- History of client signal strength (can help identify geographic location)
- Routing tables
- Stored packets before they are forwarded
- Packet counts and statistics
- ARP table (MAC address to IP address mappings)
- DHCP lease assignments
- Access control lists
- I/O memory
- Running configuration
- Processor memory
- Flow data and related statistics

6.2.3.2 Persistent

Again, like wired routers and switches, WAPs are not designed to include much local persistent storage space. The WAP operating system and startup configuration files are maintained in persistent storage by necessity. Persistent evidence you may find on a WAP includes:

- Operating system image
- Boot loader
- Startup configuration files

6.2.3.3 Off-System

Wireless access points can be configured to send event logs to remote systems for off-site aggregation and storage. Syslog and SNMP are commonly supported. Enterprise-class devices may include other options, often proprietary. Check the documentation for the model you are investigating and review local configuration to locate devices that may contain off-system WAP logs.

6.3 Wireless Traffic Capture and Analysis

Capturing and analyzing wireless traffic often provides valuable evidence in an investigation, for the same reasons we discussed in Chapter 3. However, there are some additional complexities involved in capturing wireless traffic, as opposed to sniffing traffic on the wire. In this section, we review some important notes for capturing and analyzing wireless traffic. For further discussion of passive evidence acquisition and analysis, please see Chapter 3, “Evidence Acquisition.”

6.3.1 Spectrum Analysis

There are, literally, an infinite number of frequencies over which data can be transmitted through the air. Sometimes the most challenging part of an investigator's job is simply identifying the wireless traffic in the first place.

For Wi-Fi traffic, the IEEE utilizes three frequency ranges:

- 2.4 GHz (802.11b/g/n)[19]
- 3.6 GHz (802.11y)[20]
- 5 GHz (802.11a/h/j/n)[21]

Each of these frequency ranges is divided into distinct channels, which are smaller frequency bands (for example, the IEEE has specified 14 channels in the 2.4 GHz range). Although the IEEE has set globally recognized frequency boundaries for 802.11 protocols, individual countries typically allow only a subset of these frequency ranges.

The precise frequencies in use vary by country. For example, the United States only allows WiFi devices to communicate over channels 1–11 in the 2.4 GHz range, while Japan allows transmission over all 14 channels. As a result, WiFi equipment manufactured for use in the United States is generally not capable of transmitting or receiving traffic on all of the channels used in Japan. This has important consequences for forensic investigators. For example, an attacker can purchase a Japanese WAP that supports Channel 14 and plug it into a corporate network in the United States, and U.S. wireless clients will not "see" the access point.

Wireless security researcher Joshua Wright has also published articles about the use of 802.11n in "Greenfield" (GF) mode. 802.11n devices operating in Greenfield mode are not visible to 802.11a/b/g devices. As a result, investigators scanning for wireless devices using 802.11a/b/g cards will not detect the 802.11n network. Please see Section 6.4.2, "Rogue Wireless Access Points," for more details.

When monitoring for the presence of wireless traffic, make sure that you fully understand the capabilities of your monitoring device, as well as the potential for devices that operate outside your range of detection.

---

19. IEEE, "IEEE Standard for Information Technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput" (October 29, 2009): Annex J, http://standards.ieee.org/getieee802/download/802.11n-2009.pdf (accessed December 31, 2011).

20. IEEE, "IEEE Standard for Information Technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 3: 3650-3700 MHz Operation in USA" (November 6, 2008): Annex J, http://standards.ieee.org/getieee802/download/802.11y-2009.pdf (accessed December 31, 2011).

21. IEEE, "IEEE Standard for Information Technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput" (October 29, 2009): Annex J, http://standards.ieee.org/getieee802/download/802.11n-2009.pdf (accessed December 31, 2011).

Spectrum analyzers are designed to monitor RF frequencies and report on usage. They can be very helpful for identifying stealthy rogue wireless devices and WiFi channels in use. MetaGeek's Wi-Spy product line supports the 2.4 GHz and 5 GHz frequency bands (as well as 900 MHz), and range in price from $100 to $1,000. AirMagnet (owned by Fluke Networks) also produces a popular wireless spectrum analyzer that can "identify, name and find: Bluetooth devices, 2.4G cordless phones, microwave ovens, RF Jammers, analog video cameras, etc."[22]

6.3.2 Wireless Passive Evidence Acquisition

In order to capture wireless traffic, investigators need an 802.11 wireless card capable of running in Monitor mode. Many wireless cards do not support this capability. Furthermore, in order to ensure totally passive monitoring, it is preferable to use a special-purpose WiFi monitoring card that can be configured to operate completely passively.

Riverbed Technology offers the AirPcap USB adapters, that are designed for exactly this task. The AirPcap USB adapter plugs into a USB port and can monitor Layer 2 WiFi traffic (one channel at a time). AirPcap software runs on Windows, integrates with Wireshark, and can be configured to automatically decrypt WEP-encrypted frames. The AirPcap "Classic" and "Tx" models support the 2.4 GHz 802.11b/g band, while the "Nx" model additionally supports 802.11n. The "Nx" model also includes an external antenna connector.[23] Figure 6–8 shows an example of the AirPcap USB dongle.

Figure 6–8. The AirPcap USB adapter from Riverbed Technology (previously CACE Technologies).

22. "WLAN Design, Security and Analysis," *Fluke Networks*, 2011, http://www.airmagnet.com/products/spectrum_analyzer/.

23. "Riverbed Technology—AirPcap," 2011, http://www.cacetech.com/products/airpcap.html.

For Linux users, the AirPcap USB adapter can be used via a modified driver (although the AirPcap software is still Windows-only). Josh Wright provides a patch for the zd1211rw wireless driver, which supports sniffing using the AirPcap dongle.[24]

Once you have the ability to monitor Layer 2 802.11 traffic, you can use standard tools such as tcpdump, Wireshark, and tshark to capture and analyze it.

Regardless of whether or not a WAP's traffic is encrypted, investigators can gain a great deal of information by capturing and analyzing 802.11 management traffic. This information commonly includes:

- Broadcast SSIDs (and sometimes even nonbroadcast ones)
- WAP MAC addresses
- Supported encryption/authentication algorithms
- Associated client MAC addresses

Even when the WAP traffic is encrypted, there is a single shared key for all stations. This means that anyone who gains access to the encryption key can listen to all traffic relating to all stations (as with physical hubs). For investigators, this is helpful because local IT staff can provide authentication credentials, which facilitate monitoring of all WAP traffic. Furthermore, there are well-known flaws in common WAP encryption algorithms such as WEP, which can allow investigators to circumvent or crack unknown encryption keys.

Once an investigator has gained full access to unencrypted 802.11 traffic contents, this data can be analyzed in the same manner as any other unencrypted network traffic.

6.3.3 Analyzing 802.11 Efficiently

So, you have some 802.11 frames. During the course of an investigation, you may search for the answers to questions such as:

- Are there any beacons in the wireless traffic?
- Are there any probe responses?
- Can you find all the BSSIDs/SSIDs from authenticated/associated traffic?
- Can you find malicious traffic? What does that look like?
- Is the captured traffic encrypted using WEP/WPA? Is anyone trying to break the encryption?

6.3.3.1 tcpdump and tshark

It's certainly true that you could use Wireshark to sort out the endianness problem for you, and you could use the graphical interface to try to zero in on the answers to any of the above questions. However, for large packet captures in particular, tcpdump and tshark tend to be more efficient and scalable.

---

24. http://www.willhackforsushi.com/code/zd1211rw-airpcap-linux-2.6.31.diff. (Accessed Jan. 6, 2012.)

With nothing but a powerful filtering language and an understanding of how 802.11 is structured—and how it transmits the bits—you can very quickly hone in on important wireless traffic. The following discussion presents useful BPF filters and display filters that can be used to filter 802.11 traffic.

Find the WAPs: Finding Beacon frames with tcpdump and BPF filters is straightforward, as shown below. Recall from Section 6.1.2.1 that Beacon frames are a type of management frame (type 0) with subtype 0x08. With a "Version" field of 0b00, the 0-byte offset of the 802.11 frame header (referred to as "wlan[0]") is 0b00001000. In order of transmission (remember that 802.11 is "mixed-endian") that becomes 0b10000000, or 0x80.

```
'wlan[0] = 0x80'
```

The 802.11 specification includes a 1-bit field called "ESS capabilities," which has a Wireshark field name of "wlan_mgt.fixed.capabilities.ess." According to the IEEE's 802.11 specification, "WAPs set the ESS subfield to 1 and the IBSS subfield to 0 within transmitted Beacon or Probe Response management frames."[25] Let's use tshark to search for Beacon or Probe Response frames where the ESS subfield is set to 1 and the IBSS subfield is set to 0, as shown below:

```
$ tshark -nn -r wlan.pcap -R '((wlan.fc.type_subtype == 0x08 || wlan.fc.
    type_subtype == 0x05) && (wlan_mgt.fixed.capabilities.ess == 1) && (
    wlan_mgt.fixed.capabilities.ibss == 0))'
  1    0.000000 00:23:69:61:00:d0 -> ff:ff:ff:ff:ff:ff 802.11 105 Beacon frame
      , SN=3583, FN=0, Flags=........, BI=100, SSID=Ment0rNet
265  20.409086 00:23:69:61:00:d0 -> 00:11:22:33:44:55 802.11 211 Probe
    Response, SN=3801, FN=0, Flags=........, BI=100, SSID=Ment0rNet
270  20.597504 00:23:69:61:00:d0 -> 00:11:22:33:44:55 802.11 211 Probe
    Response, SN=3804, FN=0, Flags=........, BI=100, SSID=Ment0rNet
335  23.318463 00:23:69:61:00:d0 -> 00:11:22:33:44:55 802.11 211 Probe
    Response, SN=3837, FN=0, Flags=........, BI=100, SSID=Ment0rNet
412  26.317951 00:23:69:61:00:d0 -> 00:11:22:33:44:55 802.11 211 Probe
    Response, SN=3873, FN=0, Flags=........, BI=100, SSID=Ment0rNet
[...]
```

Find the Encrypted Data Frames: Similarly, how can we filter quickly down to encrypted data frames? Just for fun, let's use a BPF filter to accomplish this. 802.11 data frames are version 0, type 2, subtype 0 (in binary 0b00100000). In order of transmission, the first byte ("wlan[0]") is 0b00001000, which in hexadecimal is 0x08.

As discussed earlier, the "Protected" bit indicates whether the frame is encrypted using WEP, TKIP, or AES-CCMP. The Protected bit is located at bit 6 of the 1-byte offset of the 802.11 frame (refer to Figures 6–1 and 6–3). With fields reversed within the byte for transmission, the Protected bit is the second bit received in the 1-byte offset ("wlan[1]"). Consequently, we have to construct a bitmask of 0b01000000 (0x40 in hexadecimal) to test whether the Protected bit is set.

---

25. IEEE, "IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications" (June 12, 2007): 251, http://standards.ieee.org/getieee802/download/802.11-2007.pdf. (accessed December 31, 2011).

The combination of the two tests, shown below, produces all of the encrypted data packets in a given capture![26]

```
'wlan[0] = 0x08 and wlan[1] & 0x40 = 0x40'
```

6.4 Common Attacks

Often, investigators suspect that a wireless network has been or is currently under attack. Common attacks on wireless networks include:

- **Sniffing** An attacker eavesdrops on the network
- **Rogue Wireless Access Points** Unauthorized wireless devices that extend the local network, often for an end-user's convenience
- **The Evil Twin Attack** An attacker sets up a WAP with the same SSID as a legitimate WLAN
- **WEP Cracking** An attacker attempts to recover the WEP encryption key to gain unauthorized access to a WEP-encrypted network.

It is important for network forensic investigators to recognize the signs of common attacks. We discuss each of these in detail below.

6.4.1 Sniffing

Eavesdropping on wireless traffic is extremely common, in part because it is so easy to do! From script kiddies in coffeeshops to professional surveillance teams, wireless traffic monitoring is, frankly, popular. Even where it is completely illegal, the risk of detection is exceptionally low, and the information gained can be very valuable. Both forensic investigators and attackers alike know how to passively monitor wireless traffic and use this technique to their advantage.

Wireless LANs, by virtue of their physical medium, can be accessed over great distances. Although WLANs can be designed to serve a specific geographic range, it is challenging for network administrators to limit the signal to that area and prevent leakage.

The FCC stipulates rules that govern the effective range of 802.11 transmissions. Based on these rules, theoretically the distance from which a station can interact with a wireless access point is limited to roughly 200 feet or 61 meters.[27] However, directional antennae can be constructed from off-the-shelf components that can dramatically increase the effective

26. IEEE, "IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications" (June 12, 2007): 60–64.

27. "Title 47 CFR Part 15: Low Power Broadcast Radio Stations, Audio Division (FCC) USA," 2011, http://www.fcc.gov/mb/audio/lowpwr.html.

ranges. (As we discussed in Section 3.1.2, one research team claimed a successful data transfer of 3Mbps over a distance of 238 miles![28])

Eavesdropping on telecommunications (including those transmitted over RF) is a violation of wiretap statutes in many jurisdictions. Remember that even stations that are not associated with a wireless network can capture and analyze WAP traffic. Forensic investigators should be aware that an attacker may have access to the network via a WAP, and that they may be able to monitor local traffic or communicate on the LAN from a location far outside what is considered normal range, a great distance away.

6.4.2 Rogue Wireless Access Points

For $40, anyone can purchase a cheap WAP and plug it into the company network. Often, employees do this simply for the sake of convenience, not realizing that it opens the company to attack. Criminals also deliberately plant wireless access points that allow them to bypass the pesky firewall and remotely access the network later on. These days, disgruntled employees can easily hide a WAP behind the file cabinet before cleaning out their desks and then access the company network months later from the parking lot.

Many companies conduct regular "war-walking" scans to detect rogue access points (i.e, using Kismet or NetStumbler) or invest in commercial wireless intrusion detection systems (WIDSs). However, there are sneaky ways to bypass traditional war-walking and WIDSs.

Forensic investigators should be aware of the methods that attackers can use to place rogue access points and evade detection. Rogue access points can be used to covertly extend the range of an internal network, facilitating access from far outside the physical bounds that network administrators might expect. Rogue access points may also allow for untracked LAN access, and act as a pivot point for attacks.

Conversely, in certain situations a forensic investigator may be charged with monitoring a network in which the network administrators are hostile or unaware of the investigation. In these circumstances, where law and ethics allow, it may be the forensic investigator employing these same techniques for the purposes of covert monitoring and evidence acquisition.

6.4.2.1 Changing the Channel

In the United States, the FCC has licensed 11 channels for 802.11b/g/n, which have center frequencies between 2.412 GHz to 2.462 GHz. However, most of Europe allows 13 channels (up to 2.472 GHz) and Japan allows 802.11b all the way up to channel 14, or 2.484 GHz.[29]

Cards manufactured for the United States often don't support channel 14, since it's illegal to transmit on that frequency. There's overlap between the channels, but at 2.484 GHz, channel 14 is far enough away from channel 11 that network cards are unlikely to pick up much signal on channel 11. If an attacker were to configure a WAP to illegally transmit on channel 14 and export data at 2.484 GHz, security teams monitoring U.S. channels would probably never detect it.

---

28. Michael Kanellos, "Ermanno Pietrosemoli has set a new record for the longest communication Wi-Fi link," *Historia de Internet en Amrica Latina y el Caribe*, June 2007, http://interred.wordpress.com/2007/06/18/ermanno-pietrosemoli-has-set-a-new-record-for-the-longest-communication-wi-fi-link/.

29. "List of WLAN channels—Wikipedia, the free encyclopedia."

Similar tactics are effective in other countries, when attackers use frequencies outside the bounds of normal wireless device operation.

6.4.2.2 802.11n Greenfield Mode

The IEEE's 802.11n ("MIMO"-based) specification is designed to allow much greater throughput than 802.11a/b/g (100Mbps or more).[30] The 802.11n standard specifies two modes:[31]

- "Mixed mode," which allows it to work with legacy 802.11a/b/g networks,
- "Greenfield" (GF) or "high-throughput-only" mode, which takes full advantage of the enhanced throughput but is not visible to 802.11a/b/g devices. Older devices will see GF-mode traffic only as noise.

Not visible to 802.11a/b/g devices? That means if you're war-walking with an 802.11a/b/g card, you can't see 802.11n devices operating in Greenfield (GF) mode. Even before the specification was finalized, 802.11n devices were already available for as little as $50—easy to buy, easy to plug into the company's network. However, many companies have not yet purchased 802.11n-compatible equipment and hence cannot detect GF-mode 802.11n rogue WAPs.

Josh Wright submitted a vulnerability report explaining this, in which he wrote: "With the inability to decode GF mode traffic, an attacker can position a malicious rogue WAP on a victim network using the GF mode preamble. This would allow an attacker to evade wireless intrusion detection systems (WIDS) based on non-HT devices. This includes all WIDS devices based on 802.11a/b/g wireless cards."[32]

6.4.2.3 Bluetooth Access Point

When you think about Bluetooth, you probably envision your tiny little headset that crackles and hisses every time you walk too far away from your phone. That's because your Bluetooth headset is designed for a Class 2 Bluetooth network, which is fairly low-power (2.5mW) and has a maximum range of about 9m.[33]

However, there's more to Bluetooth than your rinky-dink headset. Bluetooth Class 1 devices are much more powerful, with ranges similar to 802.11b WAPs. A Bluetooth Class 1 device can transmit up to 100mW, with a typical range of up to about 91m (or possibly

30. Joshua Wright, "Wireless Ethical Hacking, Penetration Testing, and Defense: Wireless Architecture & Analysis," The SANS Institute, 2008.

31. IEEE, "IEEE Standard for Information technology—Telecommunications and information exchange between systems— Local and metropolitan area networks—Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Amendment 5: Enhancements for Higher Throughput" (October 29, 2009), http://standards.ieee.org/getieee802/download/802.11n-2009.pdf (accessed December 31, 2011).

32. Joshua Wright, "GF Mode WIDS Rogue AP Evasion," *Wireless Vulnerabilities and Exploits*, November 13, 2006, http://www.wirelessve.org/entries/show/WVE-2008-0005.

33. Karen Scarfone and John Padgette, "Guide to Bluetooth Security: Recommendations of the National Institute of Standards and Technology," Special Publication 800-121, National Institute of Standards and Technology, September 2008, http://csrc.nist.gov/publications/nistpubs/800-121/SP800-121.pdf (accessed December 31, 2011).

miles, if the receiver has a directional antenna).You can buy a Class 1 Bluetooth WAP for $100–$200.[34]

Can you discover Bluetooth WAPs while war-walking? Not if you're just using an 802.11 card. Even if you're using a spectrum analyzer like WiSpy, you may not notice it. Bluetooth uses Frequency Hopping Spread Spectrum,[35] and hops 1,600–3,200 times a second across 79 channels throughout the 2.4–2.4835 GHz band. Because it's spread out across the spectrum, it can be hard to notice and easily mistaken for noise by the untrained eye. Most Wireless IDS systems and security teams simply don't look for it (yet).[36, 37]

6.4.2.4 Wireless Port Knocking

Remember port knocking? Instead of installing a backdoor to listen on a particular port (where it might be noticed), l33t h4x0rs installed rootkits that would wait for a particular sequence of ports to be scanned, at which point the knocker's IP address would be granted access. With wireless knocking, a rogue WAP sits on the network in monitor mode, listening for probe requests. When the rogue WAP receives a packet (or sequence of packets) with the preconfigured SSID, it awakens and switches to master mode. The program "WKnock" is designed for this purpose,[38] and it can be installed on any WAP supported by the OpenWRT framework. During times when the rogue WAP isn't active, it is silent and can't be detected using common wireless scanning tools. Sneaky![39]

6.4.3 Evil Twin

The "Evil Twin" attack is when an attacker sets up a WAP with the same SSID as one that is used in the local environment, usually in order to conduct a man-in-the-middle attack on an 802.11 client's traffic.

By default, commercial 802.11 clients associate with the SSID that their operators tell them to. If there is more than one WAP with the same SSID, as will be the case with most centrally-managed wireless network (either corporate or in a Wi-Fi "hotspot") then the client will associate with the WAP providing the strongest signal. When the Evil Twin's signal strength is stronger than the "real" WAP, 802.11 clients will associate with the Evil Twin.

It is trivial for any 802.11 device to masquerade as the closest infrastructure WAP for any given SSID. Any 802.11 device can be made to advertise itself as an available peer. These advertisements can be of two kinds: ad-hoc and infrastructure. By default, commercial

34. *Ibid.*

35. Sherri Davidoff, "Philosecurity, Blog Archive: Off the Grid," July 28, 2008, http://philosecurity.org/2008/07/28/off-the-grid.

36. Joshua Wright, "Wireless Ethical Hacking, Penetration Testing, and Defense: Wireless Security Exposed, Part 4," The SANS Institute, 2008.

37. Karen Scarfone and John Padgette, "Guide to Bluetooth Security: Recommendations of the National Institute of Standards and Technology," Special Publication 800-121, National Institute of Standards and Technology, September 2008, http://csrc.nist.gov/publications/nistpubs/800-121/SP800-121.pdf (accessed December 31, 2011).

38. "rstack: wknock," 2011, http://rstack.org/oudot/wknock.

39. Oudot Laurent, "WLAN and Stealth Issues," 2005, http://www.blackhat.com/presentations/bh-europe-05/BH_EU_05-Oudot/BH_EU_05-Oudot.pdf.

WAPs are infrastructure devices and, by default, most commercial operating systems that support 802.11 networking devices allow them to advertise as ad-hoc networks for peer-to-peer purposes. However, it is not difficult to switch an 802.11 interface on a desktop or laptop into infrastructure mode. With Linux, it's as easy as a single "iwconfig" command.

The "Evil Twin" ruse allows any sufficiently strong 802.11 broadcaster to become a "man-in-the-middle" between the unwitting client and every other system that it communicates with. Sufficiently strong broadcasting can be accomplished over surprisingly wide geographic areas.

Once a client has connected to the "Evil Twin," the attacker can intercept traffic, replace images or words on the fly, conduct SSL-stripping attacks, harvest credentials, and more.

6.4.4 WEP Cracking

Security professionals often joke that "WEP" stands for "Weak Encryption Protocol." This isn't far off the mark (although "WEP" really stands for "Wired Equivalent Privacy," as we discussed earlier). Due to flaws in the protocol, there are tools that can help attackers "crack" WEP keys in minutes, and thereby gain access to any WEP-protected network or packet capture.

WEP is designed to encrypt the payload of data frames on a wireless network using a shared key. The key, once selected, is distributed to all stations as a "pre-shared key" (PSK). The PSK itself is never exposed on the network, and so it is expected to be shared in some out-of-band way between the stations that need it.

Each station encrypts the payload of all data frames with the PSK and a randomly selected initialization vector (IV) so that the encryption key changes for every frame. The problem with using an IV in a reversible, symmetric encryption algorithm, such as RC4, is that stations have to supply the IV in plain text. Each station adds a cleartext 24-bit IV to each frame, but 24 bits is actually quite small when you consider the number of frames that can be transmitted across a WLAN. With only 24 bits of IV, the randomized values are bound to repeat at some point, given enough traffic. (This is guaranteed to happen after 2^{24}—or 16,777,216—frames. With a "maximum transmit unit" (MTU) of 1,500 bytes, that's less than 24GB of network data.)

As it turns out, however, after only a few thousand packets you can reliably guess that at least some of those packets have been encrypted with the same IV, but have different plain text input and ciphertext output. This enables attackers to leverage the "related-key attack", based on the knowledge of some of the bits of the key material.

An attacker's ability to leverage the related key attack depends on the volume of IVs exposed. On a quiet network, it may take weeks to capture enough IVs to crack the key. Fortunately for the attacker (unfortunately for the rest of us), there are weaknesses in WAP behaviors and implementations that allow attackers to force stations on a WLAN to generate large volumes of IVs. Using widely published tools, attackers can force the generation of enough IVs to crack a WEP key within minutes—even on an unused WLAN.

If you see anomalous behavior from an unknown station on a WEP-encrypted WAP, it could be that the station is attempting to crack the WEP key in order to gain access to the network. Commonly, WEP-cracking tools used on relatively quiet networks are designed to force local stations to generate unnecessary packets, with lots of IVs to speed cracking.

6.5 Locating Wireless Devices

Perhaps the single most challenging aspect of wireless networks for the investigator is the inherent difficulty in physically locating devices of interest. A compromised laptop may physically move throughout an enterprise's network; a rogue wireless access point may be hidden in crafty places like under ceiling tiles.

Strategies for locating wireless devices include:

1. Gather station descriptors, such as MAC addresses, which can help provide a physical description so that you know what to look for;
2. For clients, identify the WAP that the station is associated with (by SSID);
3. Leverage commerical enterprise wireless mapping software;
4. Poll the device's signal strength; and
5. Triangule on the signal.

Of course all of this takes time, and is far more challenging if the device sought for is mobile and only transiently on the network—exactly the sort of thing that Wi-Fi networks were designed to accommodate.

6.5.1 Gather Station Descriptors

You can learn a lot about what a wireless device probably looks like from its network traffic. For example, recall our earlier discussion from Chapter 4, "Packet Analysis," in which we learned that every network card is assigned a unique OUI by the manufacturer. The 802.11 frame indicates the source and destination station MAC addresses. (For wireless access points, the "BSSID" field in the 802.11 header is also the MAC address of the WAP's network card.) Although MAC addresses can be changed, in most cases, no one bothers to change them. Hence, from sniffing Layer 2 network traffic and examining the MAC addresses in 802.11 frames of interest, you can make an educated guess as to the manufacturer of the device generating the traffic. Figure 6–9 shows the 802.11 frame of traffic between an Apple device and a Cisco WRT54G wireless router. Note that Wireshark automatically translates the OUI into a manufacturer description.

The content of wireless traffic itself can provide a surprising amount of insight regarding the physical description of a device. In Figure 6–10, we were able to crack the WEP key of the wireless traffic and decrypt the contents of the data frames. Now, we can see the contents of communications between the Apple device and its Layer 3 endpoint (routed through the Cisco WAP, of course). The traffic includes HTTP data, which contains User-Agent headers sent by the Apple device. The frame highlighted in Figure 6–10 reveals a User-Agent string "iTunes-iPad/3.2.1 (16GB)." That's handy! Now we know that we're most likely looking for a 16GB iPad, running OS version 3.2.1. This evidence correlates nicely with the Apple MAC address we examined moments ago.

6.5.2 Identify Nearby Wireless Access Points

Your strategy for locating a wireless device will depend in part on the function of the device. For example, you may be searching for a rogue wireless access point or a roving endpoint

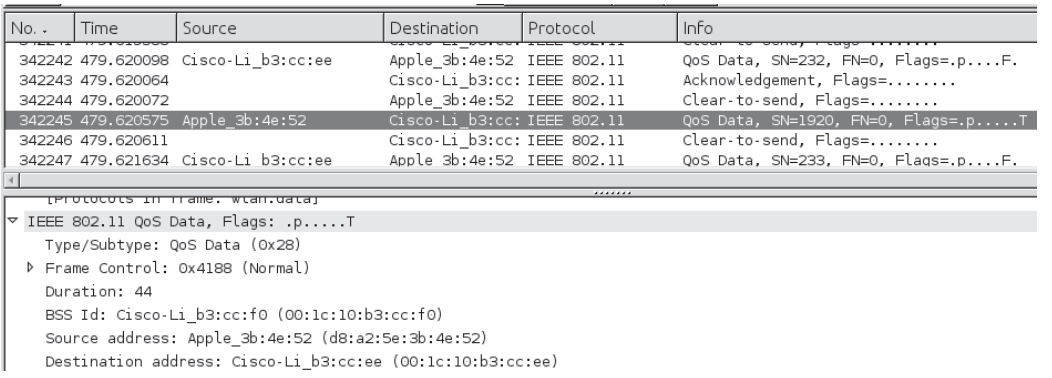

Figure 6–9. An 802.11 frame from an Apple device to a Cisco wireless router. Note that Wireshark automatically translates the OUI into a human-readable manufacturer description.

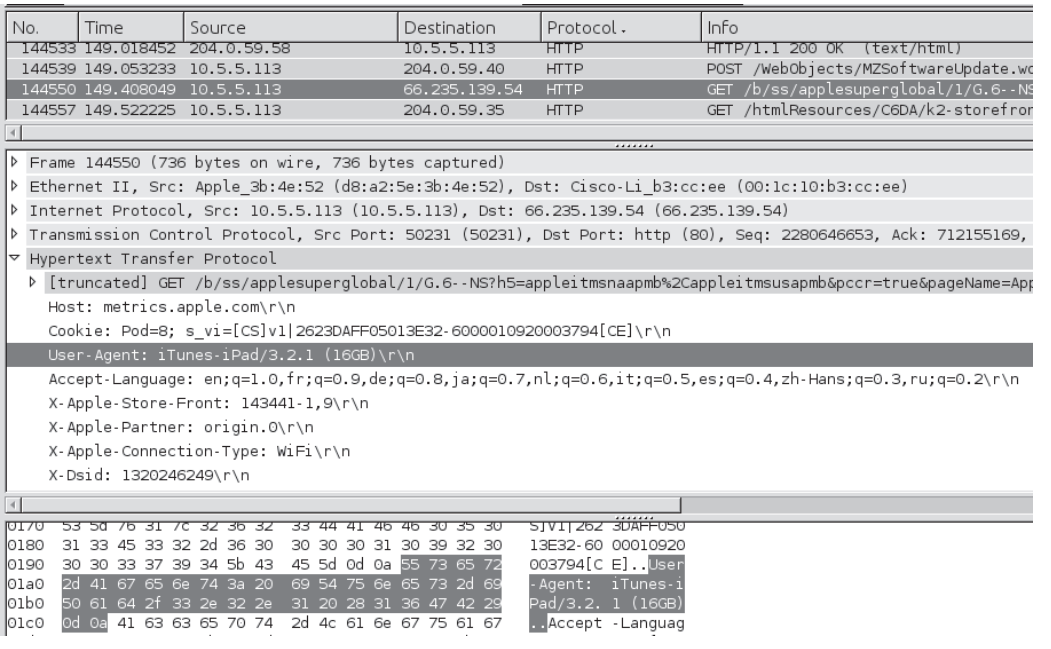

Figure 6–10. With the packet capture WEP-decrypted, we can see the User-Agent client-side HTTP header, which seems to confirm that the device is indeed an Apple.

station. In the case where you are searching for a client station that is actively associating with other WAPs, it is often helpful to identify which WAPs the station is associating with. Generally (although not always), endpoint stations associate with a wireless access point that is physically close by. In the case of a wireless bridged network, clients typically associate with the WAP in the bridged network that has the strongest signal, which is also often physically closest.

There are basically two ways to find out which WAPs a rogue endpoint client is associated with or attempting to associate with: WAP logs and traffic monitoring.

If you are lucky enough to be in an environment that captures wireless authentication attempts on a central logging system, you may be able to watch station association requests and responses by examining logs on the central server.

Otherwise, you can passively monitor the wireless traffic for association requests, responses, and other Layer 2 traffic related to the MAC address of interest. Generally, this requires that either you know the general vicinity of the rogue device already and can sniff traffic in that area, or that you have access to a wireless intrusion detection system with sensors distributed around a wide area.

In this way, you can track the station as it moves from device to device and locate the client using locations of known WAPs.

6.5.3 Signal Strength

There are many tools such as NetStumbler or Kismet that will list the nearby wireless access points and show you their relative signal strengths. Often, you can locate a mysterious wireless device simply by viewing the signal strengths using one of these applications and walking in the direction of increasing signal strength. This works well in situations where the station of interest is not mobile.

6.5.3.1 Received Signal Strength Indication (RSSI)

It is sometimes possible to see both the IEEE 802.11 Received Signal Strength Indication (RSSI) and the Transmit (Tx) Rate information when viewing a packet capture—but only if the tool that captured the packets supplies that data in its own additional framing. The 802.11 specification simply doesn't include such information in the data link–layer header.

If available, per-frame RSSI and Tx Rate information can be added manually to Wireshark's Packet List pane by editing user preferences.[40]

6.5.3.2 NetStumbler

NetStumbler[41] is a Windows tool designed to discover 802.11 networks. Though it is extremely popular for blackhats and whitehats alike, it is not totally passive, which means that its presence and activities can be detected by other wireless auditing tools. Like most tools of its kind, it supports GPS integration for the mapping of signals to physical locations, making it useful for "wardriving" or "warwalking." NetStumbler is free for download, though not open source.

Due to considerable architectural differences between XP and Vista/Windows 7, NetStumbler does not work on the latter. Vistumbler is a similar tool designed to run on Vista, though provided by different authors, and so it has a different user interface and functionality.[42, 43] A more popular replacement for all three platforms is inSSIDer.[44]

---

40. A. Orebaugh et al., *Wireshark & Ethereal: Network Protocol Analyzer Toolkit* (Syngress, 2006).

41. Mariusm, "stumbler dot net," February 16, 2010, http://www.stumbler.net/.

42. *Ibid.*

43. "Vistumbler," December 12, 2010, http://vistumbler.sourceforge.net/.

44. "inSSIDer," http://www.metageek.net/products/inssider/.

6.7 Case Study: HackMe, Inc.

The Case: *September 17th, 2010: Inter0ptic is on the lam and is pinned down. The area is crawling with cops, and so he must stay put. But he also desperately needs to be able to get a message out to Ann and Mr. X. Lucky for him, he detects a wireless access point (WAP) in the building next door that he might be able to use. But it is using encryption, and there are no other opportunities available. What is Inter0ptic to do?*

Meanwhile ... *Next door, Joe is a sysadmin at HackMe, Inc. He runs the technical infrastructure for a small company, including a WAP that is used pretty much exclusively by him. He's trying to use it now, and has discovered that he's begun to get dropped. He captures some traffic, but he really has no idea how to interpret it. Suddenly he discovers he can't even login to administer his WAP at all!*

The Challenge: **You are the forensic investigator.** Your team got a tip that Inter0ptic might be hunkered down in the area. Can you figure out what's going on and track the attacker's activities?

The following questions will help guide your investigation:

- What are the BSSID and SSID of the WAP of interest?
- Is the WAP of interest using encryption?
- What stations are interacting with the WAP and/or other stations on the WLAN?
- Are there patterns of activity that seem anomalous?
- How are they anomalous: Consistent with malfunction? Consistent with maliciousness?
- Can we identify any potentially bad actors?
- Can we determine if a bad actor successfully executed an attack?

Evidence: Joe has provided you with a packet capture (wlan.pcap) and permission to inspect it in any way you need to either solve his problem, catch Inter0ptic, or both. He also helpfully tells you that his own system's MAC address is 00:11:22:33:44:55, and reiterates that no one else should be using his WAP.

6.7.1 Inspecting the WAP

The most obvious place to begin analysis is Joe's WAP. Along the way we expect—or at least hope—to learn something about the stations with which it was communicating, and to be able to infer a whole lot from the anomalous traffic we're about to examine. Let's begin by identifying and inspecting the WLAN under investigation.

6.7.1.1 Inspecting Beacon Frames

Probably the most straightforward way to identify the WAPs in a packet capture is to simply filter on Beacon frames. Figure 6–12 demonstrates how Wireshark can be used with a display filter on the appropriate frame type (0) and subtype (8): "wlan.fc.type_subtype == 0x08." Note also the "BSS Id" in the frame: 00:23:69:61:00:d0.

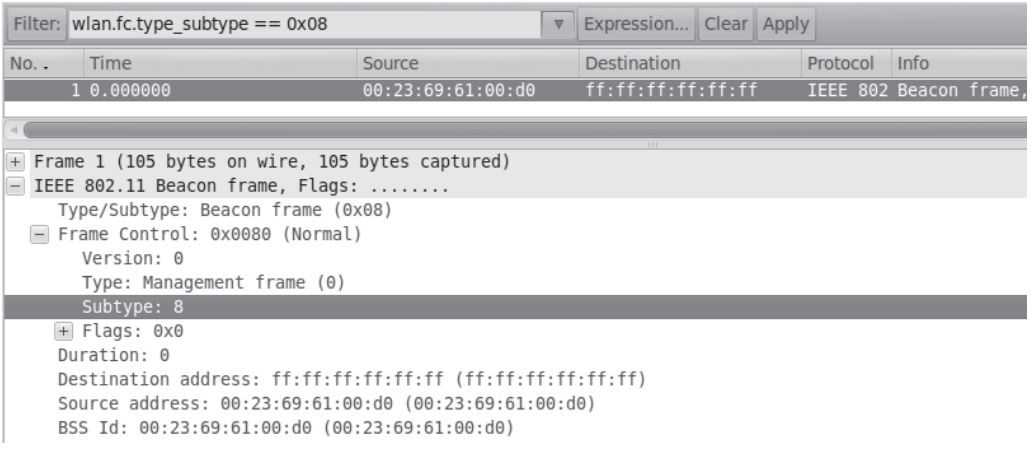

Figure 6–12. An 802.11 management frame shown in Wireshark. As you can see in the Packet Details pane, this frame is type 0, subtype 8: a Beacon frame.

Using tcpdump with the BPF language, we can easily find this Beacon frame too, so long as we mind our endianness:

```
$ tcpdump -nne -r wlan.pcap 'wlan[0] = 0x80' reading from file wlan.pcap,
    link-type IEEE802_11 (802.11) 09:56:41.085810 BSSID:00:23:69:61:00:d0 DA:
    ff:ff:ff:ff:ff:ff SA:00:23:69:61:00:d0 Beacon (Ment0rNet) [1.0* 2.0* 5.5*
    11.0* 18.0 24.0 36.0 54.0 Mbit] ESS CH: 2, PRIVACY
```

We see the same BSSID as before, and some other useful information (SSID, channel, etc.). But what if the WAP of interest was specifically configured *not* to send Beacon frames? That's not as big a problem for us as many people might think.

6.7.1.2 Filter on WAP-Announcing Management Frames

Let's use our tshark invocation from Section 6.3.3.1 to filter traffic and display only Beacon and Probe Response frames that have the ESS subfield set to 1 and the IBSS subfield set to 0. (Recall that, by specification, WAPs set these fields accordingly.) Even if a WAP is not broadcasting Beacon frames, it may still send Probe Responses to stations that initiate Probe Requests.

```
$ tshark -nn -r wlan.pcap -R '((wlan.fc.type_subtype == 0x08 || wlan.fc.
    type_subtype == 0x05) && (wlan_mgt.fixed.capabilities.ess == 1) && (
    wlan_mgt.fixed.capabilities.ibss == 0))'
  1    0.000000 00:23:69:61:00:d0 -> ff:ff:ff:ff:ff:ff 802.11 105 Beacon frame
       , SN=3583, FN=0, Flags=........, BI=100, SSID=Ment0rNet
265  20.409086 00:23:69:61:00:d0 -> 00:11:22:33:44:55 802.11 211 Probe
    Response, SN=3801, FN=0, Flags=........, BI=100, SSID=Ment0rNet
270  20.597504 00:23:69:61:00:d0 -> 00:11:22:33:44:55 802.11 211 Probe
    Response, SN=3804, FN=0, Flags=........, BI=100, SSID=Ment0rNet
335  23.318463 00:23:69:61:00:d0 -> 00:11:22:33:44:55 802.11 211 Probe
    Response, SN=3837, FN=0, Flags=........, BI=100, SSID=Ment0rNet
412  26.317951 00:23:69:61:00:d0 -> 00:11:22:33:44:55 802.11 211 Probe
    Response, SN=3873, FN=0, Flags=........, BI=100, SSID=Ment0rNet
[...]
```

- Electrical systems
- Laundry machines[21]
- Bathrooms[22]

Laundry Event Logging

As the Internet emerged in the mid-1990s, a student at MIT, Philip Lisiecki, got tired of having to walk all the way to the basement of his dormitory in order to check to see if a laundry machine was available. He decided to use photoresistors to monitor the laundry machines' indicator lights, and then rigged up the system to send data across the dormitory's old phone wiring.

"Once it was all running, everyone liked it." Mr. Lisiecki commented. "I could tell people used it since every time I turned my machine off for a half hour, someone with a laundry basket would wander by my room to find out what was wrong."[23]

Ultimately, laundry events were collected on a central server, laundry.mit.edu, and accessible over the World Wide Web. In 1999, the university newspaper ran a story on the system with the following report:

> Shortly after the laundry server was created, housemaster Nina Davis-Millis, an MIT information technology librarian, suggested that it be included in a New York Public Library exhibit on innovative uses of the Internet. Her friend, who was organizing the exhibit, included it in a proposal for the exhibit.
>
> "Her superiors were heartily displeased with her," said Ms. Davis-Millis. "They told her that she was too gullible, that she apparently was not familiar with the noble MIT tradition of hacking, but that it ought to have been obvious to her that hooking washers and dryers to the Internet was impossible." Thus, on the grounds that it couldn't be done, Random Hall's Internet laundry connection was not included in the NYPL Internet exhibit.
>
> To which Mr. Lisiecki replies, "They seem to have a fundamental misunderstanding of the Internet: nothing is too trivial."[24]

8.1.3.1 Example—Camera Logs

Below is an example of surveillance logs for an Axis camera system, generated by Zoneminder, an open-source, Linux-based "video camera security and surveillance solution" (http://www.zoneminder.com). The log sample below was kindly provided by Dr. Johannes Ullrich of the SANS Institute, who explained that the software "compares images and sends the alerts whenever the image comparison shows motion in the field of view."

---

21. Kevin Der, "Laundry Monitoring to Go Online for All Dormitories," *The Tech*, March 7, 2006, http://tech.mit.edu/V126/N9/9laundrytext.html.

22. Riad Wahby, "Random Hall Bathroom Server," 2001, http://bathroom.mit.edu/.

23. Robert J. Sales, "Random Hall residents monitor one of MIT's most-washed web sites—MIT News Office," April 14, 1999, http://web.mit.edu/newsoffice/1999/laundry-0414.html.

24. *Ibid.*

```
Feb 27 04:04:49 enterpriseb zma_m7[5628]: INF [frontaxis: 86496 - Gone into
    alarm state]
Feb 27 04:04:50 enterpriseb zma_m7[5628]: INF [frontaxis: 86498 - Gone into
    alert state]
Feb 27 04:04:50 enterpriseb zma_m7[5628]: INF [frontaxis: 86499 - Gone back
    into alarm state]
Feb 27 04:04:50 enterpriseb zma_m3[5648]: INF [AxisPTZ: 91951 - Gone into
    alarm state]
Feb 27 04:04:51 enterpriseb zma_m3[5648]: INF [AxisPTZ: 91952 - Gone into
    alert state]
Feb 27 04:04:51 enterpriseb zma_m7[5628]: INF [frontaxis: 86501 - Gone into
    alert state]
Feb 27 04:05:23 enterpriseb zma_m3[5648]: INF [AxisPTZ: 91986 - Gone into
    alarm state]
Feb 27 04:05:24 enterpriseb zma_m7[5628]: INF [frontaxis: 86535 - Gone into
    alarm state]
Feb 27 04:05:25 enterpriseb zma_m7[5628]: INF [frontaxis: 86536 - Gone into
    alert state]
Feb 27 04:05:25 enterpriseb zma_m3[5648]: INF [AxisPTZ: 91992 - Gone into
    alert state]
```

8.1.3.2 Example—Uninterruptible Power Supply Logs

Since power failures can have catastrophic impacts on network availability, network administrators naturally want to control and monitor UPS systems remotely. Apcupsd is a mature, open-source package for controlling and monitoring APC-brand UPS systems.[25] It is supported on a wide variety of platforms, including UNIX and Linux-based systems, as well as most popular versions of Microsoft Windows.[26]

Below is an example of UPS logs generated by apcupsd. Many thanks to Dr. Johannes Ullrich for providing these sample logs.

```
Feb 13 03:26:22 enterpriseb apcupsd[2704]: Power failure.
Feb 13 03:26:25 enterpriseb apcupsd[2704]: Power is back. UPS running on
    mains.
Feb  2 13:52:09 enterpriseb apcupsd[2704]: Communications with UPS lost.
Feb  2 13:52:16 enterpriseb apcupsd[2704]: Communications with UPS restored.
Jan 29 23:30:28 enterpriseb apcupsd[2704]: Power failure.
Jan 29 23:30:31 enterpriseb apcupsd[2704]: Power is back. UPS running on
    mains.
Jan 13 09:08:51 enterpriseb apcupsd[2704]: Power failure.
Jan 13 09:08:55 enterpriseb apcupsd[2704]: Power is back. UPS running on
    mains.
Dec 30 17:16:32 enterpriseb apcupsd[2704]: Power failure.
Dec 30 17:16:35 enterpriseb apcupsd[2704]: Power is back. UPS running on
    mains.
```

25. "APC Product Information for Uninterruptible Power Supply (UPS)," 2011, http://www.apc.com/products/category.cfm?id=13.

26. Adam Kropelin and Kern Sibbald, "APCUPSD User Manual," *APC UPS Daemon*, January 16, 2010, http://www.apcupsd.com/manual/manual.html.

8.1.4 Network Equipment Logs

Enterprise-class network equipment can generate extensive event logs. Often these logs are designed to be sent to a remote server via syslog or SNMP because the network devices themselves have very limited storage capacity.

Network equipment can include, among other things:

- Firewalls
- Switches
- Routers
- Wireless access points

8.1.4.1 Example—Apple Airport Extreme Logs

Below is an example of event logs downloaded from an Apple Airport Extreme. Notice that these logs include association and dissassociation events, authentication logs, and records of accepted connections. Once again, the logs do not include a year.

```
Apr 17 13:01:29 Severity:5      Associated with station 00:16:eb:ba:db:01
Apr 17 13:01:29 Severity:5      Disassociated with station 00:16:eb:ba:db:01
Apr 17 13:01:29 Severity:1      WPA handshake failed with STA 00:16:eb:ba:db
    :01 likely due to bad password from client
Apr 17 13:01:29 Severity:5      Deauthenticating with station 00:16:eb:ba:db
    :01 (reserved 2).
Apr 17 13:01:30 Severity:5      Associated with station 00:16:eb:ba:db:01
Apr 17 13:01:30 Severity:5      Disassociated with station 00:16:eb:ba:db:01
Apr 17 13:01:31 Severity:5      Associated with station 00:16:eb:ba:db:01
Apr 17 13:01:34 Severity:5      Associated with station 00:16:eb:ba:db:01
Apr 17 13:01:34 Severity:5      Installed unicast CCMP key for supplicant
    00:16:eb:ba:db:01
Apr 17 13:13:01 Severity:5      Disassociated with station 00:16:cb:08:27:ce
Apr 17 13:13:01 Severity:5      Rotated CCMP group key.
Apr 17 13:40:03 Severity:5      Associated with station 00:16:cb:08:27:ce
Apr 17 13:40:03 Severity:5      Installed unicast CCMP key for supplicant
    00:16:cb:08:27:ce
Apr 17 13:40:43 Severity:5      Connection accepted from [fe80::216:cbff:fe08
    :27ce%bridge0]:51161.
Apr 17 13:40:45 Severity:5      Connection accepted from [fe80::216:cbff:fe08
    :27ce%bridge0]:51162.
Apr 17 13:40:45 Severity:5      Connection accepted from [fe80::216:cbff:fe08
    :27ce%bridge0]:51163.
Apr 17 13:49:18 Severity:5      Clock synchronized to network time server
    time.apple.com (adjusted +0 seconds).
Apr 17 13:57:13 Severity:5      Rotated CCMP group key.
```

For more details on network equipment logs, please see Chapter 9, “Switches, Routers, and Firewalls,” and Chapter 6, “Wireless: Network Forensics Unplugged.”

8.2 Network Log Architecture

The forensic quality of retained logs, and the strategies and methods for obtaining them, are strongly influenced by the environment's network log architecture. Disparate logs accumulated on a fleet of systems don't really help an enterprise security staff understand the "big picture" of what is happening on the network. Distributed logs also make it difficult for security staff to audit the past history of security-related events. Even worse for the investigator, it can become a nightmare to locate and obtain important evidence.

The answer to this problem is to centralize event logging in such a way that all events of interest are aggregated and can be correlated between multiple sources. It may not be the case that the target environment is instrumented in such a way, but we'll discuss ways that this can be achieved, either by IT staff in advance or on-the-fly to facilitate an investigation.

8.2.1 Three Types of Logging Architectures

There are essentially three types of log architectures: local, remote decentralized, and centralized.

8.2.1.1 Local

Logs are collected on individual local hard drives. This is extremely common because it is the default configuration for most operating systems, applications, physical devices, and network equipment. However, local log aggregation presents issues for forensic applications, such as:

- Collecting logs from different systems can be a lot of work. In some cases, log collection causes modification of the local system under investigation, which is certainly not desirable.
- Logs stored locally on a compromised or potentially compromised system may be modified or deleted. Even if there is no evidence to indicate modification, logs stored on compromised systems cannot be trusted.
- Time skew on disparate local systems is often significant, and can make it very difficult to correlate logs and create valid timelines.
- Typically, logs stored on local systems are not centrally configured, and the output formats may vary between systems (or may only include sparse, default log data).
- Only a limited amount of logs may be stored to conserve local disk space.

8.2.1.2 Remote Decentralized

Logs are sent to different remote storage systems throughout the network. Different types of logs may be stored on different servers. This is commonly seen in environments where there is decentralized management of IT resources, such as in universities where individual departments or labs manage their own small groups of servers.

- Remote storage of logs increases their forensic value. When logs are sent to a remote system, they are far less likely to be affected by a local system compromise (at the

very least, they cannot be altered or modified after they are sent, unless the logging server is compromised as well).

- Time skew can be partially mitigated by having the logging servers timestamp incoming logs, although time skew between servers may still be an issue.
- Collecting logs from a logging server is usually far less work than collecting logs from endpoint devices, especially since the logging server is more likely to be under direct administrative control. That said, collecting logs from different log servers may still require substantial effort and coordination between teams.
- Sending logs to a remote server across the network introduces new challenges. Namely, reliability is a primary concern. If there is a network outage, logs may be dropped and lost forever. Security is also a concern; when transmitted in cleartext, as is most common, an attacker on the local network may be able to intercept, read, and perhaps even modify logs in transit. These issues can be addressed through the use of protocols that provide support for reliability such as TCP or RELP and encryption protocols such as TLS. However, configuring support for security features can be cumbersome, and network administrators in decentralized environments often do not have the resources to address these issues.

8.2.1.3 Centralized

Logs are centralized and aggregated on a central log server or a group of synchronized, centrally managed log servers. For the purposes of network forensics, a centralized logging infrastructure is typically the most desirable, for the following reasons:

- Logs are stored on a remote server, where they are not subject to modification or deletion in the event of an endpoint device compromise.
- Time skew can be addressed by stamping incoming logs as they arrive. Furthermore, when logging configuration is centralized, endpoint devices can be configured to maintain synchronized time and include granular time information in log output (so long as the endpoint device software supports these features).
- Centralized management typically allows for easy access to log data, and also facilitates on-the-fly configuration changes when needed to support an ongoing investigation.
- Issues of reliability and security of logs in transit can be centrally addressed. Network administrators can configure support for TCP, RELP, TLS, and other security features in central logging servers and centrally controlled clients.
- Aggregated logs can be easily analyzed using centralized log aggregation and analysis tools. (Please see Section 8.2.3 for details.)

As discussed previously, many network devices do not have sufficient storage capacity to maintain extensive forensic data. Fortunately, most network devices and conventional servers can be configured to send logs to a remote server that can aggregate forensic data from many sources. Central logging servers are simply servers configured to receive and store logs sent by other systems. They often store logs from many sources, including routers, firewalls, switches, and other servers. This helps system administrators keep tabs on many systems, and it enables investigators to find a wealth of data in one place.

The evidence stored on a central logging server varies greatly, depending on what systems were sending logs to it. Typically, you will find logs from many servers and workstation operating systems that were previously sent to the central logging server for storage and analysis. It is also common to find firewall logs, which include dates, times, source, destination, and protocols of the packets being logged.

8.2.2 Remote Logging: Common Pitfalls and Strategies

Automated remote logging is generally considered best practice in the log management industry. However, from a forensic perspective, there are potential pitfalls to keep in mind, and ways that investigators can compensate.

When event logs are sent across the network to a central server, they are placed at risk of loss or modification in transit. In addition, forensic investigators must consider issues such as time skew and confidentiality of the event logs in transit. Here is a brief discussion of major factors to consider when remote event logging is employed in a network forensic investigation, including reliability, time skew, confidentiality, and integrity.

8.2.2.1 Reliability

Can logs be lost as they are transmitted across the network? Frequently the answer is "yes." For example, clients that rely on the traditional syslog daemon to send logs across the network must rely on UDP as a transport-layer protocol. UDP is a connectionless protocol that does not include support for reliable transport. When a syslog message is transmitted across the network via UDP, if the datagram is dropped in transit, the server will have no record of it and the client will not know to retransmit. UDP datagrams are also commonly dropped when the receiving application is overloaded due to a high volume of traffic.

For forensic investigators, reliability of event log communication is an important issue. With unreliable event logging architectures, it is possible for an attacker to execute a denial-of-service attack or initiate a network outage in order to prevent critical information from being logged on a central server. Accidental loss is also a problem. While investigators may be able to piece together a timeline of events from existing logs, if there is a chance that critical details are missing, the investigation may fail or the case may fall apart in court.

To address the issue of reliability, offshoots of the syslog daemon have added native support for transport of syslog messages over TCP. TCP is a connection-oriented protocol with built-in support for reliability, so if a packet is dropped in transit, the server will notice a missing sequence number or the client will not receive an acknowledgment of transmission and will resend.

Although TCP improves reliability at the transport layer, there are still higher-layer issues. Rainer Gerhards, author of rsyslog, has published a nice article where he discusses how local buffering of TCP packets on the client system can lead to dropped syslog messages in the event of a network or server outage.[27] To address this issue, he developed the lightweight RELP,[28] which is designed to ensure reliable transfer of syslog messages at a higher layer.

---

27. Rainer Gerhards, "Rainer's Blog: On the (un)reliability of plain tcp syslog . . . ," April 2, 2008, http://blog.gerhards.net/2008/04/on-unreliability-of-plain-tcp-syslog.html.

28. Rainer Gerhards, "RELP—The Reliable Event Logging Protocol (Specification)," March 19, 2008, http://www.librelp.com/relp.html.

8.2.2.2 Time Skew

Time skew between endpoint systems is one of the biggest challenges for forensic investigators. It is difficult, if not impossible, to correlate logs between endpoint systems when local clock times (and therefore event log timestamps) are off. Even when the time skew between systems can be determined for a specific point in time, the clock on an endpoint system may have been running slower or faster at different points.

The best way to manage this problem is to synchronize clocks on all systems using NTP or a similar system. This can prevent problems due to clock skew during subsequent log analysis. Not all devices support time synchronization, however. Another option is for the central event logging server to add a timestamp to logs as they arrive. While this can be useful, it does not take into account network transit time; there is always a delay between the time that logs are generated on the endpoint system and the time that the logs are received by a remote logging server.

Logging output formats may not include enough information to properly correlate timestamps between different systems. For example, as we have seen, often the year is not included by default in event logging output. Furthermore, the time zone is also typically not included by default, which can make it very difficult for investigators to correlate logs between systems located in geographically dispersed areas. When configuring log output formats for potential forensic use, make sure to include complete, high-precision timestamps with time zone information.

8.2.2.3 Confidentiality

You might not expect that maintaining the confidentiality of event logs is important, but event logs can reveal extensive amounts of information about user habits, system software and directories, security issues, and more (this is why they are so highly valuable for forensics!). Anyone with access to the LAN (wired or wireless) or a device on the network path may be able to capture and analyze the traffic. To maintain the confidentiality of event logs in transit, use a protocol such as TLS/SSL that ensures the data is encrypted as it is transmitted across the network.

8.2.2.4 Integrity

Ensuring the integrity of event logs in transit is extremely important. By default, most remote logging utilities do not provide any assurance of integrity. Event logs transmitted over UDP or TCP without higher-layer encryption may be intercepted and modified in transit. Even worse, an attacker could inject fake event logs into the network traffic. This is quite easy to do for many types of remote logging servers, such as traditional syslog servers listening on a UDP port.

Fortunately, many event logging architectures now support TLS/SSL, either natively or through the use of tunneling proxies such as stunnel. You can use TLS/SSL to protect the data in transit and mutually authenticate the server and client event logging systems.

8.2.3 Log Aggregation and Analysis Tools

There are many tools available to facilitate log aggregation on central systems. Log aggregation tools typically work in a client-server model. Typically, an agent is installed

on the endpoint system (or, in some cases, a native tool may be able to export logs). A compatible central logging server is set up to listen on the network and receive logs as they are transmitted. Often, the central logging server software also includes powerful analysis capabilities.

Common agents installed on endpoints include:

- Syslog (and derivative) daemons, as previously discussed.
- System iNtrusion Analysis and Reporting Environment (SNARE)[29]—An open-source agent for Windows, Linux, Solaris, and more.

Central aggregation and analysis software includes:

- Splunk[30]—Log monitoring, reporting, and search tool.
- System Center Operations Manager (SCOM), formerly Microsoft Operations Manager (MOM)[31]—A monitoring and log aggregation product designed for Windows systems.
- Distributed log Aggregation for Data analysis (DAD)—An open-source log aggregation and analysis tool released under GPL.[32] (Figure 8–2 is a screenshot of the open-source DAD log analysis tool).
- Cisco's Monitoring, Analysis and Response System (MARS)[33]—Security monitoring for network devices and hosts (including Windows, Linux, and UNIX).
- ArcSight[34]—Commercial third-party log management and compliance solutions.

8.2.3.1 Splunk

Splunk is a proprietary, portable, highly extensible log aggregation and analysis tool. Figure 8–3 shows an example of Splunk. We'll revisit Splunk several times throughout this book because it's inexpensive (free for individual use up to 500 MB/day), versatile, scalable, and popular.

Splunk has a web-based interface and a database on the back end. It can accept input in a variety of forms, from reading a flat file to directly receiving syslog data over the network. Once Splunk has processed the data, you can run searches and reports.[35]

---

29. "Snare—Audit Log and EventLog analysis," 2011, http://www.intersectalliance.com/projects/index.html.

30. "Splunk | Operational Intelligence, Log Management, Application Management, Security and Compliance," 2011, http://www.splunk.com.

31. "System Center Operations Manager," *Wikipedia,* June 23, 2011, http://en.wikipedia.org/wiki/Microsoft_Operations_Manager.

32. D. Hoelzer, "DAD," *SourceForge*, June 29, 2011, http://sourceforge.net/projects/lassie/.

33. "Cisco Security Monitoring, Analysis, and Response System," *Wikipedia,* October 19, 2010, http://en.wikipedia.org/wiki/Cisco_Security_Monitoring,_Analysis,_and_Response_System.

34. "ArcSight," *Wikipedia,* July 14, 2011, http://en.wikipedia.org/wiki/ArcSight.

35. "Splunk | Operational Intelligence, Log Management, Application Management, Security and Compliance," 2011, http://www.splunk.com.

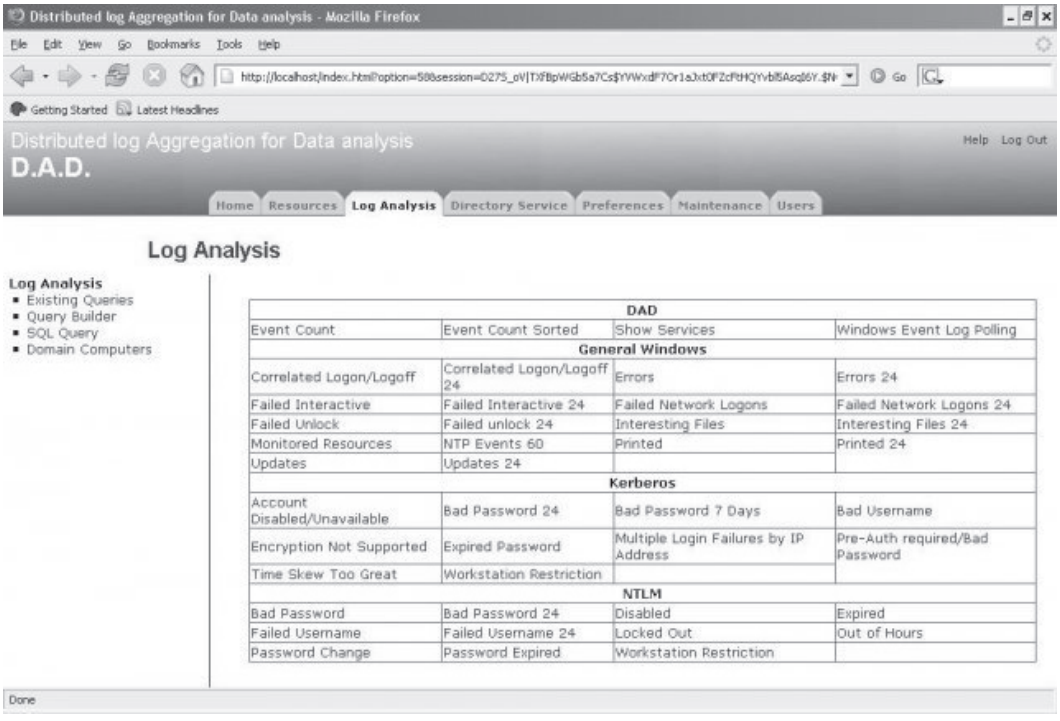

Figure 8–2. A screenshot of the DAD open-source log aggregation and analysis tool. Image courtesy of D. Hoelzer. Reprinted with permission.[36]

8.3 Collecting and Analyzing Evidence

Since the topic of network forensics relating to event logs is so broad, we'll use this as an opportunity to review and reinforce our network forensics methodology, OSCAR.

8.3.1 Obtain Information

When collecting and analyzing event logs, here is some specific information you may need to obtain:

- **Sources of Event Logs** Identify sources of event logs that are likely to relate to your investigation. You can accomplish this by conducting interviews with key personnel, reviewing network architecture documents, and reading IT policies and procedures that pertain to the environment under investigation. You will want to answer questions such as:
  - What event logs exist?
  - Where are they stored?

36. "dbimage.php (JPEG Image, 640x463 pixels)," http://sourceforge.net/dbimage.php?id=92531.

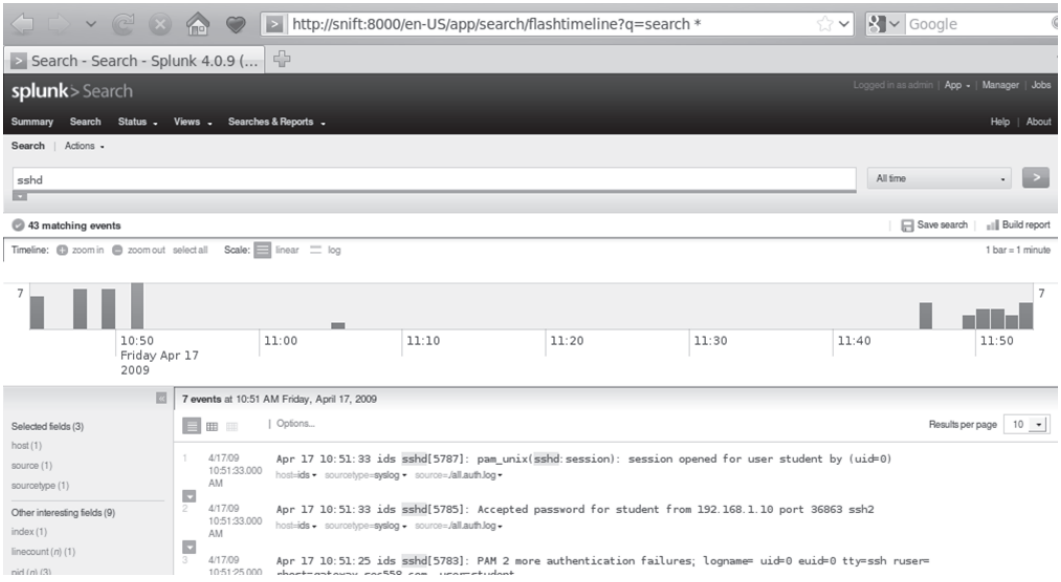

Figure 8–3. A simple example showing SSH service authentication logs in Splunk.

  - What are my technical options for accessing them?
  - Who controls the event logs?
  - How do we go about getting permission and access to collect them?
  - How forensically sound are the event logs?
  - Do the targeted systems have the capacity for additional logging to be configured?

- **Resources** Identify the resources you have available for event log collection, aggregation, and analysis. This includes equipment, communications capacity, time, money, and staff. For example, if you only have a 1TB hard drive for event log evidence storage, but there are 20TB of logs on the central logging server under investigation, you will either need to purchase more storage space or select a subset of logs to gather. Similarly, if you must collect the logs remotely but the network latency is high, this can limit the amount of data you are able to transfer in the time you have available. Questions to consider include:
  - How much storage space do I have available?
  - How much time do I have for collection and analysis?
  - What tools, systems, and staff are available for collection and analysis?
- **Sensitivity** For network-based investigations in particular, you have to consider how the sources of evidence and network itself will be impacted by evidence collection. Some equipment, such as routers and firewalls, may be under heavy load and operating close to processor/memory/bandwidth capacity. Retrieving evidence from

these systems may cause network or equipment slowness or outages, depending on the chosen method of collection. You will need to answer questions such as:

- How critical are the systems that store the event logs?
- Can they be removed from the network?
- Can they be powered off?
- Can they be accessed remotely?
- Would copying logs from these systems have a detrimental impact on equipment or network performance? If so, can we minimize the impact by collecting evidence at specific times or by scheduling downtime?

8.3.2 Strategize

In most enterprises, there are so many sources of event logs that taking the time to strategize is crucial. Otherwise, you may find that you run out of time or hard drive space before you have gathered the most important evidence, or you may overlook a valuable source of information.

As part of the "strategize" phase, review the information you've obtained, list and prioritize sources of evidence, plan the acquisition, and communicate with your team and enterprise staff.

8.3.2.1 Review Information

Once you've finished obtaining information, take the time to review all the information you have regarding the investigation. This may include:

- Goals and time frame of the investigation (very important!). It is worth reviewing your goals regularly during the investigation so that you can maintain perspective and stay on track.
- Potential sources of evidence.
- Resources available to you, such as hard drives for storing copies of event logs, secure storage space, staff, forensics workstations, and time.
- Sensitivity of networks and equipment that may be affected.

8.3.2.2 Prioritize Sources of Evidence

Acquiring evidence is expensive—literally. Every byte of data you copy takes time to transfer and uses up hard drive space. If you're acquiring evidence over a network, copying log files can use up a large amount of bandwidth and slow down the network. Furthermore, the more evidence you acquire, the more data you have to sift through later during the analysis phase.

In any organization, there are likely to be an overwhelming number of possible sources of event logs, including workstations, servers, switches, routers, firewalls, NIDS/NIPS, access control systems, web proxies, and more. Usually, only a small percentage of these logs contain evidence relevant to your investigation. In order to use your resources efficiently, review the

list of possible sources of evidence and identify those that are likely to be of the highest value to you.

Next, consider how much effort is required to obtain each source of evidence. When logs are centralized, it is usually fairly straightforward to gather copies of them. However, when logs are distributed on a variety of systems—such as hundreds of workstations or application servers managed by different departments—then technical or political hurdles can dramatically slow down the process. It is important to take these factors into consideration and anticipate challenges so that you can plan and budget accordingly.

After you've decided which sources of evidence are the most important, and estimated the resources required to obtain them, prioritize your evidence collection so that you can realize the greatest value from your efforts.

8.3.2.3 Plan Acquisition

In order to actually obtain copies of event logs, you will likely need to work with system administrators that manage the equipment on which the event logs reside. Before you actually set foot onsite to acquire the evidence, work with your primary contact to determine who can best provide you with access to the evidence. Then, plan your method for acquisition. Will you have physical access to the system, or will you acquire evidence remotely? When and where will you acquire the evidence? The time of day may be especially important if the investigation must remain secret, or if the equipment that stores the evidence is under heavy load at certain hours.

8.3.2.4 Communicate

No investigator is an island. Once you have developed a plan (usually in conjunction with your investigative team and local contacts), make sure to communicate the final plan to everyone involved. Agree on a method and times for regular communication and updates, such as daily emails or weekly conference calls.

8.3.3 Collect Evidence

The method you use for collecting event log evidence will vary depending on the environment's event logging architecture, your sources of evidence, and your available resources (among other factors). Potential methods include physical connection, manual remote connection, central log aggregation, and passive evidence acquisition.

8.3.3.1 Physical Connection

For logs stored locally on endpoint devices, you may choose to create a bit-for-bit forensic image of the physical storage media (such as a hard drive), and extract event log files directly from it using traditional hard drive forensic techniques. The benefits of this method are that you can retain an exact copy of the drive for later presentation in court (if necessary), and that from a forensics perspective, there are widely accepted standards for the process of forensic hard drive analysis.

However, if the event logs of interest are stored on more than a few endpoint systems, it may be simply impractical to invest in the time and equipment necessary to forensically image multiple drives. Another major drawback is that logs stored locally are at higher risk

of modification in the event of system compromise, and as a result are often considered less forensically valuable than logs stored on remote systems.

For logs stored on a central logging server, it is sometimes appropriate to take a bit-for-bit forensic image of the logging server's hard drive. Again, this has the benefit of allowing a forensic copy of the server's drive to be preserved and presented later. It can also allow for a very detailed analysis of logging server configuration. Supplemental information such as precise versions of event logging software can be helpful for later analysis.

Commonly, network forensic investigators simply copy the logfiles off either an endpoint system or a central logging server using a physical port (i.e., eSATA or USB). This has the strong advantage of having a relatively low impact on system resources (i.e., copying files takes far less time, storage space, and I/O than making a bit-for-bit forensic duplicate of the drive). In addition, the system does not need to be taken offline or powered down in order to copy files. If you use this method, make sure to capture cryptographic checksums of the source and destination files to ensure that you have made an accurate duplicate.

Physical collection of event logs is also useful when you want to minimize the network footprint of the investigation.

8.3.3.2 Manual Remote Connection

You may prefer to collect logs through manual remote examination of endpoint devices using services such as SSH, RDP, or an administrative web page. The benefits of this method are that it may enable you to examine systems that are geographically farther away than you could access otherwise, and it may also enable you to collect logs directly from many more sources than you could otherwise.

One drawback of manual remote collection is that you will modify the system under examination simply by accessing it remotely (it is even possible to cause log rollover simply by logging into the device, if the logging system has reached a preset limitation on storage space). You will create network activity through the process of manual remote examination, which can also contribute to network congestion. Make sure you are aware of bandwidth and throughput limitations before transferring large quantities of event logs across the network.

8.3.3.3 Central Log Aggregation

If you are lucky, the event logs are already being sent to a central logging server (or a synchronized group of central logging servers). In this case, you will want to begin by researching the underlying log collection architecture to ensure that it is forensically sound and will meet your needs for evidence collection. For example, you should know the transport-layer protocol in use for log transmission, as well as mechanisms for authentication of logging client and server and encryption of data in transit, to determine the risk of event log loss or modification.

You can access the evidence on a central logging server in multiple ways, depending on how it is set up:

- **Console** Log onto the central logging server using SSH, RDP, or direct console connection, depending on the specific configuration. Browse files, copy specific logs for later analysis, burn them onto a CD, or simply view them.

- **Web interface** Many organizations use a log analysis tool such as Splunk, which facilitates centralized log analysis. Often, these include helpful web interfaces, with search and report-generating capabilities that can be extremely useful for identifying suspicious activity and correlating logs.
- **Proprietary interface** Some logging servers are accessed using proprietary client software, which provide graphical analysis/report capabilities.

In certain situations, you may choose to take a forensic image of the central logging server's hard drive(s). This can be very resource-intensive. See "Physical Collection," above, for details.

8.3.3.4 Passive Evidence Acquisition

In some cases, you may want to collect event logs as they are transmitted across the network through passive evidence acquisition techniques (please see Chapter 3, "Evidence Acquisition," for details). This is effective in environments where you have access to the network segments over which the event log data is transmitted, and when the log data is not encrypted in transit (or in the rare situation where you have the ability to decrypt the log data in transit). Passive evidence acquisition may be your best option for event log collection in an environment where the IT staff are either unaware of your investigation or uncooperative.

8.3.4 Analyze

Strategies for conducting event log analysis are as varied as the sources of event logs themselves and the goals of specific investigations. For discussions of event log analysis relating to specific types of logs, please see Chapter 10, "Web Proxies"; Chapter 9, "Switches, Routers, and Firewalls"; and Chapter 7, "Network Intrusion Detection and Analysis."

General techniques include:

- Dirty Values—Searching for specific keywords in logs.
- Filtering—Narrowing down your search space by selecting logs based on time, source/destination, content, or other factors.
- Activity Patterns—Analyzing logs for patterns of activity and identifying suspicious activity based on the results.
- Fingerprinting—Creating a catalog of complex patterns and correlating these with specific activities to facilitate later analysis.

Figure 8–3 shows an example of analysis using Splunk. In this case, we have searched for all logs containing the word "sshd." This effectively filters the logs so that they only include information relating to the SSH remote login service. We can see the results graphically represented, and can click on any time to view the logs in detail. You can see that there were seven results for our search at 10:51 AM on Friday, April 17 2009. These logs appear to be attempts to SSH into the account "student" on the server "ids." At first the SSH attempts failed, but at 10:51:33 there was a successful login to "student" from 192.168.1.10.

Based on these results, our next step might be to examine the patterns of activity specifically relating to the "student" account on any system. Perhaps the "student" account was compromised through a password-guessing attack—or perhaps the user had simply forgotten the password temporarily. We could also examine all logs relating to the "ids" system to see if there was any further evidence of suspicious behavior.

Analysis tools are not perfect! Notice that Splunk listed a year (2009) in Figure 8–3. However, there is no year in the original syslog event logs—just a month, day, and time. Analysis tools can sometimes produce unexpected or incorrect results. Whenever possible, correlate events using multiple sources of evidence, and confirm findings by checking original evidence.

8.3.5 Report

Event logs are frequently used as the basis for conclusions drawn in reports. Here are a few good tips for incorporating evidence from event logs into your forensic reports:

- A picture is worth a thousand words. It is always a good idea to include graphical representations of event log analysis when you have the option. Charts and graphs generated by Splunk and similar tools can be very powerful.
- Make sure to include detailed information regarding your sources of event logs and your process for collecting them. Generally this is appropriate for an appendix of the report or supplemental materials.
- Remember to include information regarding your methodology and the analysis tools you used. This is especially important because analysis tools are not perfect. The more widely known and tested your tools, the more likely they are to be accepted in a courtroom setting.
- Always retain and reference your original sources of evidence so that you can support your reported findings.

8.4 Conclusion

Event logs are some of the most valuable sources of evidence for forensic investigators, particularly when they are stored on a secure central server and can be correlated with multiple log sources. Application servers, firewalls, access control systems, network devices, and many other types of equipment generate event logs and are often capable of exporting them to a remote log server for aggregation.

It is important for the forensic investigator to be aware of common pitfalls associated with event log analysis, including incorrect or incomplete timestamps, questions of reliability and integrity, and confidentiality. With these in mind, event logs are an important source of evidence, and can be analyzed with a variety of command-line or visual tools.

8.5 Case Study: L0ne Sh4rk's Revenge

The Case: *Inspired by Mr. X's successful exploits at the Arctic Nuclear Fusion Research Facility, L0ne Sh4rk decides to try the same strategy against a target of his own: Bob's Dry Cleaners! The local franchise destroyed one of his favorite suits last year and he has decided it is payback time. Plus, they have a lot of credit card numbers.*

Meanwhile ... *Unfortunately for L0ne Sh4rk, Bob's Dry Cleaners is on the alert, having been attacked by unhappy customers before. Security staff notice a sudden burst of failed login attempts to their SSH server in the DMZ (10.30.30.20), beginning at 18:56:50 on April 27, 2011. They decide to investigate.*

Challenge: **You are the forensic investigator.** Your mission is to:

- Evaluate whether the failed login attempts were indicative of a deliberate attack. If so, identify the source and the target(s).
- Determine whether any systems were compromised. If so, describe the extent of the compromise.

Bob's Dry Cleaners keeps credit card numbers and personal contact information for their Platinum Dry Cleaning customers (many of whom are executives). They need to make sure that this credit card data remains secure. If you find evidence of a compromise, provide an analysis of the risk that confidential information was stolen. Be sure to carefully justify your conclusions.

Network: Bob's Dry Cleaners network consists of three segments:

- Internal network: 192.168.30.0/24
- DMZ: 10.30.30.0/24
- The "Internet": 172.30.1.0/24 [Note that for the purposes of this case study, we are treating the 172.30.1.0/24 subnet as "the Internet." In real life, this is a reserved nonroutable IP address space.]

Evidence: Security staff at Bob's Dry Cleaners collect operating system logs from servers and workstations, as well as firewall logs. These are automatically sent over the network from each system to a central log collection server running rsyslogd (192.168.30.30). Security staff have provided you with log files from the time period in question. These log files include:

- **auth.log**—System authentication and privileged command logs from Linux servers
- **workstations.log**—Logs from Windows workstations
- **firewall.log**—Cisco ASA firewall logs

Security staff also provide you with a list of important systems on the internal network:

Hostname	Description	IP address(es)
ant-fw	Cisco ASA firewall	192.168.30.10 10.30.30.10 172.30.1.253
baboon-srv	Server running SSH, NTP, DNS	10.30.30.20
cheetah-srv	Server running rsyslogd	192.168.30.30
dog-ws	Workstation	192.168.30.101
elephant-ws	Workstation	192.168.30.102
fox-ws	Workstation	192.168.30.100
yak-srv	Server	192.168.30.90

8.5.1 Analysis: First Steps

Let's begin by examining the logs relating to the failed login attempts. Based on reports from security staff, we know that the activity began at 18:56:50 and targeted 10.30.30.20, which corresponds with the hostname "baboon-srv." Since this is a Linux server, let's browse for corresponding logs in the auth.log evidence file. The first failed login attempts we see are as follows:

```
2011-04-26T18:56:50-06:00 baboon-srv sshd[6423]: pam_unix(sshd:auth):
    authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost
    =172.30.1.77  user=root
2011-04-26T18:56:53-06:00 baboon-srv sshd[6423]: Failed password for root
    from 172.30.1.77 port 60372 ssh2
2011-04-26T18:56:56-06:00 baboon-srv sshd[6423]: last message repeated 2
    times
2011-04-26T18:56:56-06:00 baboon-srv sshd[6423]: PAM 2 more authentication
    failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=172.30.1.77  user=
    root
```

From these records, we see that the remote host 172.30.1.77 attempted to login to the SSH server on baboon-srv targeting the account "root." The "root" account is the default administrative user on most Linux/UNIX systems. This is a very common target for brute-force attacks, and a failed remote login attempt is certainly suspicious.

8.5.2 Visualizing Failed Login Attempts

Note that each initial "authentication failure" log is followed by additional entries that indicate that there were two more failed login attempts. It's important to remember that failed login attempts are not recorded individually, but are instead recorded as a series of event logs in the pattern above.

Next, let's use a visualization tool to get a better picture of the volume and time frame of the failed login attempts. Figure 8–4 is a screenshot of Splunk showing all activity from auth.log from the host "baboon-srv." As you can see, the bulk of the activity occurred between 18:56 and 19:05.

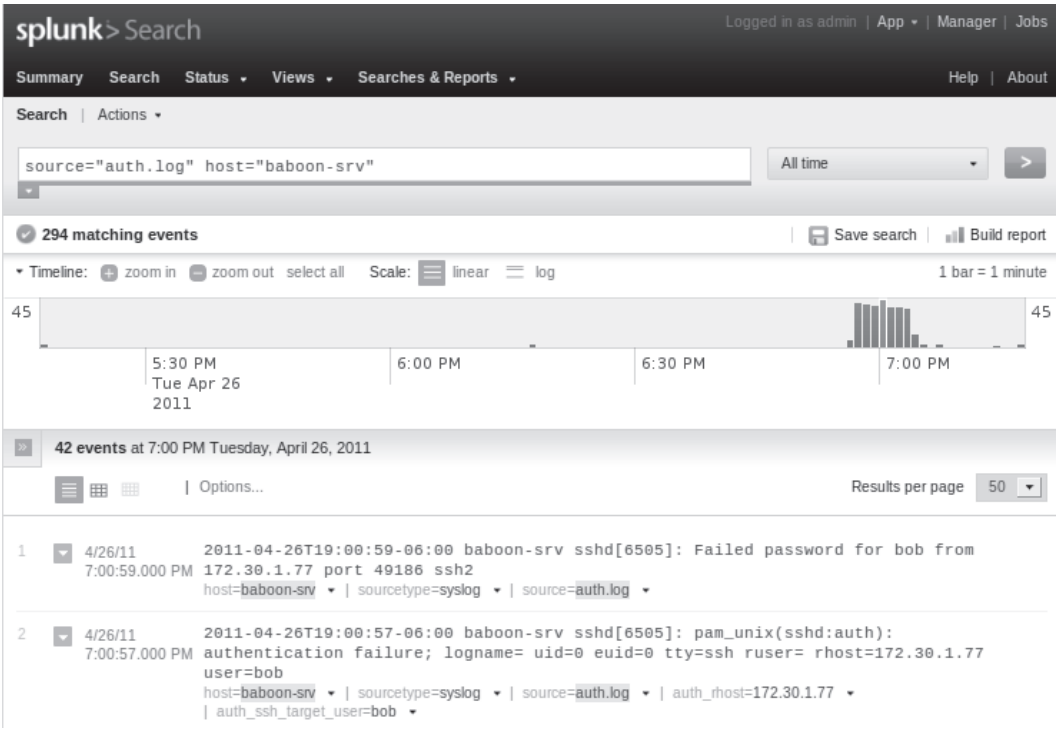

Figure 8–4. A chart in Splunk showing all activity from auth.log relating to "baboon-srv." The bulk of the activity occurs between 18:56 and 19:05.

After importing our log files into Splunk, we can use regular expressions to define specific fields in the logs that are of interest to us, such as a field named "auth_rhost," which specifies the source of the remote login attempt (see "rhost=" in the SSH event log). Zooming in on our time frame of interest, we can select each field, filter on it, and view statistics. Figure 8–5 shows remote SSH login attempts between 18:56 and 19:06, with the auth_rhost field selected. As you can see, only one remote host attempted to login to baboon-srv, and that was 172.30.1.77.

Drilling down even further, we see that the login attempts have a distinct, regular pattern. Figure 8–6 shows a closeup of SSH remote login attempts during just one minute (18:57:00–18:57:59). As you can see, there are two events logged approximately every six seconds, with only slight variation. The corresponding events, shown below the chart, are a record of one failed remote login attempt, followed by a record of two more failed remote login attempts (these are the only event logs that contain the "auth_rhost" field, which we have filtered on). This means there are a total of three failed login attempts every six seconds, for an average of one login attempt every two seconds.

The regularity of these failed login attempts is a strong indicator that the remote system is running a brute-force password-guessing attack utility, such as "medusa." Such utilities are designed to use a password dictionary to attempt to guess a login password for a remote

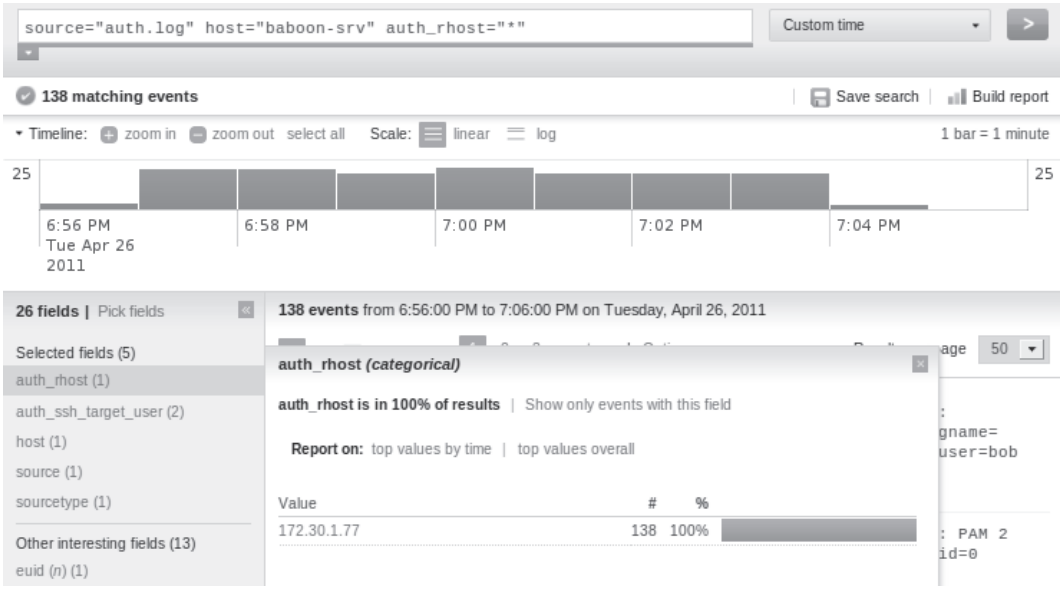

Figure 8–5. A screenshot of Splunk showing remote SSH login attempts between 18:56 and 19:06, with the auth_rhost field selected. There is only one remote host attempting to login to baboon, and that is 172.30.1.77.

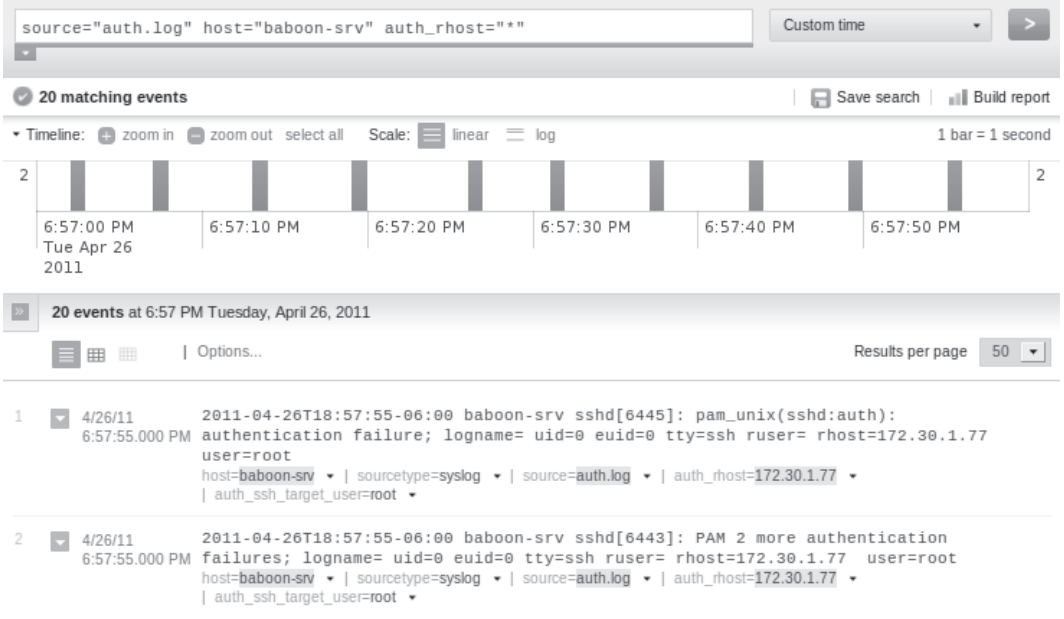

Figure 8–6. A screenshot of Splunk showing SSH remote login attempts during just one minute (18:57:00–18:57:59). Note the regular pattern of two log events every six seconds, which after careful examination of the logs translates to an average of one login attempt every two seconds.

system. The attack utility is typically configured to run either until the attack is successful or the wordlist is exhausted. Since the SSH server needs time to process each login attempt, brute-force utilities are commonly set to space login attempts by at least one to three seconds, or longer if the attack is intended to be slow and stealthy.

8.5.3 Targeted Accounts

Now that we have clear indication of a brute-force password-guessing attack against the SSH server running on baboon-srv, the next questions are: What accounts were targeted? Was the attack successful?

In Splunk, let's also define a field called "auth_ssh_target_user," which contains the username targeted in the remote SSH login attempts (see the "user=" tag in the SSH event logs). We can simply select that field in Splunk and view statistics relating to event logs that contain this field. Figure 8–7 shows that only two accounts were targeted, "root" and "bob," along with relative percentages of the logs that contain authentication failure messages relating to each account.

To generate these statistics, we filtered only on event logs containing "auth_ssh_target_user," which matches events of the following formats:

```
2011-04-26T18:57:19-06:00 baboon-srv sshd[6433]: pam_unix(sshd:auth):
    authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost
    =172.30.1.77  user=root
2011-04-26T18:57:26-06:00 baboon-srv sshd[6433]: PAM 2 more authentication
    failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=172.30.1.77  user=
    root
```

As you can see, there are two types of matching events, one that records one login attempt, and the other that records two login attempts. We can use the "grep" and "wc" shell commands to quickly count the number of each type of log for each of the targeted accounts, and calculate a total number of failed login attempts for each targeted account.

As shown in the results below, there were 41 + (2 * 40) = 121 failed login attempts for the "root" account:

```
$ grep "authentication failure" auth.log | grep "baboon-srv" | grep "user=
    root"  | grep -c "pam_unix(sshd:auth): authentication failure"
41
$ grep "authentication failure" auth.log | grep "baboon-srv" | grep "user=
    root"  | grep -c "PAM 2 more authentication failures"
40
```

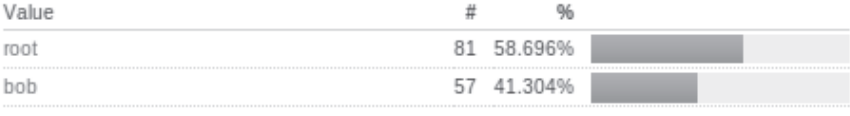

Figure 8–7. In Splunk, we defined a field called "auth_ssh_target_user," which contains the username targeted in the remote SSH login attempts. Only two accounts were targeted: "root" and "bob."

Likewise, there were 29 + (2 * 28) = 85 failed login attempts for the "bob" account.

```
$ grep "authentication failure" auth.log | grep "baboon-srv" | grep "user=bob
    "  | grep -c "pam_unix(sshd:auth): authentication failure"
29
$ grep "authentication failure" auth.log | grep "baboon-srv" | grep "user=bob
    "  | grep -c "PAM 2 more authentication failures"
28
```

We can also graph the number of event logs relating to each user over time, as shown in Figure 8–8. Notice that the failed login attempts for the "root" account occur first and are immediately followed by attempts to login to the account "bob." Again, this fits common activity patterns of brute-force password-guessing utilities, which are often configured with a list of usernames as input, and conduct attacks against each account in series.

8.5.4 Successful Logins

Now that we have strong evidence of a brute-force password-guessing attack, let's turn our attention to the question of whether the attack was successful.

In the auth.log file, the last failed SSH login attempt against baboon-srv is at 19:04:05, for the account "bob," as shown below:

```
$ grep "authentication failure" auth.log | grep "baboon-srv" | grep "sshd" |
    tail -1
2011-04-26T19:04:05-06:00 baboon-srv sshd[6561]: pam_unix(sshd:auth):
    authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost
    =172.30.1.77  user=bob
```

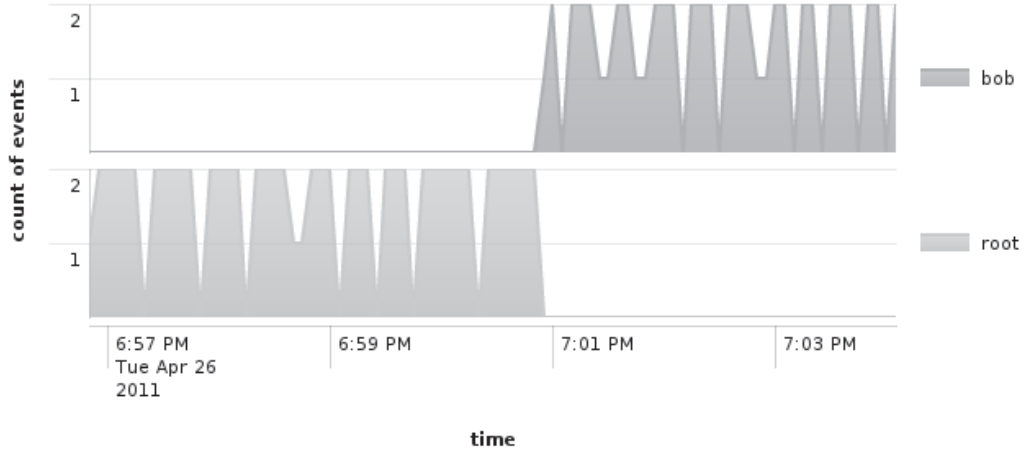

Figure 8–8. A graph created in Splunk, showing the number of event logs relating to each user over time. The failed login attempts for the "root" account occur first, and are immediately followed by attempts to login to the account "bob."

10.8 Case Study: Inter0ptic Saves the Planet (Part 2 of 2)

The Case: *In his quest to save the planet, Inter0ptic has started a credit card number recycling program. ''Do you have a database filled with credit card numbers, just sitting there collecting dust? Put that data to good use!'' he writes on his web site. ''Recycle your company's used credit card numbers! Send us your database, and we'll send YOU a check.''*

For good measure, Inter0ptic decides to add some bells and whistles to the site, too ...

Meanwhile ... *MacDaddy Payment Processor deployed Snort NIDS sensors to detect an array of anomalous events, both inbound and outbound. An alert was logged at 08:01:45 on 5/18/11 concerning an inbound chunk of executable code sent to port 80/tcp for inside host 192.168.1.169 from external host 172.16.16.218. Here is the alert:*

```
[**] [1:10000648:2] SHELLCODE x86 NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
05/18-08:01:45.591840 172.16.16.218:80 -> 192.168.1.169:2493
TCP TTL:63 TOS:0x0 ID:53309 IpLen:20 DgmLen:1127 DF
***AP*** Seq: 0x1B2C3517  Ack: 0x9F9E0666  Win: 0x1920  TcpLen: 20
```

We analyzed the Snort alert and determined the following likely events (see the case study in Chapter 7 for more details):

- From at least 07:45:09 MST until at least 08:15:08 MST on 5/18/11, internal host 192.168.1.169 was being used to browse external web sites, some of which delivered web bugs, which were detected and logged.
- At 08:01:45 MST, an external web server 172.16.16.218:80 delivered what it stated was a JPEG image to 192.168.1.169, which contained an unusual binary sequence that is commonly associated with buffer overflow exploits.
- The ETag in the external web server's HTTP response was:
  1238-27b-4a38236f5d880
- The MD5sum of the suspicious JPEG was:
  13c303f746a0e8826b749fce56a5c126
- Less than three minutes later, at 08:04:28 MST, internal host 192.168.1.169 spent roughly 10 seconds sending crafted packets to other internal hosts on the 192.168.1.0/24 network. Based on their nonstandard nature, the packets are consistent with those used to conduct reconnaissance via scanning and operating system fingerprinting.

Challenge: **You are the forensic investigator.** Your mission is to:

- Examine the Squid cache and extract any cached pages/files associated with the Snort alert shown above.
- Determine whether the evidence extracted from the Squid cache corroborates our findings from the Snort logs.

- Based on web proxy access logs, gather information about the client system 192.168.1.169, including its likely operating system and the apparent interests of any users.
- Present any information you can find regarding the identity of any internal users who have been engaged in suspicious activities.

Network: The MacDaddy Payment Processor network consists of three segments:

- Internal network: 192.168.1.0/24
- DMZ: 10.1.1.0/24
- The "Internet": 172.16.0.0/12 [Note that for the purposes of this case study, we are treating the 172.16.0.0/12 subnet as "the Internet." In real life, this is a reserved nonroutable IP address space.]

Other domains and subnets of interest include:

- .evl—a top-level domain (TLD) used by Evil systems.
- example.com—MacDaddy Payment Processor's local domain. [Note that for the purposes of this case study, we are treating "example.com" as a legitimate second-level domain. In real life, this is a reserved domain typically used for examples, as per RFC 2606.]

Evidence: You are provided with two files containing data to analyze:

- **evidence-squid-cache.zip**—A zipfile containing the Squid cache directory ("squid") from the local web proxy, www-proxy.example.com. Helpfully, security staff inform you that since MacDaddy Payment Processor's network connection has been slow, the web proxy is tuned to retain a lot of pages in the local cache.
- **evidence-squid-logfiles.zip**—Snippets of the "access.log" and "store.log" files from the local Squid web proxy, www-proxy.example.com. The access.log file contains web browsing history logs, and the store.log file contains cache storage records, both from the same time period as the NIDS alert.

10.8.1 Analysis: pwny.jpg

Lets begin by examining the Squid proxy cache for traces of the suspicious image that we found in Snort. The Squid header we received contained a pseudo-unique ETag value of "1238-27b-4a38236f5d880." Using Linux command-line tools, we can search the Squid cache and list the cache file that contains this ETag, as shown below:

```
$ grep -r '1238-27b-4a38236f5d880' squid
Binary file squid/00/05/0000058A matches
```

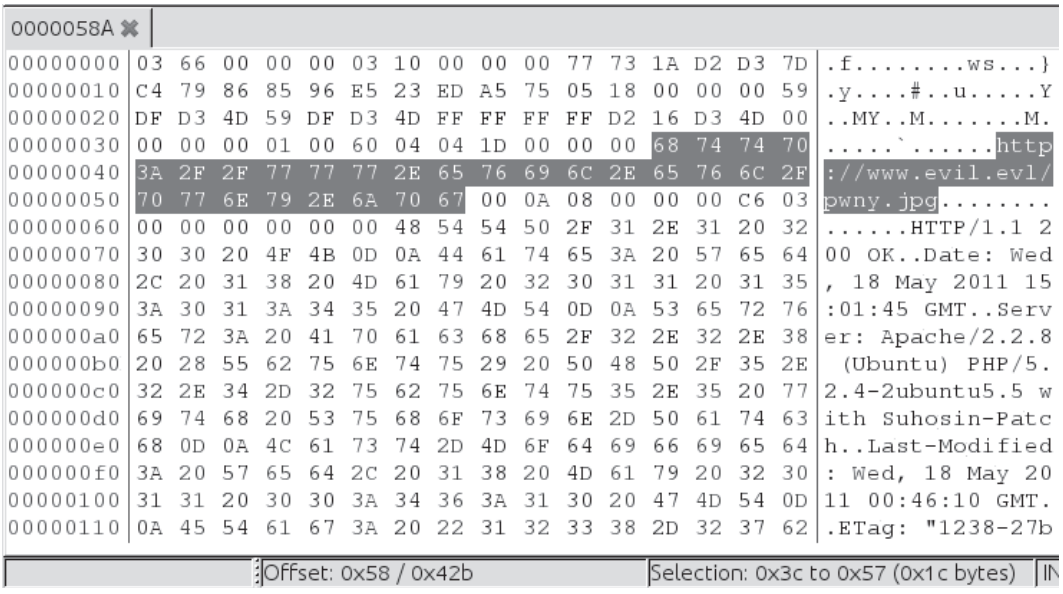

Figure 10–16. Opening the cached page in "Bless," we can find the URI of the requested cached object in the Squid metadata.

It appears that the page we're looking for is cached in the file "squid/00/05/0000058A." Opening the cached page in "Bless," we can find the URI of the requested cached object in the Squid metadata, as shown in Figure 10–16. It appears that the URI of the cached object was:

http://www.evil.evl/pwny.jpg

Immediately following the metadata are the HTTP headers, as follows:

```
HTTP/1.1 200 OK
Date: Wed, 18 May 2011 15:01:45 GMT
Server: Apache/2.2.8 (Ubuntu) PHP/5.2.4-2ubuntu5.5 with Suhosin-Patch
Last-Modified: Wed, 18 May 2011 00:46:10 GMT
ETag: "1238-27b-4a38236f5d880"
Accept-Ranges: bytes
Content-Length: 635
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: image/jpeg
```

These precisely match the HTTP headers within the packet we carved earlier from the Snort tcpdump.log file, as shown in Chapter 7. From these HTTP headers, we can deduce that this Squid cache file likely contains a JPEG image 635 bytes in length.

JPEG files begin with the magic number "0xFFD8," so we can simply search the Squid cache file for that hex sequence and cut everything before it, as you can see in Figure 10–17. We save this edited cache file as "0000058A-edited.jpg."