Weaknesses in Wireless LAN Session Containment

Joshua Wright
jwright@hasborg.com

5/19/2005

Abstract

This paper describes the characteristics of wireless LAN session containment techniques used to stop an unauthorized station from connecting to a monitored access point. Using the traffic analysis techniques described in this paper, an attacker can fingerprint the type of wireless LAN intrusion detection system deployed to monitor and protect the wireless network, and potentially evade the session containment functionality altogether.

Introduction

Session containment is a technique implemented by wireless LAN IDS vendors to prevent unauthorized stations from connecting to an authorized or rogue access point. Also known as wireless intrusion prevention, this functionality has been available for several years and has been gaining more widespread adoption by organizations seeking to mitigate wireless security vulnerabilities. Session containment has recently been described by one vendor as "an essential component of the layered approach to security" (AirDefense, 2005).

When a WLAN IDS identifies an unauthorized station on a wireless network, it may attempt to prevent the station from accessing network resources. This is accomplished by mounting a denial of service (DoS) attack against the rogue access point or station, leveraging weaknesses in the IEEE 802.11 specification to disconnect one or more users from the wireless network. When the disconnect message is repeated continuously, the rogue station is unable to connect to the wireless network, preventing a potential network intrusion.

This technique is applied to mitigate various wireless security vulnerabilities, including the threat of rogue access points and unauthorized stations. In the case of rogue access point protection, session containment is applied to prevent any station from connecting to the access point, by leveraging DoS attack mechanisms against all stations that attempt to connect to the rogue device. In the case of a rogue station connecting to an authorized access point, only the rogue station is targeted, preventing the disruption of other authorized user connections on the network. In both cases the result is the same; unauthorized users are stopped from connecting to the monitored access point.

While session containment may seem attractive as another layered wireless LAN defense mechanism, it is not free from other serious risks. Because the otherwise passive IDS sensor now becomes an active network participant, an attacker can use various techniques to gather additional information about the architecture and configuration of the network. In some cases, this information can be used to bypass the protection offered by session containment tools, a serious risk for organizations relying on this technology to mitigate wireless network risks.

This paper will review how session containment is implemented by assessing various vendor products, and identifying weaknesses observed these implementations.

Session Containment Implementations

When implementing a mechanism to disconnect users from a protected access point, vendors must consider several factors:

- Preventing unauthorized access. The goal of session containment against an unauthorized station is to stop access to the distribution system or wired network. The selection of a technique that reliably stops access to the network is a major consideration for the WLAN IDS vendor.
- Minimizing impact to the wireless spectrum or channel. A WLAN IDS vendor can easily prevent all access to a monitored access point by implementing a denial of service attack against the wireless spectrum, such as an RF jamming attack. This has the negative side-affect of preventing all access to the spectrum, including potentially authorized stations and access points that are accessing a nearby production network. A WLAN IDS vendor must implement a technique to disconnect unauthorized stations with minimal impact to other production wireless networks.
- Limiting DoS scope to designated stations. A vendor may opt to provide sufficient fidelity in their session containment implementation such that they can disconnect a single unauthorized station, preserving the connectivity of other authorized users. This requirement will also influence the implementation of the session disconnect technique.

Considering these implementation factors, vendors have implemented session containment by transmitting spoofed deauthenticate and/or disassociate management frames. By transmitting these frames with a spoofed source MAC address of the access point or victim station, a WLAN IDS vendor can force a client to disconnect from the network, forcing them to repeat the IEEE 802.11 authentication and association process to regain access to the network. By repeating the transmission of these frames, a WLAN IDS can sustain a DoS attack against a target MAC address, preventing access to the network.

The following trace is an example of one vendor's implementation of session containment against a rogue station:

```
1. 00:90:4b:2d:65:24  -> 00:12:17:9f:08:73  ICMP Echo (ping) request
2. 00:12:17:9f:08:73  -> 00:90:4b:2d:65:24  ICMP Echo (ping) reply
3. 00:12:17:9f:08:71  -> ff:ff:ff:ff:ff:ff  IEEE 802.11 Deauthentication
4. 00:90:4b:2d:65:24  -> ff:ff:ff:ff:ff:ff  IEEE 802.11 Probe Request, SSID: "linksys-a"
5. 00:12:17:9f:08:71  -> 00:90:4b:2d:65:24  IEEE 802.11 Probe Response, SSID: "linksys-a"
6. 00:12:17:9f:08:71  -> ff:ff:ff:ff:ff:ff  IEEE 802.11 Deauthentication
7. 00:90:4b:2d:65:24  -> 00:12:17:9f:08:71  IEEE 802.11 Authentication
8. 00:12:17:9f:08:71  -> 00:90:4b:2d:65:24  IEEE 802.11 Authentication
9. 00:90:4b:2d:65:24  -> 00:12:17:9f:08:71  IEEE 802.11 Reassociation Request, SSID:
   "linksys-a"
10.    00:12:17:9f:08:71  -> 00:90:4b:2d:65:24  IEEE 802.11 Reassociation Response
11.    00:12:17:9f:08:71  -> ff:ff:ff:ff:ff:ff  IEEE 802.11 Deauthentication
```

In this trace, an authenticated, associated station at 00:90:4b:2d:65:24 is exchanging ICMP echo request and response traffic with another station at 00:12:17:9f:08:73. After the ICMP exchange, a deauthenticate request is sent to the broadcast address from the access point at 00:12:17:9f:08:71, which causes the wireless station to reconnect to the network beginning with a probe request frame. A second deauthenticate notice is transmitted in frame 6. Since this frame is transmitted before the station reauthenticates to the network, it is silently ignored, and the station continues the authentication and reassociation process. The deauthenticate frame transmitted in frame 11 does successfully disconnect the client, forcing them to repeat the connect process.

In this case, the deauthenticate frames are transmitted by the WLAN IDS sensor with a spoofed source MAC address of the access point. This makes the station believe that the access point is disconnecting them from the network, forcing them to reconnect. Sustaining these spoofed frames will keep the station from being able to transmit on the network. This technique is employed by most vendors to implement session containment, with minor variations.

Detecting Containment Attempts

Even though WLAN IDS vendors utilize an IEEE 802.11 sanctioned technique to inform a rogue station that they have been disconnected from the network, the technique used to contain the station is anomalous. An attacker can use knowledge of how vendors implement session containment to identify the presence of a WLAN IDS system that is being used to restrict their access to the network.

Sequence Number Analysis

WLAN IDS vendors use various techniques to identify the presence of spoofed frames on the network. One method of detecting spoofed frames is to monitor the network for abnormalities in the selection of sequence numbers, as described in (Wright, 2003).

The sequence number field is a 12-bit counter present in every management and data frame transmitted on an IEEE 802.11 wireless network. This field is designed to associate multiple fragments with a single frame. As a modulo 4096 sequential counter, each frame from a transmitter should use a sequence number greater than the sequence number of the previously transmitted frame.

A packet with a spoofed MAC that does not carefully select a sequence number to match that of the current counter can be identified as anomalous by a WLAN IDS. Vendors of WLAN IDS products use this technique to identify the presence of an attacker on the network, generating an alert indicating an out-of-sequence packet on the network. Although modern wireless cards make it trivial for an attacker to spoof both the sequence number and MAC address when transmitting a forged frame, few public attack tools bother to set the sequence number in an attempt to evade detection.

Since the WLAN IDS is transmitting a deauthenticate frame with a spoofed source MAC address of the protected access point, an attacker can use the same detection technique to identify the spoofed frames. In the analysis for this paper, none of the evaluated WLAN IDS products attempted to evade detection by selecting sequence numbers that match those of the legitimate source MAC address. The following trace is an example of a session containment from a popular WLAN IDS vendor, indicating the sequence number selection for each frame.

```
$ tethereal -r capture.dump -n -R "wlan.sa eq 00:12:17:9f:08:72" -V | egrep
"Type\/Subtype|Sequence"
   Type/Subtype: Beacon frame (8)
   Sequence number: 2837
   Type/Subtype: Deauthentication (12)
```

```
    Sequence number: 0
    Type/Subtype: Probe Response (5)
    Sequence number: 2839
    Type/Subtype: Probe Response (5)
    Sequence number: 2840
    Type/Subtype: Deauthentication (12)
    Sequence number: 0
```

In this example, the tethereal tool is used to read from the capture.dump file with a display filter that returns packets with the source MAC address of 00:12:17:9f:08:72. Filtering the output to display the packet type/subtype information and the sequence number, we see two patterns of sequence number selection. The first pattern starts with the expected behavior for sequence numbers, incremented by a positive integer for each packet sent in the pattern 2837, 2839, 2840 (frame 2838 may have been corrupted during transmission). The second pattern, however, uses a fixed sequence number counter of 0 for deauthenticate frames. From this output, we can deduce that the deauthenticate frames are not transmitted from the node at 00:12:17:9f:08:72, but are spoofed from another station within range of the receiver.

Not all vendors use the same fixed value for the spoofed deauthenticate frames, but all vendors have sufficient characteristics to differentiate legitimate and spoofed frames transmitted on the network.

Disconnect Notice Anomaly Analysis

Another technique for identifying an attempt to engage a client with session containment is to apply basic anomaly analysis techniques to the deauthenticate frames transmitted on the network.

An access point will legitimately transmit a deauthenticate frame to a selected station when it experiences a resource constraint and must reduce the number of connected clients, or to respond to another anomalous event on the network. The IEEE 802.11-1999 specification for the MAC layer identifies several different reason codes that are used as a fixed parameter in the payload of deauthenticate and disassociate frames to identify the reason a station was disconnected, as shown below (IEEE80211).

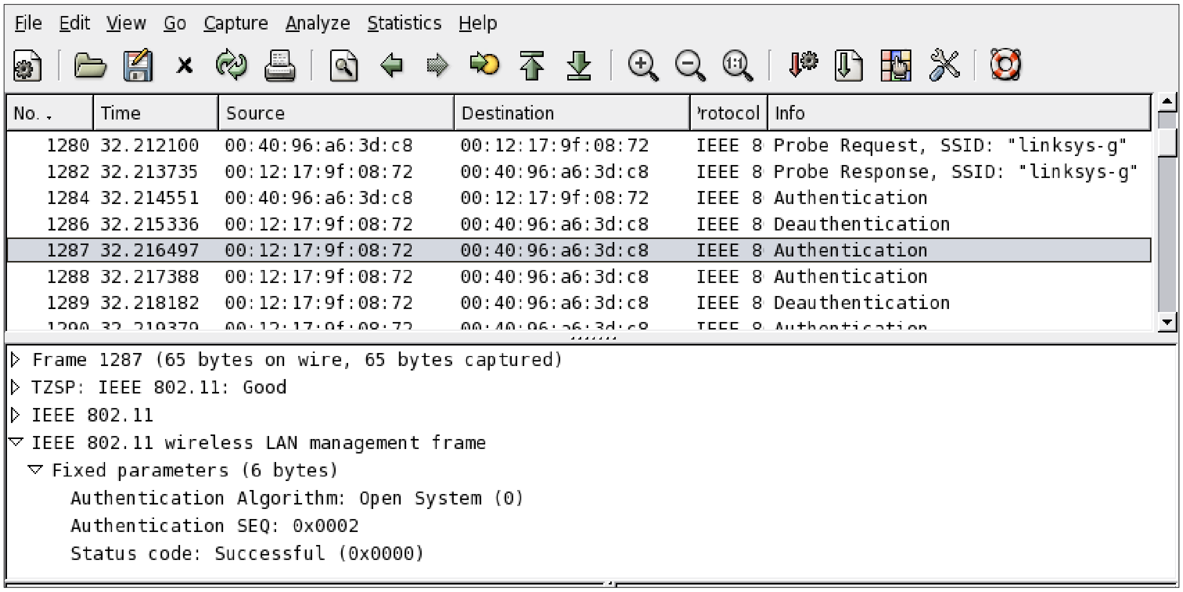
Reason Code	***Meaning***
0	Reserved
1	Unspecified reason
2	Previous authentication no longer valid
3	Deauthenticated because sending station is leaving (or has left) IBSS or ESS
4	Disassociated due to inactivity
5	Disassociated because access point is unable to handle all currently associated stations
6	Class 2 frame received from nonauthenticated station

Reason Code	***Meaning***
7	Class 3 frame received from nonassociated station
8	Disassociated because sending station is leaving (or has left) BSS
9	Station requesting (re)association is not authenticated with responding station
10-65535	Reserved

While a legitimate activity to notify a client of their deauthenticated or disassociated status from an access point, it is anomalous for an access point to transmit a flood of these messages to persistently disconnect a station from accessing the network. Identification of a flood of deauthenticate or disassociate notices over a fixed period of time is another technique that can be used by an attacker to identify the presence of a WLAN IDS system.

Due to the nature of the spoofed traffic, an attacker can use a simpler technique to identify the presence of spoofed deauthenticate frames from a WLAN IDS system. The IEEE 802.11-1999 specification indicates that deauthenticate and disassociate frames are not a request from the transmitter; rather they are a notification that the recipient has already been deauthenticated or disassociated from the network. This prompts the recipient to reset their connection state and reconnect to the network.

Since the deauthenticate and disassociate messages are spoofed however, the access point has not already terminated the connectivity for the station that is the target of session containment. This is evidenced by the transmission of frames from the access point to the unauthorized station immediately following the disconnect message. In the case of a legitimate deauthenticate or disassociate message, the recipient has already been disconnected from the network, and would not continue to receive traffic until it reconnected to the network. In the case of spoofed deauthenticate frames however, the access point does continues to treat the station as if they were still authenticated to the network, sending them data frames as if nothing has changed with the status of the client. This behavior is demonstrated in a packet trace from a station being contained by a WLAN IDS as shown below.



```
No. .  Time       Source             Destination        'rotocol Info
  1280 32.212100  00:40:96:a6:3d:c8  00:12:17:9f:08:72  IEEE 8  Probe Request, SSID: "linksys-g"
  1282 32.213735  00:12:17:9f:08:72  00:40:96:a6:3d:c8  IEEE 8  Probe Response, SSID: "linksys-g"
  1284 32.214551  00:40:96:a6:3d:c8  00:12:17:9f:08:72  IEEE 8  Authentication
  1286 32.215336  00:12:17:9f:08:72  00:40:96:a6:3d:c8  IEEE 8  Deauthentication
  1287 32.216497  00:12:17:9f:08:72  00:40:96:a6:3d:c8  IEEE 8  Authentication
  1288 32.217388  00:12:17:9f:08:72  00:40:96:a6:3d:c8  IEEE 8  Authentication
  1289 32.218182  00:12:17:9f:08:72  00:40:96:a6:3d:c8  IEEE 8  Deauthentication
  1290 32.219379  00:12:17:9f:08:72  00:40:96:a6:3d:c8  IEEE 8  Authentication
```

```
▷ Frame 1287 (65 bytes on wire, 65 bytes captured)
▷ TZSP: IEEE 802.11: Good
▷ IEEE 802.11
▽ IEEE 802.11 wireless LAN management frame
  ▽ Fixed parameters (6 bytes)
      Authentication Algorithm: Open System (0)
      Authentication SEQ: 0x0002
      Status code: Successful (0x0000)
```

This Ethereal screen capture displays a station at 00:40:96:a6:3d:c8 attempting to connect to an access point at 00:12:17:9f:08:72. The station exchanges a probe request and a probe response with the access point, and then attempts to authenticate. After transmitting the authenticate request message in frame 1284, the access point appears to have deauthenticated the station in frame 1286. Immediately following the deauthenticate message however, the access point transmits an authentication response frame (selected in the example above) indicated a successful status code. This is another indicator for the attacker that the deauthenticate frames are not legitimate, since no traffic from the access point should follow a deauthenticate notice until the station has reauthenticated to the network.

Signal Strength Analysis

Another technique that WLAN IDS providers use to identify spoofed traffic on the wireless network is to assess the observed signal strength for transmitters on the network.

WLAN IDS sensors are typically deployed at an oversubscribed model compared to a traditional WLAN access point deployment. This is in an effort to save on WLAN IDS deployment costs, since it is not necessary to have a WLAN IDS sensor for each access point on the network. It is not uncommon for organizations to deploy one WLAN IDS sensor for every 4 or 5 access points.

Because of the over-subscription deployment model, a WLAN IDS sensor will likely be at a different location than the access point it is protecting with session containment. This will cause variations in the signal levels for multiple transmitters appearing to originate from the same source MAC address. Since the signal level for a stationary transmitter (such as an access point) should remain relatively consistent for each transmitted frame, the receipt of a deauthenticate frame with the same source MAC address and a significantly different signal level would indicate the presence of a second transmitter. A signal level that is consistently higher or lower than that of the access point would indicate that the transmitter is closer or farther than the legitimate source, respectively, although this is subject to the relative transmit strength and antenna selection for both the access point and the WLAN IDS sensor.

Upon receipt of a frame, the wireless card will report signal level information which is recorded in the DLT_IEEE802_11_RADIO_AVS libpcap frame encapsulation type on Linux systems. The following trace is an example of the relative signal levels for the same source MAC address during session containment. Note that the signal level precedes the frame type in this example; additional spacing has been added for clarity.

```
$ tethereal -r capture2.dump -n -R "wlan.sa eq 00:0f:66:e3:76:3b" -V | egrep
"Signal|Type\/Subtype"

   Signal: 0xbc (DID 0x6044,  Status 0x0, Length 0x4)
   Type/Subtype: Beacon frame (8)

   Signal: 0xbc (DID 0x6044,  Status 0x0, Length 0x4)
   Type/Subtype: Beacon frame (8)

   Signal: 0xe3 (DID 0x6044,  Status 0x0, Length 0x4)
   Type/Subtype: Dissassociate (10)

   Signal: 0xbb (DID 0x6044,  Status 0x0, Length 0x4)
   Type/Subtype: Association Response (1)

   Signal: 0xe2 (DID 0x6044,  Status 0x0, Length 0x4)
   Type/Subtype: Dissassociate (10)

   Signal: 0xba (DID 0x6044,  Status 0x0, Length 0x4)
   Type/Subtype: Association Response (1)
```

In this trace we extract the signal level and frame type/subtype from the stored capture file capture2.dump, applying a filter to return only packets from the source MAC address 00:0f:66:e3:76:3b. The first two frames are type beacon with a consistent signal level of 0xbc. The next frame is type disassociate with a signal level of 0xe3. By itself, this could be the result of a variation in RF characteristics, and not sufficient to identify the presence of spoofed frames.

The fourth frame is type association response, with a signal level of 0xbb. This signal level is consistent with the first two beacon frames. The fifth frame is another disassociate notice with a signal level of 0xe2, followed by an association response frame with a signal level that returns to 0xba.

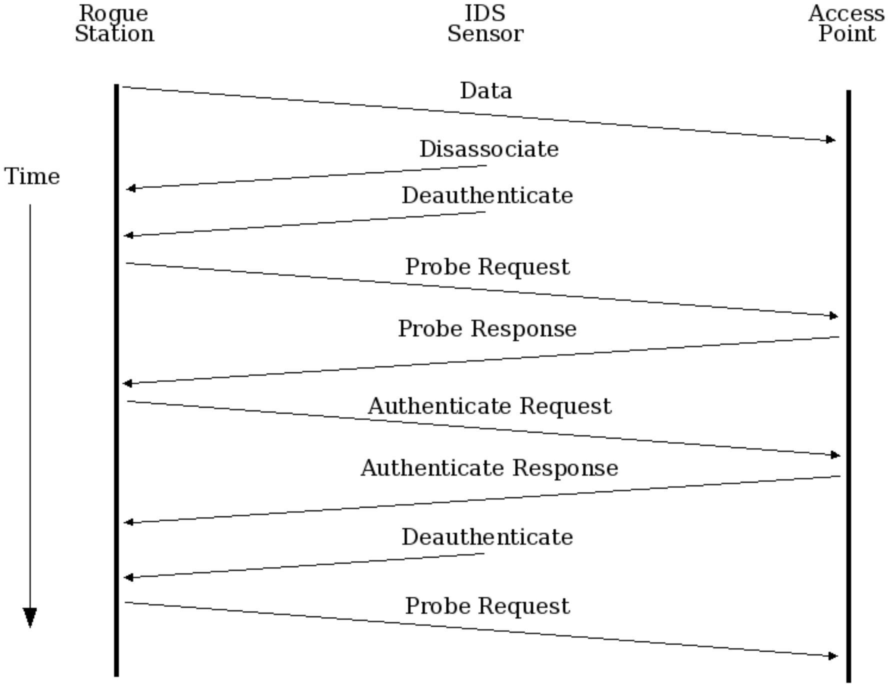
From this output, an attacker can deduce that the disassociate frames are not from the legitimate transmitter, but are from a spoofed source MAC address that is likely closer to the attacker than the protected access point.

Disconnect Request Timing

The technique used to time the transmission of deauthenticate or disassociate notices against a rogue client is another interesting implementation option selected by WLAN IDS vendors. Recall that the IDS must stop a rogue device from accessing the network while minimizing the impact to the medium. Failure to minimize impact to the medium during session containment could adversely affect other nearby networks on the same channel.

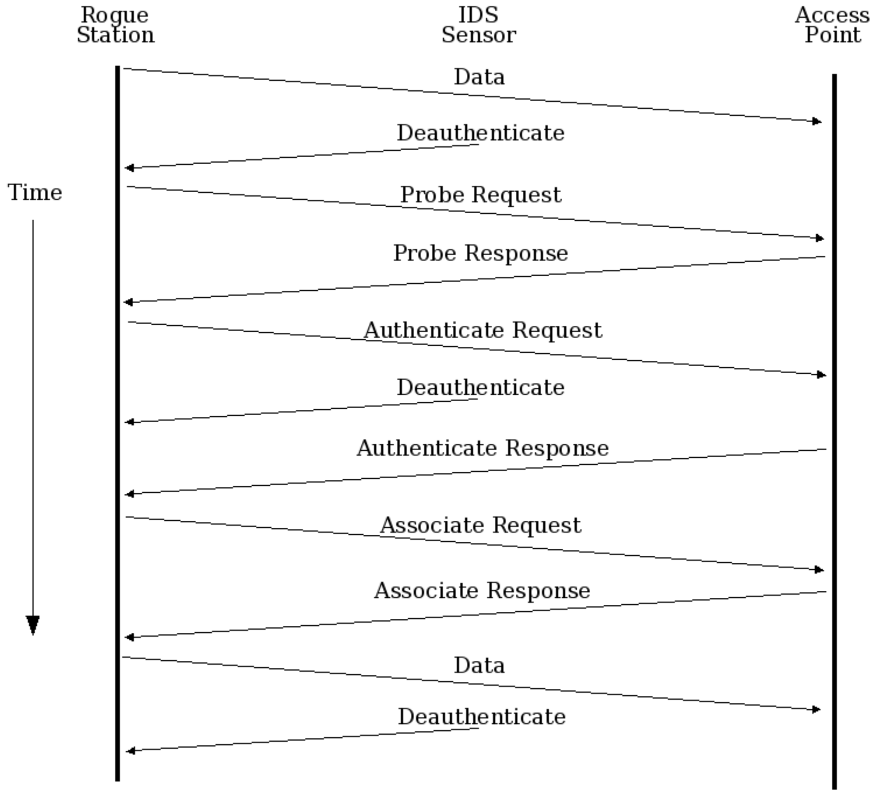
The most effective technique to time the delivery of deauthenticate and disassociate notices while minimizing impact to the medium is to monitor for the completion of the IEEE 802.11 authentication processes, spoofing a deauthenticate messageimmediately following the completion of the authentication exchange. Similarly, a disassociate frame can be transmitted immediately following the association process to prevent a rogue station from transmitting on the network. This technique would minimize the number of spoofedframes that needto be transmitted to contain a rogue station, while effectively stopping access to the network.

The following diagram illustrates this disconnect request timing technique.



This illustration was modeled after an observed session containment against a rogue station. In this case, the WLAN IDS intelligently transmits a disassociate frame followed by a deauthenticate frame, causing the rogue station to disconnect from the network. When the rogue station tries to connect to the accesspoint again, the WLAN IDS waits until the authentication process is complete, and then repeats the deauthenticate message,forcing the station to repeat the connection process.

Surprisingly however, few of the vendors evaluated in the research for this paper implement such a technique. Rather, most vendors implement the less-sophisticated technique of repeating the transmission of a deauthenticate notice at a fixed interval. This has the disadvantage of utilizing the spectrum more than absolutely necessary to contain a rogue station, and makes it possible for an attacker to transmit one or more frames between deauthenticate frames when a sufficiently long delay between deauthenticate notices is used. The following diagram illustrates this fixed-delay disconnect timing technique.



This illustration was also modeled after an observed session containment against a rogue station. In this case, the WLAN IDS initiates a deauthenticate DoS against a rogue station by transmitting the deauthenticate notice at a fixed interval. After observing the rogue station transmitting on the network, the IDS sensor transmits a spoofed deauthenticate frame, causing the station to disconnect from the network. After disconnecting, the station exchanges a probe request and a probe response with the access point, and then transmits an authentication request. After the authentication request, the IDS sensor transmits another deauthenticate message, but this is silently ignored by the client because it has not completed the authentication exchange with the access point. Immediately following the spoofed deauthenticate frame, the access point returns an authentication success message. After successfully authenticating, the rogue station associates to the network and is able to transmit a single data frame into the protected network before the next deauthenticate frame is transmitted by the WLAN IDS sensor.

In order to limit the ability for a rogue station to transmit any data frames before being deauthenticated from the network, some WLAN IDS providers have opted to shorten the interval between deauthenticate frames. While this is an effective technique to disconnect a rogue station before completing the authentication and association process, it requires more utilization of the wireless medium to be effective, an undesirable side-effect.

Evading Session Containment

Despite weaknesses with the timing of deauthenticate frames, some WLAN IDS session containment implementations suffer from an implementation weakness that allows an attacker to evade attempts at controlling access to the wireless network altogether.

Session containment is very similar to a feature in traditional wired IDS systems known as session sniping. One use of session sniping is to tear-down established TCP connections between an attacker and a monitored device by transmitting spoofed TCP RST packets. With this feature, the IDS can mitigate the effect of an attack by preventing any further data from being transmitted over the established connection.

A classic weakness in session sniping implementations is for the IDS to transmit the spoofed TCP RST packet only to the attacker. This thinking assumes the attacker will honor the TCP RST packet, and tear-down the connection. A simple modification to the Linux kernel can be applied to ignore the TCP RST packet, leaving the connection in an established state. The IDS reports that the connection was terminated since it issued the TCP RST packet, but the attacker is still able to leverage the established connection.

A similar weakness exists in some WLAN IDS session containment implementations. In these cases, the WLAN IDS transmits spoofed deauthenticate and/or disassociate frames to the rogue station, assuming the attacker believes the frame was transmitted by the access point as a notification of being deauthenticated or disassociated from the network. However, since the attacker can readily identify when spoofed deauthenticate frames are being transmitted by the WLAN IDS; he can opt to simply ignore the frames by modifying the wireless card drivers.

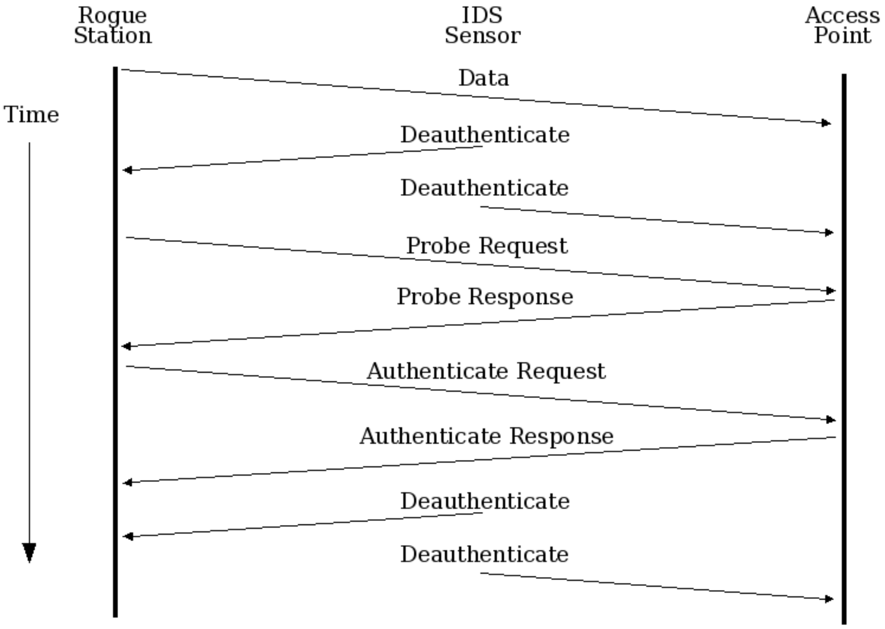
To test this theory, a Linux host with a Proxim 8480-WD a/b/g card (Atheros 5212 chipset) was configured with a modified version of the MADWIFI drivers from http://madwifi.sourceforge.net/. The driver was modified to ignore deauthenticate and disassociate frames, accomplished by commenting out the ieee80211_new_state() call in net80211/ieee80211_input.c. A patch in unified diff format is supplied in Appendix A.

After connecting to a protected access point, the WLAN IDS was configured to contain the rogue session. The following trace was captured with the modified driver during session containment.

```
0.000000  00:20:a6:4f:01:40  -> 00:0f:66:e3:e4:01  ICMP Echo (ping) request
0.002061  00:0f:66:e3:76:39  -> 00:20:a6:4f:01:40  ICMP Echo (ping) reply
0.086957  00:0f:66:e3:76:3b  -> 00:20:a6:4f:01:40  IEEE 802.11  Deauthentication
0.087733  00:0f:66:e3:76:3b  -> 00:20:a6:4f:01:40  IEEE 802.11  Deauthentication
1.000843  00:20:a6:4f:01:40  -> 00:0f:66:e3:e4:01  ICMP Echo (ping) request
1.002904  00:0f:66:e3:76:39  -> 00:20:a6:4f:01:40  ICMP Echo (ping) reply
1.008565  00:0f:66:e3:76:3b  -> 00:20:a6:4f:01:40  IEEE 802.11  Deauthentication
1.039706  00:0f:66:e3:76:3b  -> 00:20:a6:4f:01:40  IEEE 802.11  Deauthentication
2.001665  00:20:a6:4f:01:40  -> 00:0f:66:e3:e4:01  ICMP Echo (ping) request
2.003707  00:0f:66:e3:76:39  -> 00:20:a6:4f:01:40  ICMP Echo (ping) reply
2.118839  00:0f:66:e3:76:3b  -> 00:20:a6:4f:01:40  IEEE 802.11  Deauthentication
2.119614  00:0f:66:e3:76:3b  -> 00:20:a6:4f:01:40  IEEE 802.11  Deauthentication
2.134993  00:0f:66:e3:76:3b  -> 00:20:a6:4f:01:40  IEEE 802.11  Deauthentication
2.221238  00:0f:66:e3:76:3b  -> 00:20:a6:4f:01:40  IEEE 802.11  Deauthentication
3.002545  00:20:a6:4f:01:40  -> 00:0f:66:e3:e4:01  ICMP Echo (ping) request
3.004559  00:0f:66:e3:76:39  -> 00:20:a6:4f:01:40  ICMP Echo (ping) reply
```

In this trace, a wireless station at 00:20:a6:4f:01:40 is exchanging ICMP echo requests and responses with another host while connected to an access point at 00:0f:66:e3:76:3b. Configured to ignore the deauthenticate notices using the patch supplied in Appendix A, the station is able to bypass the attempt to contain the session, and continues to exchange data with another system despite repeated deauthenticate notices.

Fortunately, many vendors recognize this implementation weakness in session containment, and transmit bidirectional deauthenticate notices, targeting the wireless station and the protected access point. This technique is illustrated in the following diagram.



In this example, the IDS spoofs the identity of both the access point and the rogue station when sending deauthenticate messages. Under most circumstances, a deauthenticate frame sent to the rogue station will cause the attacker to disconnect from the network. In the cases where the attacker ignores deauthenticate frames, the deauthenticate messagesent to the access point will prevent them from accessing the network until they reauthenticate and reassociate.

WLAN IDS Fingerprinting

The ability to fingerprint the selection of wireless IDS products in an organization is an important step for a sophisticated attacker wishing to avoid detection or masquerade their attack or intent. A significant weakness in the implementation of session containment is that an otherwise passive WLAN IDS system becomes an active transmitter on the network. As such, an attacker that can coax a WLAN IDS system into engaging a station with session containment can use the characteristics of the traffic to identify the vendor that is being used for wireless network monitoring.

Once an attacker can identify the vendor equipment being used to monitor the network, they can plan their attack according to the strengths and weaknesses of the vendor's IDS implementation. While the signatures used by WLAN IDS vendors to identify attacks are seldom publicly available and likely considered trade secrets, an attacker can use knowledge of the system to carefully plan their attack to evade detection.

Several characteristics can be used to identify the vendor implementing session containment:

- Disconnect technique. The frame type used to disconnect the station from the network can be evaluated for a fingerprinting factor. Of the vendors evaluated for this paper, deauthenticate frames are common, with one vendor also implementing disassociate frames.

- Direction. Some vendors may send deauthenticate messages to both the access point and the station. Others may transmit deauthenticate messages only in a single direction.
- Sequence number selection. The selection of sequence numbers in frames transmitted by the WLAN IDS provides additional clues useful for fingerprinting.
- Reason code. When transmitting a deauthenticate or disassociate frame, the transmitter must include a reason code. The reason code is selected by the vendor, and may differ between implementations.
- Fragment number. The fragment number is associated with the sequence number, used to uniquely identify each fragmented portion of a single packet. This field should only be set if the fragment flag in the frame control header is set, otherwise it is ignored.
- Timing. The transmission frequency of deauthenticate frames can also provide hints to fingerprint the WLAN IDS vendor when the vendor applies a fixed duration between deauthenticate frames.

For the purposes of research, four vendor's WLAN IDS products with session containment capabilities were selected for analysis. The fingerprint statistics for each of these vendors is listed below.

Network Chemistry

Product	RF Protect Server version 4.0.5, sensor version 5.1.70-01-03
Disconnect technique	Deauthenticate only
Disconnect direction	Bidirectional
Reason code, AP to STA	Reason code # 2
Reason code, STA to AP	Reason code # 3
Timing between deauthenticate messages	Fixed duration between deauthenticate frames, every .10 seconds
Sequence number selection	Sequential counter, modulo 4096 starting at 0 with first deauthenticate frame
Fragment number selection	Sequential counter, modulo 15 starting at 0. Fragment bit in the frame control header is not set.
Notes	The use of the fragment number field in deauthenticate messages is unique to Network Chemistry's product. This may be an inter-process communication mechanism used by Network Chemistry to associate the transmission and delivery of deauthenticate frames.

AirTight

Product	SpectraGuard Server 3.0.08, sensor version 3.0.8

Disconnect technique	Deauthenticate only
Disconnect direction	Bidirectional
Reason code, AP to STA	Reason code# 2
Reason code, STA to AP	Reason code# 3
Timing between deauthenticate messages	Fixed duration between deauthenticate frames, every .15 seconds
Sequencenumber selection	Fixed sequence counter. Deauthenticate frames from the STA to the AP uses a sequence number of 0. Deauthenticate frames from the AP to the STA uses a sequence number of 1. Occasionally, sequence numbers 2 and 3 were observed for traffic from STA to AP and AP to STA, respectively; reason unknown.
Fragment number selection	Always 0
Notes	The combination of reason code, sequence number selection and timing between deauthenticate messagesuniquely identifies AirTight's SpectraGuard product.

AirDefense

Product	AirDefense Enterprise 6.0, sensor model M400, version 4.1.4.7
Disconnect technique	Combination deauthenticate and disassociate
Disconnect direction	Unidirectional, AP to STA only
Reason code, AP to STA	Reason code# 2
Reason code, STA to AP	n/a
Timing between deauthenticate messages	Intelligent timing of deauthenticate and disassociate frames. Timing depends on the characteristics of the client being contained.
Sequencenumber selection	Sequential counter, modulo 4096 starting at 0 with first deauthenticate or disassociate frame
Fragment number selection	Always 0
Notes	The unidirectional characteristics of the AirDefense Enterprise 6.0 product are unique among those vendors tested. Intelligent timing as well as combined use of deauthenticate and disassociate frames uniquely identified the AirDefense session containment implementation.

AirMagnet

Product	AirMagnet Enterprise server version 5.2.0, sensor 5.2.0-2927
Disconnect technique	Deauthenticate only
Disconnect direction	Bidirectional
Reason code, AP to STA	Reason code #5
Reason code, STA to AP	Reason code #3
Timing between deauthenticate messages	When sending deauthenticate frames from STA to AP, a single deauthenticate frame is sent after .03 second delay, followed by five deauthenticate frames .02 seconds later. When sending deauthenticate frames from AP to STA, five frames are sent in rapid succession every .05 seconds.
Sequence number selection	Always 0
Fragment number selection	Always 0
Notes	The use of a fixed sequence number counter and a unique deauthenticate reason code for AP to STA traffic uniquely characterizes the AirMagnet Enterprise product.

Additional vendor products were not available for assessment. The author welcomes contributions of packet captures that include session containment examples along with the product name and version information.

Recommendations

Mitigating the risks associated with WLAN session containment implementations can be separated into vendor-specific and consumer recommendations.

Vendor Recommendations

Vendors who have implemented or are planning to implement session containment techniques should use intelligent timing techniques to disconnect an unauthorized station in favor of fixed-delay deauthenticate transmission. By closely monitoring the activities of an attacker, vendors can implement a two-stage disconnect process, transmitting a deauthenticate frame after the station completes the authentication exchange, and a disconnect frame after the association exchange if necessary. In this fashion, the IDS can more efficiently control the attacker's access to the network, while carefully monitoring their activity. This will also allow the IDS to raise a high-priority alert when it detects the presence of a station that is intentionally ignoring spoofed deauthenticate frames.

In order to mitigate session containment evasion, vendors should transmit bidirectional disconnect frames (deauthenticate and disassociate) to the access point and the unauthorized station. While it may seem attractive to spoof deauthenticate frames only to the access point, this technique may not be effective against all types of access points, especially those with built-in DoS defense mechanisms. Vendors may consider using a third disconnect technique if both the AP and the unauthorized station do not respond to the deauthenticate frames, such as an RF-jamming attack. Alternatively, integration with the wired-infrastructure would allow the vendor to more effectively disable the AP wired network port, at the cost of additional implementation complexity.

Consumer Recommendations

The documentation accompanying several vendors products indicate that session containment should not be solely used to mitigate the threat of rogue APs. Organizations should heed this advice, applying incident response measures to quickly locate and remove rogue access points when they are identified.

Organizations can test their vendor's session containment implementations to identify if they are vulnerable to the evasion attacks described in this paper by watching for deauthenticate and/or disassociate frames transmitted to the unauthorized station and the access point. If the WLAN IDS does not implement bidirectional session containment, consumers can leverage their purchasing and service renewal influence with vendors to influence product changes that can be applied with future software updates.

When implementing WLAN IDS systems, organizations should consider the weaknesses described in this paper before configuring session containment features. Some organizations may not be especially concerned about the risks of IDS fingerprinting, especially if they have implemented a robust network security mechanism based on a strong 802.1x/EAP type and WPA or WPA2. Other organizations may want to retain the silence of WLAN IDS systems to mitigate fingerprinting techniques that could be leveraged to select attacks that are not detected by the IDS.

Conclusion

When implemented carefully, session containment can be a valuable mechanism to augment a secure wireless network deployment. The use of session containment does not come without risks however, including WLAN IDS fingerprinting and possible evasion. Only organizations that are well-informed about the risks associated with session containment can make an educated decision about deploying this technology on their network.

The author would like to thank Frank Bulk and Mike Kershaw for their assistance in the research and review of this paper.

Works Cited

AirDefense. "Wireless Security: Are Encryption, Authentication & VPNs Enough?" URL: http://www.airdefense.net/webcasts/gartner/May05/ (17 May 2005).

IEEE80211. ANSI/IEEE Std 802.11, 1999 Edition (R2003), "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", 12 June 2003.

Wright, Joshua. "Detecting Wireless LAN MAC Address Spoofing", URL: http://802.11ninja.net/papers/wlan-mac-spoof.pdf (17 May 2005).

Appendix A

The changes presented below were applied to the MADWIFI driver source for Linux systems from the main CVS branch as of 05/17/2005. An electronic copy of this patch is available at http://802.11ninja.net/code/madwifi-ignore-discon.diff.

```
diff -ur madwifi-orig/net80211/ieee80211_input.c madwifi/net80211/ieee80211_input.c
--- madwifi-orig/net80211/ieee80211_input.c  2005-05-17  08:52:44.000000000  -
0400
+++ madwifi/net80211/ieee80211_input.c       2005-05-17  09:03:02.000000000  -
0400
@@ -2230,8 +2230,9 @@
                ic->ic_stats.is_rx_deauth++;
                switch (ic->ic_opmode) {
                case IEEE80211_M_STA:
-                       ieee80211_new_state(ic, IEEE80211_S_AUTH,
-                           wh->i_fc[0] & IEEE80211_FC0_SUBTYPE_MASK);
+                       /* ieee80211_new_state(ic, IEEE80211_S_AUTH,
+                           wh->i_fc[0] & IEEE80211_FC0_SUBTYPE_MASK); */
+         printk("MADWIFI: Ignoring deauthenticate notice.");
                        break;
                case IEEE80211_M_HOSTAP:
                        if (ni != ic->ic_bss) {
@@ -2266,8 +2267,9 @@
                ic->ic_stats.is_rx_disassoc++;
                switch (ic->ic_opmode) {
                case IEEE80211_M_STA:
-                       ieee80211_new_state(ic, IEEE80211_S_ASSOC,
-                           wh->i_fc[0] & IEEE80211_FC0_SUBTYPE_MASK);
+                       /* ieee80211_new_state(ic, IEEE80211_S_ASSOC,
+                           wh->i_fc[0] & IEEE80211_FC0_SUBTYPE_MASK); */
+         printk("MADWIFI: Ignoring disassociate notice.");
                        break;
                case IEEE80211_M_HOSTAP:
                        if (ni != ic->ic_bss) {
```