Penetration Testing](#), 2007

Case Study Cracking WEP

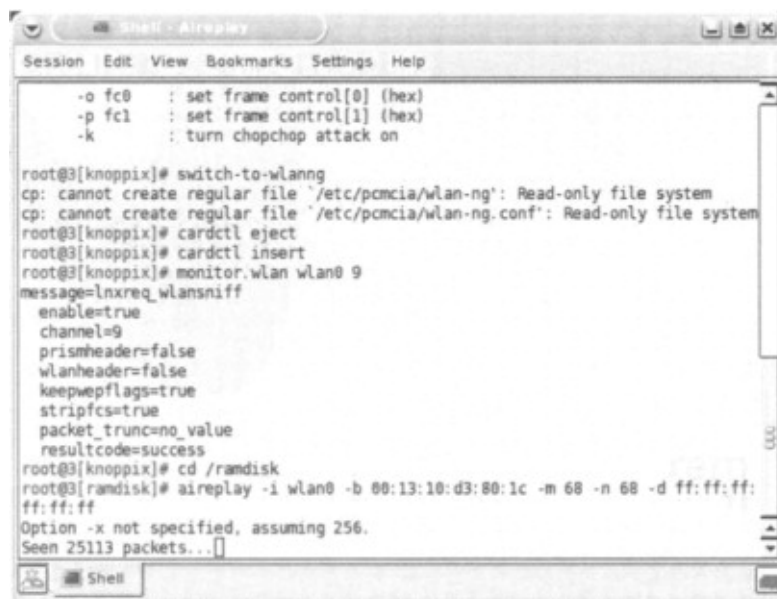
You have been assigned to perform a red team penetration test against Roamer Industries. You have been given no information about the wireless network or the internal network. You have to use publicly available sources to gather information. You know that Roamer Industries has deployed a wireless network, but that is all of the information you have.

Before you do anything else, you investigate the company by performing searches on Google and other available search engines, as well as the USENET newsgroups. You also go to the Roamer Industries public Web site to look for information and perform an ARIN WHOIS lookup on the IP address of their Web site. Quite a bit of important information is gleaned from these searches. The address of their office complex is listed on their Web site. The WHOIS lookup reveals the name and e-mail address of an individual that you discover is a system administrator, judging from the posts he has made on USENET. Additionally, you discover that they are using Microsoft Structured Query Language (SQL) server on at least one system, because that administrator described a configuration issue he was having while setting the server up on a Microsoft Structured Query Language (MSSQL) newsgroup.

Since you have been specifically tasked to test the WLAN, you note the address of the office complex where the WLAN is located and head to that area. Upon arrival, you fire up Kismet and drive around the building several times. You find 23 access points in the area of your target; 15 of them are broadcasting the SSID, but none are named Roamer Industries. This means that you have to gather the SSIDs of the other eight (obviously cloaked) networks. Since you don't want to inadvertently attack a network that does not belong to your target and thus violate your Rules of Engagement, you have to be patient and wait for a user to authenticate so that you can capture the SSIDs. It takes most of a day to gather the SSIDs of the eight cloaked networks, but once you have them all, you can try to determine which network belongs to your target. None of the SSIDs are easily identifiable as belonging to them, so you go back to Google and perform searches for each SSID you discovered. About halfway through the list of SSIDs you see something interesting: one of the SSIDs is InfoDrive. Your search for *InfoDrive Roamer Industries* locates a page on the Roamer Industries Web site describing a research and development project named InfoDrive. While it is almost certain that this is your target's network, before proceeding, you contact your white cell to ensure that this is their network. Once you have confirmation, you are ready to continue on with your penetration test.

Opening the Kismet dumps with Ethereal, you discover that WEP encryption is in use on the InfoDrive network. Now you are ready to start your attack against the WLAN. First, you fire up Aireplay and configure it to capture an ARP packet that you can

inject into the network and generate the traffic necessary to capture enough unique IVs to crack the WEP key. Once Aireplay is ready, you start Void11 and perform a deauthentication flood. Within a few minutes, Aireplay has captured a packet that it believes is suitable for injection (see Figure 7.17).



```
Session  Edit  View  Bookmarks  Settings  Help

-o fc0 : set frame control[0] (hex)
-p fc1 : set frame control[1] (hex)
-k      : turn chopchop attack on

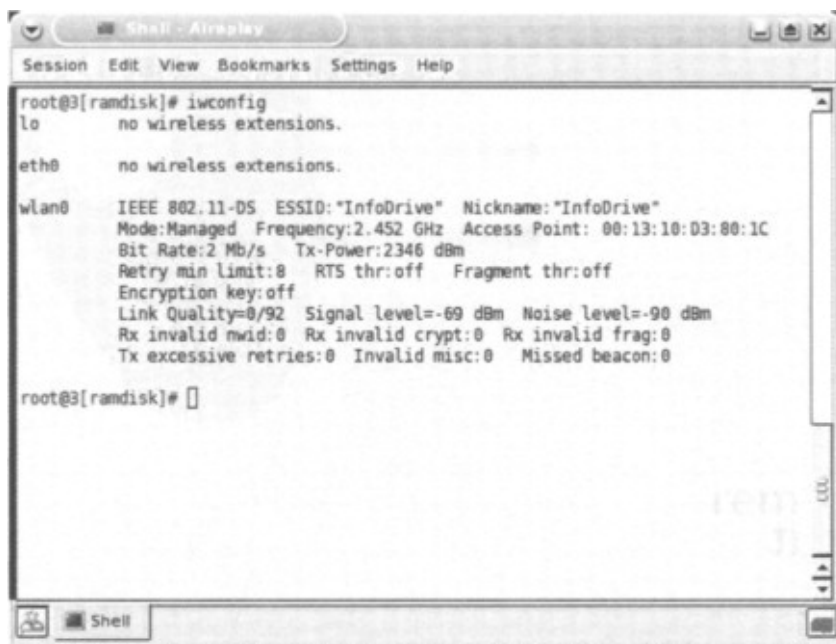
root@3[knoppix]# switch-to-wlanng
cp: cannot create regular file '/etc/pcmcia/wlan-ng': Read-only file system
cp: cannot create regular file '/etc/pcmcia/wlan-ng.conf': Read-only file system
root@3[knoppix]# cardctl eject
root@3[knoppix]# cardctl insert
root@3[knoppix]# monitor.wlan wlan0 9
message=lnxreq_wlansniff
enable=true
channel=9
prismheader=false
wlanheader=false
keepwepflags=true
stripfcs=true
packet_trunc=no_value
resultcode=success
root@3[knoppix]# cd /ramdisk
root@3[ramdisk]# aireplay -i wlan0 -b 00:13:10:d3:80:1c -m 68 -n 68 -d ff:ff:ff:
ff:ff:ff
Option -x not specified, assuming 256.
Seen 25113 packets...
```

Figure 7.17. Aireplay Searches for a Suitable Packet for Injection

Based on your criteria, you decide that this packet is probably going to work and begin the injection attack. Now that Aireplay is injecting traffic, you start Airodump to collect the packets and determine the number of unique IVs you have captured. Aireplay works quickly, and after about 20 minutes you have collected over 200,000 unique IVs. You decide it is worth checking to see if you have gathered enough IVs for Aircrack to successfully crack the WEP key. Once you have fired up Aircrack and provided your Airodump capture file as input, you find that you have not collected enough IVs yet. You continue your injection and packet collection for another 15 minutes, at the end of which you have collected over 370,000 unique IVs. You try Aircrack again. This time, you are rewarded with the 64-bit WEP key “2df6ef3736.”

Armed with your target's WEP key, you configure your wireless adapter to associate with the target network:

Issuing the **iwconfig** command with no switches returns the information about the access point that you are currently associated with. Your association was successful (see Figure 7.18).



```
root@3[ramdisk]# iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

wlan0     IEEE 802.11-DS  ESSID:"InfoDrive"  Nickname:"InfoDrive"
Mode:Managed  Frequency:2.452 GHz  Access Point: 00:13:10:03:00:1C
Bit Rate:2 Mb/s   Tx-Power:2346 dBm
Retry min limit:0   RTS thr:off   Fragment thr:off
Encryption key:off
Link Quality=0/92  Signal level=-69 dBm  Noise level=-90 dBm
Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
Tx excessive retries:0 Invalid misc:0  Missed beacon:0

root@3[ramdisk]#
```

Figure 7.18. Successful Association to the Target WLAN

Now that you have associated, you need to see if you can get an IP address and connect to the network resources. First, you try running **dhclient wlan0** to see if they are serving DHCP addresses. This doesn't work, so you go back to Kismet and look at the IP range that Kismet discovered. Kismet shows that the network is using the 10.0.0.0/24 range. You have to be careful here, because you don't want to take an IP address that is already in use. You look at the client list in Kismet and determine that 10.0.0.69 is available. Now you have to make some educated guesses as to how the network is set up. First, you try configuring your adapter with a default subnet mask of 255.255.255.0 and 10.0.0.1 as the default gateway:

```
ifconfig wlan0 10.0.0.69 netmask 255.255.255.0
route add default gw 10.0.0.1
```

Next, you ping the router to see if you have connectivity. Sure enough, you do. At this point, you have successfully established a foothold on the wireless network. Now you can probe the network for vulnerabilities and continue your red team engagement. The first avenue to explore would be the MS SQL server, since you know that this is a service that is often configured in an insecure manner. Since your target's administrator was asking for configuration help on a public newsgroup, chances are that he or she is not an extremely experienced MS SQL administrator, so your chances are good. From here, you continue your penetration test following your known methodologies. The WLAN was the entry vector you needed.

[> Read full chapter](#)

Building penetration test labs

10.2.1.1.3 Securing install disks

In a [penetration test](#) lab, you will use many different types of operating systems and software applications. It is important to store these disks in a secure manner, for two reasons. First, disks grow invisible legs and tend to walk out of your lab (intentionally or not). Second, you have to ensure the integrity of the disks you work with.

With regard to install disks “walking out,” anyone who has had to support a network finds himself short of disks. Sometimes it is because people borrow them, or sometimes the network administrators forget and leave disks in CD trays. You can prevent this by enforcing detailed procedures. However, the issue of install disk integrity is a more serious matter. Some operating system and patch disks are delivered through well-defined and secure channels (e.g., the Microsoft MSDN subscription will mail updates). However, more often than not, patches and updates are downloaded over the Internet. How does a person who downloads software over the Internet know that what he is downloading is a true copy of the file, and not corrupted or maliciously altered? Through hashes.

Although few people do this, all applications and software downloaded for use in a [penetration test lab](#) should be verified using a [hash function](#). The most popular is MD5, and for those security-conscious people and companies that provide downloads, a published MD5 value is usually associated with each download. Once the pen-test team has downloaded a file, they must verify that they have a true copy of the file by conducting an MD5 hash against it, and comparing it to the file author's published value. Then they should record the value somewhere for future reference, such as a binder stored in a safe.

You should run MD5 hashes against the install disks regularly, especially before you use them in the pen-test lab. This assures the [penetration test team](#) that they are using a true copy of the file. Verifying the hash can often provide defense against someone using the wrong version of the intended application. By comparing the MD5 hash of an application against a printed list, you will quickly know whether you have the wrong disk or file. This extra validation is a valuable safeguard against innocent mistakes that can ruin weeks' worth of work, if the wrong software is used by accident. Explaining to a boss that you have to repeat a two-week penetration test effort because you used a wrong software version can have a nasty result, especially during your next performance review.

Warning

Be aware that the same program can have different hash values, depending on the operating system. An MD5 hash in one Linux distribution might be different in another distribution, resulting in a false positive. It is important to keep track of which distro you are using when you record the hash.

[> Read full chapter](#)

Building penetration test labs

Jeremy Faircloth, in [Penetration Tester's Open Source Toolkit \(Fourth Edition\)](#), 2017

Securing install media

In a [penetration test](#) lab, you will use many different types of operating systems and software applications. It is important to store this media in a secure manner, for two reasons. First, media tends to grow invisible legs and walk out of your lab (intentionally or not). Second, you have to ensure the integrity of the media that you work with.

With regard to install media “walking out,” anyone who has had to support a network finds himself short of or missing some media. Sometimes it is because people borrow them, or sometimes the network administrators forget and leave disks in CD trays or USB devices attached to the last system that they worked on. You can prevent this by enforcing detailed procedures. However, the issue of installation media integrity is a more serious matter. Some operating system and patch disks are delivered through well-defined and secure channels (e.g., the Microsoft MSDN subscription will mail updates). However, more often than not, patches and updates are downloaded over the internet. How does a person who downloads software over the internet know that what he is downloading is a true copy of the file, and not corrupted or maliciously altered? Through hashes.

Although few people do this, all applications and software downloaded for use in a [penetration test lab](#) should be verified using a [hash function](#). The most popular is MD5, and for those security-conscious people and companies that provide downloads, a published MD5 value is usually associated with each download. Once the pen-test team has downloaded a file, they must verify that they have a true copy of the file by conducting an MD5 hash against it, and comparing it to the file author’s published value. Then they should record the value somewhere for future reference such as a binder stored in a safe.

You should run MD5 hashes against the install media regularly, especially before you use it in the pen-test lab. This assures the [penetration test team](#) that they are

using a true copy of the file. Verifying the hash can often provide defense against someone using the wrong version of the intended application. By comparing the MD5 hash of an application against a printed list, you will quickly know whether you have the wrong disk or file. This extra validation is a valuable safeguard against innocent mistakes that can ruin weeks' worth of work in the event that the wrong software is used by accident. Explaining to a boss that you have to repeat a 2-week penetration test effort because you used an incorrect software version can have a nasty result, especially during your next performance review.

Warning

Be aware that the same program can have different hash values, depending on the operating system. An MD5 hash in one Linux distribution might be different in another distribution, resulting in a false positive. It is important to keep track of which distro you are using when you record the hash.

[> Read full chapter](#)

Reporting Results

Thomas Wilhelm, in [Professional Penetration Testing \(Second Edition\)](#), 2013

Title Page

The title page is pretty self-explanatory and will be a way to introduce the topic of the report, as well the author and the [penetration test](#) team's organization. The title page is a great place to brandish logos and make everything look appealing, but the primary goal of the page should be to provide a clear message of what the report is about. It is possible that the client will have multiple [penetration test reports](#) on numerous targets; if the reports are all from the same pentest team, the title page will be used to quickly identify individual reports from each other.

[> Read full chapter](#)

Penetration Testing

Sanjay Bavisi, in [Computer and Information Security Handbook \(Second Edition\)](#), 2013

6 Defining What's Expected

Someone once said: “You can’t win the game if you don’t know the rules!” That statement makes good sense for a [penetration test](#) team as well. Every penetration test must have a clearly defined set of rules by which the penetration test “game” is played. These rules are put into place to help protect both the tested organization and the [penetration test team](#) from errors of omission and commission (under normal circumstances). According to the National Institute of Standards and Technology (NIST), the “rule book” for penetration tests is often called the “Rules of Engagement.” Rules of Engagement define things like which IP addresses or hosts are and are not allowed to be tested, which techniques are and are not allowed to be used, when testing is permitted or prohibited, points of contact for both the test team and the tested organization, IP addresses of machines from which testing is conducted, and measures to prevent escalation of an incident response to law enforcement, just to name a few.

There isn’t a standard format for the Rules of Engagement, so it is not the only document that can be used to control a penetration test. Based on the complexity of the tested organization and the scope of the penetration test, the penetration test team may also use detailed test plan(s) for either or both logical and physical testing. In addition to these documents, both the client and the penetration test company conducting the operation will require some type of formal contract that spells out either explicitly, or incorporates by reference, such items as how discovered sensitive material will be handled, an indemnification statement, nondisclosure statement, fees, project schedule, reporting, and responsibilities. These are but a few of the many items that [penetration testing](#) control documents cover, but they should be sufficient for you to understand that all aspects of the penetration test need to be described somewhere in a document.

We now know what’s expected during the penetration test. Armed with this information, how does the penetration test team plan to deliver the goods? It starts with a methodology.

[> Read full chapter](#)