Data Seepage and Ferret

Robert Graham
Errata Security, Inc.
Robert_david_graham@yahoo.com

David Maynor
Errata Security, Inc.
Dave@erratsec.com

Abstract

Data leakage is a well known problem that happens when secret data is inadvertently exposed to unauthorized individuals. Data seepage is the opposite, tiny bits on innocuous information that seems worthless can be combined to form a surprisingly dangerous view on an individual. This information doesn't require a court order or a team of spies, much of it is automatically leaked when a laptop is powered on (before it has a chance to establish a VPN). Data seepage doesn't just give passively viewing individuals information about a target, but also about the internal mappings of a corporate network.

1 Introduction

A typical laptop has many applications that start during the boot process or when a network is detected. Examples of these applications are web browsers, email clients, instant messaging clients, software update utilities, music software, and corporate software. These applications can seep tiny bits of information about the owner. Separately this information appears to be benign but when combined it paints a complete picture of an individual. This paper will describe the problem of data seepage, explore examples, and give recommendations on how to mitigate the risk.

Data seepage is not a new problem. The idea that a computer will seep information during startup is well know to people who use packet-sniffers. This paper describes how to apply this information in a malicious way. When a laptop goes through start up there are many applications or services that may attempt to connect to network resources. Even if a VPN application is used a lot of information about the laptop and the owner are broadcasted during startup. This information can range from where a laptop has been previously, to what websites it visits frequently to information about a company's internal network. This information can be gathered by an attacker and combined to give a accurate view of an individual and corporate network. This information can be used for phishing attacks or make client side attacks more targeted, increasing their reliability and chance of success.

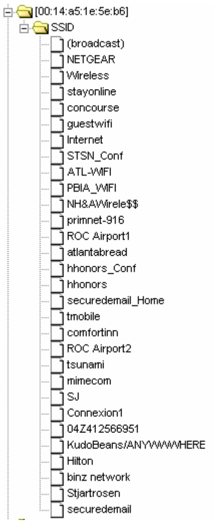
This is a problem that is well known within the military and intelligence community. The term for this is called EEFI or Essential Elements of Friendly Information. EEFI is defined as key questions likely to be asked by adversary officials and intelligence systems about specific friendly intentions, capabilities, and activities, so they can obtain answers critical to their operational effectiveness.[1] This means that non-essential information is gathered that can point to more strategic information. An example of this is food delivered to the Pentagon. It doesn't take more than a set of binoculars to tell when there is something major going on at the Pentagon, there is a surge in food delivery in the evening because many employees are staying late. The amount of food delivery that goes into the Pentagon is not a secret but it can give an indication that something major is about to occur.

2 Examples of seepage

Wifi probes

When a Windows laptop (and most other systems) is turned on, the first thing it will do is search for a list of "known" wireless access-points. It does this by sending 802.11 "probe" packets for the known access points. A "known" access point is one that the laptop has connected to in the past.

For example, a frequent-flier visits many airports and connects to the access points. By listening to a notebook, we can determine where that person has been. In the example to the right, we can see that the person has been at the Heartsfield Jackson airport in Atlanta, Georgia (SSID=ATL-WIFI) and the airport in Rochestoer, New York (SSID=ROC Airport).



```
[00:14:a5:1e:5e:b6]
  SSID
    (broadcast)
    NETGEAR
    Wireless
    stayonline
    concourse
    guestwifi
    Internet
    STSN_Conf
    ATL-WIFI
    PBIA_WIFI
    NH&AWirele$$
    primnet-916
    ROC Airport1
    atlantabread
    hhonors_Conf
    hhonors
    securedemail_Home
    tmobile
    comfortinn
    ROC Airport2
    tsunami
    mimecom
    SJ
    Connexion1
    04Z412566951
    KudoBeans/ANYWWWHERE
    Hilton
    binz network
    Stjartrosen
    securedemail
```

Imagine sitting in an airport lounge where a high-level executive of a small company announces that at some point in the past, he has connected to the internal Microsoft wireless network. This would suggest strongly that the company is about to be acquired my Microsoft.

DHCP

In a DHCP request a machine can request a previously used address. Capturing this packet will reveal what IP address the laptop previously had. Using a tool to locate the net block owner can tell you what company or ISP the laptop has been previously connected to.

DHCP also announces the computer's name, potentially information about the user, and services it's interested in. Since DHCP is used to acquire an IP address, all this information is disclosed before a user has a chanced to establish a VPN.

[1] http://usmilitary.about.com/od/glossarytermse/g/eefi.htm

mDNS and Bonjour

mDNS is a “multicast” version of DNS where the machine broadcasts information about what services it provides, and what services it might be interested in. An example of this are peer-to-peer applications, VOIP, instant-messaging, and software like Apple’s iTunes.

Whereas other services might broadcast a user name like “robg”, iTunes will broadcast the presence of your music library under a name like “Robert Graham’s Computer”. An attacker would then be able to pay a service to do a complete credit check on this name, as well as use an investigative service (such as ‘people.yahoo.com’) to find out a lot more about you, such as your family members and most of your former addresses.

NetBIOS

Microsoft’s file and print sharing is based on a protocol from the 1980s called NetBIOS. NetBIOS discloses a lot about a computer. It uses a broadcasts mechanism to register and discover “named” objects on the network.

Among the named objects will be your computer name and login name, both of which are useful for an attacker.

Other named objects are all the Microsoft servers in the corporation that the user is connected to. One of the most common NetBIOS broadcast seen in cyber-cafes is “EXCHANGE” as Outlook clients look for their corporate exchange server. Thus, by listening the broadcasts in the moments after you login into the wireless network, the attacker will already have a partial map of your internal corporate network.

Instant Messaging

One well-known problem is that people can eavesdrop on your instant-messaging conversations while in a cybercafé.

A lesser known issue is that even while you are not chatting, you are still sending an receiving status information about your friends. You may not even be aware that it’s activated. For example, if you use Yahoo mail or Google mail, these web-based clients are exchanging status information with other users of the services in your address book.

Thus, with in a few moments of logging on to the network, the attacker will know a lot about who your friends are.

Websites

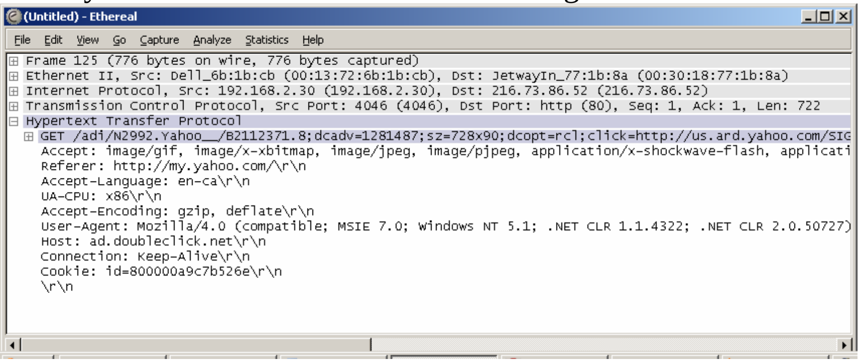
Web sites like Myspace and ESPN have features where a person can create an account to allow for customized settings and a more interactive experience. Sniffing this information can give a 3rd party viewer detailed personal information that may include links to personal sites that may contain bibliographic information and even personal pictures.

Many such sites do not send passwords in the cleartext, and can be sniffed. Many people use the same username and password for all their accounts, so compromising one of these sites can lead to further compromise of important sites, such as corporate accounts or online banking.

Web-bugs and cookies

This was a big news story, but the problem still exists. Websites track individuals with cookies and webbugs.

A person eavesdropping on this traffic can likewise track a user through cookies. For example, once an attacker knows the unique ID that DoubleClick gives to a user, they can identify that user whenever that ID is seen again in the future.



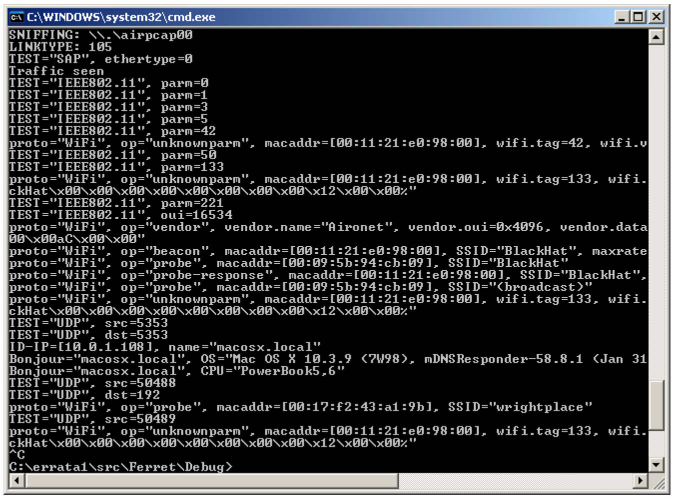
```
⊞ Frame 125 (776 bytes on wire, 776 bytes captured)
⊞ Ethernet II, Src: Dell_6b:1b:cb (00:13:72:6b:1b:cb), Dst: JetwayIn_77:1b:8a (00:30:18:77:1b:8a)
⊞ Internet Protocol, Src: 192.168.2.30 (192.168.2.30), Dst: 216.73.86.52 (216.73.86.52)
⊞ Transmission Control Protocol, Src Port: 4046 (4046), Dst Port: http (80), Seq: 1, Ack: 1, Len: 722
⊟ Hypertext Transfer Protocol
  ⊞ GET /adi/N2992.Yahoo__/B2112371.8;dcadv=1281487;sz=728x90;dcopt=rcl;click=http://us.ard.yahoo.com/SIG
    Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, applicati
    Referer: http://my.yahoo.com/\r\n
    Accept-Language: en-ca\r\n
    UA-CPU: x86\r\n
    Accept-Encoding: gzip, deflate\r\n
    User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.1.4322; .NET CLR 2.0.50727)
    Host: ad.doubleclick.net\r\n
    Connection: Keep-Alive\r\n
    Cookie: id=800000a9c7b526e\r\n
    \r\n
```

3 The FERRET tool

Ferret is a tool published on our website, www.erratasec.com. It is a typical packet-sniffer, but one designed to collect all this broadcasted information and collect it into a single location. By gathering all the bits that a person broadcasts, we can build a picture more useful than looking at any one piece of information individually.

The tool is written in portable C and has been compiled to run on many systems, including Windows, Mac OS X, Linux, and handheld devices.

Below is an example of the “raw” output from the device.

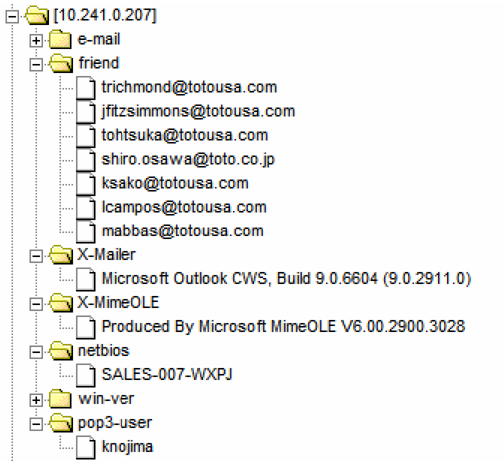


```
SNIFFING: \\.\airpcap00
LINKTYPE: 105
TEST="SAP", ethertype=0
Traffic seen
TEST="IEEE802.11", parm=0
TEST="IEEE802.11", parm=1
TEST="IEEE802.11", parm=3
TEST="IEEE802.11", parm=5
TEST="IEEE802.11", parm=42
proto="WiFi", op="unknownparm", macaddr=[00:11:21:e0:98:00], wifi.tag=42, wifi.v
TEST="IEEE802.11", parm=50
TEST="IEEE802.11", parm=133
proto="WiFi", op="unknownparm", macaddr=[00:11:21:e0:98:00], wifi.tag=133, wifi.
ckHat\x00\x00\x00\x00\x00\x00\x00\x00\x12\x00\x00%"
TEST="IEEE802.11", parm=221
TEST="IEEE802.11", oui=16534
proto="WiFi", op="vendor", vendor.name="Aironet", vendor.oui=0x4096, vendor.data
00\x00aC\x00\x00"
proto="WiFi", op="beacon", macaddr=[00:11:21:e0:98:00], SSID="BlackHat", maxrate
proto="WiFi", op="probe", macaddr=[00:09:5b:94:cb:09], SSID="BlackHat"
proto="WiFi", op="probe-response", macaddr=[00:11:21:e0:98:00], SSID="BlackHat",
proto="WiFi", op="probe", macaddr=[00:09:5b:94:cb:09], SSID="<broadcast>"
proto="WiFi", op="unknownparm", macaddr=[00:11:21:e0:98:00], wifi.tag=133, wifi.
ckHat\x00\x00\x00\x00\x00\x00\x00\x00\x12\x00\x00%"
TEST="UDP", src=5353
TEST="UDP", dst=5353
ID-IP=[10.0.1.108], name="macosx.local"
Bonjour="macosx.local", OS="Mac OS X 10.3.9 (7W98), mDNSResponder-58.8.1 (Jan 31
Bonjour="macosx.local", CPU="PowerBook5,6"
TEST="UDP", src=50488
TEST="UDP", dst=192
proto="WiFi", op="probe", macaddr=[00:17:f2:43:a1:9b], SSID="wrightplace"
TEST="UDP", src=50489
proto="WiFi", op="unknownparm", macaddr=[00:11:21:e0:98:00], wifi.tag=133, wifi.
ckHat\x00\x00\x00\x00\x00\x00\x00\x00\x12\x00\x00%"
^C
C:\errata1\src\Ferret\Debug>
```

Output is in the form of 'vectors', where a single vector is a chain of related bits of information. For example, we can associate a name with an IP address using mDNS, an IP address with a MAC address, and a MAC address with wifi probe packets to create a bigger picture of who that device is and where it has been.

These vectors can then be collected together to form a tree of information. The Ferret command-line tool can dump these vectors into an HTML file for processing by a Web 2.0 JavaScript program, and we have a tool called "Ferret Viewer" that processes the information in a GTK GUI.

The diagram below shows a typical tree. We have monitored many protocols and pulled out useful information for an IP address.



```
[10.241.0.207]
    e-mail
    friend
        trichmond@totousa.com
        jfitzsimmons@totousa.com
        tohtsuka@totousa.com
        shiro.osawa@toto.co.jp
        ksako@totousa.com
        lcampos@totousa.com
        mabbas@totousa.com
    X-Mailer
        Microsoft Outlook CWS, Build 9.0.6604 (9.0.2911.0)
    X-MimeOLE
        Produced By Microsoft MimeOLE V6.00.2900.3028
    netbios
        SALES-007-WXPJ
    win-ver
    pop3-user
        knojima
```

In this data, we get information about his login name, his computer name, versions of software running on his compute, and a list of friends.

4 Defense

The defense against data seepage is to consider every name that you use, and what that name discloses about you.

For example, when the computer asks you for you name, lie. Name your computer something like “SASQUATCH” instead of “ROBG”. When creating a user account, name is something like “ENTERPRISE” instead “RGRAHAM”. Windows asks for a “comment” field. Don’t put something helpful here, instead put something totally unhelpful. If you own a Dell notebook, put something like “ThinkPad” or “Mac OS X 10.0.2”.

The same applies to corporate servers. Name your exchange server something like “EXCHANGE” instead of “XYZ-CORP-MAIL”.

Don’t use the same username or password for important accounts. Your corporate account should be completely separate. Your banking account should be totally different. Go ahead and use the same name for MySpace and ESPN.com – hackers will discover these, but they can’t do much damage to your important accounts.

Conclusion

The more computers become “easy to use”, the more “chatty” they become. They broadcast this information to others because they want others to hear it. They want to broadcast your iTunes information, because they want nearby people to know about your iTunes library.

All this information seeping from your computer is now a danger to you. The non-secret information that you willingly broadcast to the world can now be used to attack your corporate account and your financial information. You should take steps to hide it.