The Insecurity of Wireless Networks

Article *in* IEEE Security and Privacy Magazine · July 2012

DOI: 10.1109/MSP.2012.60

4 authors, including:

F.T. Sheldon
University of Idaho
237 PUBLICATIONS **2,668** CITATIONS

SEE PROFILE

Seong-Moo Yoo
University of Alabama in Huntsville
120 PUBLICATIONS **1,685** CITATIONS

SEE PROFILE

The Insecurity of Wireless Networks

Frederick T. Sheldon | Oak Ridge National Laboratory
John Mark Weber, Seong-Moo Yoo, and W. David Pan | University of Alabama in Huntsville

Wireless is a powerful core technology enabling our global digital infrastructure. Wi-Fi networks are susceptible to attacks on Wired Equivalency Privacy, Wi-Fi Protected Access (WPA), and WPA2. These attack signatures can be profiled into a system that defends against such attacks on the basis of their inherent characteristics.

Wireless insecurity has been a critical issue since the first implementation of the IEEE 802.11 standard (used by Wi-Fi) in 1997.[1–3] When it became obvious that Wired Equivalency Privacy (WEP), an IEEE standard security algorithm for wireless networks, was compromised,[4–7] the Wi-Fi Alliance (www.wi-fi.org) developed IEEE 802.11i, called Wi-Fi Protected Access (WPA), to address the significant security flaws. WPA employs the Temporal Key Integrity Protocol (TKIP), which accounts for limitations in hardware designed for WEP encryption and reuses WEP algorithms.[8]

However, TKIP's weaknesses necessitated developing countermeasures to protect WPA from attack via TKIP. In response, the Wi-Fi Alliance developed WPA2 (802.11i-2004). Instead of TKIP, it employs the Counter Mode Cipher Block Chaining Message Authentication Code Protocol (CCMP), which is based on the Advanced Encryption Standard (AES). This encryption, which doesn't reuse WEP, is securer but has greater hardware requirements and a higher cost.[9]

Nevertheless, attacks still represent a threat to networks employing WPA and WPA2. Here, we examine the strengths and weaknesses of some of those attacks and suggest ways to detect and prevent them. Ultimately, we hope to provide a clearer picture of why and how wireless protocols and encryption must achieve a more scientific basis for detecting and preventing such attacks.

More on WPA and WPA2

WPA is intended for use with an 802.1x authentication server called WPA Enterprise. In the absence of an authentication server, WPA uses a preshared key (PSK) mode, which is less secure but more convenient. So, this mode, which is called WPA-PSK or WPA Personal, is restricted to small-office, home-office use.

WPA encryption uses the same RC4 stream cipher as WEP, but improved with a 48-bit TKIP sequence counter (TSC) instead of WEP's 24-bit key. Another improvement is that TKIP allows dynamic key sharing and key mixing with a 128-bit key, making it much harder to attack. WPA's cyclic redundancy check (CRC) is updated to make altering a packet and its CRC portion much more difficult. Furthermore, the 64-bit message integrity check (MIC) algorithm named Michael[6] counts frames, making replay attacks impossible.[10]

 Copublished by the IEEE Computer and Reliability Societies

In WPA2, the more secure AES block cipher replaces the RC4 stream cipher. Unfortunately, this necessitates replacing older hardware, owing to AES encryption's extensive processing demands.

The Evolution of WEP and WPA Attacks

One of the most basic WEP attacks is the *chopchop attack*.[10] Both WPA and WPA2 are known to be vulnerable to *brute-force attacks* (BFAs). Martin Beck and Erik Tews used the chopchop attack as the basis for a new WPA attack, now called the *Beck-Tews attack*.[10] Finn Halvorsen and Olav Haugen[9] as well as Toshihiro Ohigashi and Masakatu Morii[11] extended this type of attack. Hole 196 is an insider attack on WPA that exploits an inherent weakness of the "one-way" Group Temporal Key (GTK).[12]

The Chopchop Attack

The chopchop attack (www.aircrack-ng.org/doku.php?id=korek_chopchop) exploits WEP encryption by determining the PSK through trial and error, rather than mathematically or cryptographically. Figure 1 diagrams the attack.

The chopchop attack uses the access point (AP) to decipher wireless Address Resolution Protocol (ARP) packets. The attack chops off the packet's last byte and assumes that the encrypted byte is 0. The attack corrects the packet on the basis of a guess of 0, reencrypts it, and sends it to the AP. If the guess is correct, the AP retransmits the packet because the attack is using a multicast packet. In this case, the attacker knows the guess was correct. If the AP drops the packet, the attacker guesses 1 and restarts the process.

Consequently, the attacker can capture a WEP encrypted frame and replay it multiple times to decipher the payload one byte at a time. Attackers can decode small ARP frames in 10 to 20 seconds without breaking the WEP key.

The Brute-Force Attack

The BFA uses a library of possible PSKs to find a match for a captured handshake. If a network uses either WPA with a weak PSK or the WPA2 PSK mode, brute-force key finders such as CoWPAtty or the Aircrack-ng suite of tools (www.aircrack-ng.org) can recover the PSK. (Both CoWPAtty and Aircrack-ng are available in Backtrack Linux.) However, this attack doesn't work with 802.1x, which uses the Extensible Authentication Protocol (EAP) framework instead of the PSK.

The BFA captures a four-way handshake to obtain the temporal key used to encrypt all wireless traffic for that session. The shortcut of using a single *master key* instead of *per-user keys* eases deployment of WPA/WPA2 protected networks at the cost of making them vulnerable to BFAs during the key negotiation phase.

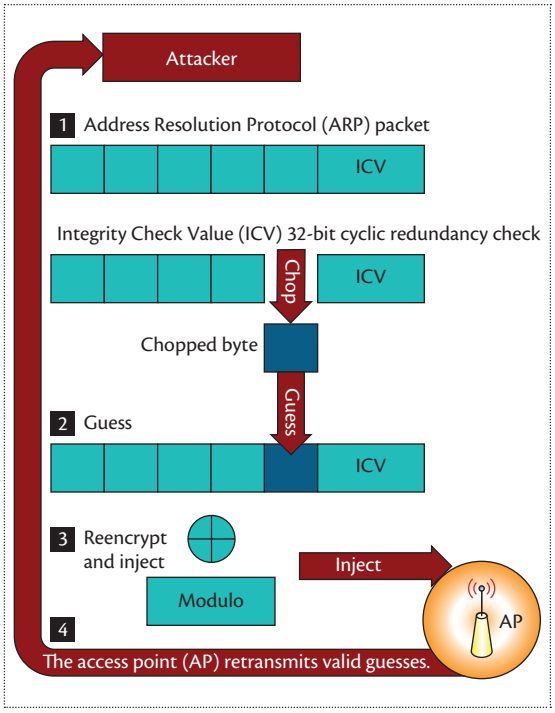

Figure 1. The chopchop attack on Wired Equivalency Privacy (WEP).[10] This attack determines the preshared key (PSK) through trial and error, rather than mathematically or cryptographically.

The attacker can simply wait for a handshake to occur or actively force one using a deauthentication attack on the target station. After capturing the handshake, the attacker can easily use the PSK library together with Aircrack-ng to continue the attack.[13]

The Original Beck-Tews Attack

Beck and Tews showed how to use the chopchop attack's methodology to decrypt packets and then use the recovered streams to air a number of custom packets. So, when an MIC failure occurs, the attacker can observe the response and waits for 60 seconds to avoid MIC countermeasures. More than two MIC failures in 60 seconds cause the AP and station to shut down for 60 seconds. Attackers using this mechanism can decode a packet at a rate of one byte per minute. So, they typically can decode small ARP frames in approximately 15 minutes. Figure 2 shows the attack flowchart.

Assuming a station is in range or the AP is near (or the attacker is using a directional antenna), the attack's success depends on the following prerequisites. The targeted AP must use IPv4 while the victim station runs WPA with a PSK. The TKIP rekey interval must be relatively long (the typical default is one hour), and the AP must have 802.11e quality of service (QoS) activated.

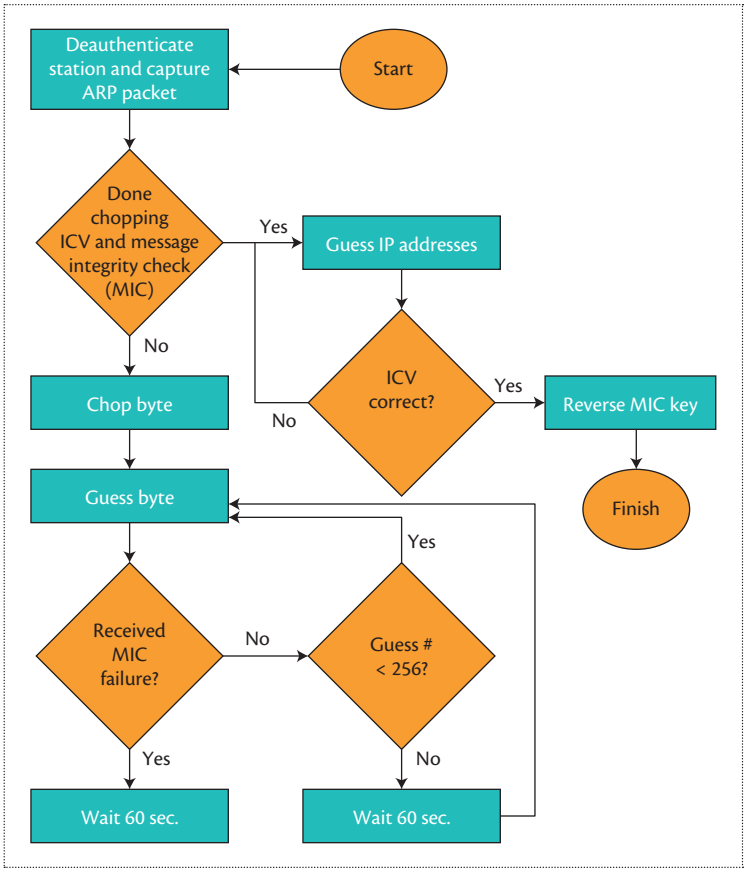

Figure 2. The original Beck-Tews attack on Wi-Fi Protected Access (WPA).[10] This attack requires the access point (AP) to have 802.11e quality of service (QoS) activated.

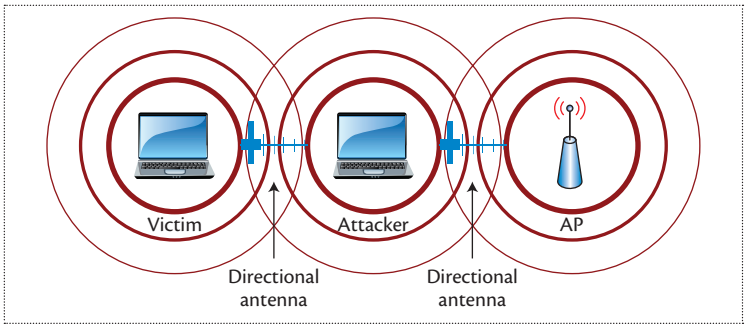

Figure 3. The Ohigashi-Morii man-in-the-middle attack.[11] This attack doesn't need 802.11e QoS activated.

In our experiments, we discovered one further prerequisite: the victim station must be using a Linux or Unix wireless stack.

The Ohigashi-Morii Attack

Ohigashi and Morii developed a man-in-the-middle (MITM) attack that doesn't need IEEE 802.11e QoS activated, thereby reducing the overall attack time. Figure 3 diagrams the basic attack. This attack prevents chopchop packets from reaching the client station and disconnecting the victim station. It uses directional antennas and selective packet replay. It can relay or withhold packets from the station or the AP as needed. (In this case, the TSC isn't incremented by the received packets.)

But in a real-world scenario, finding a disconnected client and getting between it and the AP is nearly impossible.[14] The attacker must be in range of the station and the AP, and the station must be outside the AP's range. This severely limits ease of use, raising the cost and exposure to the attacker. However, "difficult but not impossible" is just the type of opportunity well-funded advanced persistent threats look for (for example, selectively or stealthily jamming a remote or unattended critical cyberphysical station).

Halvorsen-Haugen Attacks

Halvorsen and Haugen extended the Beck-Tews attack in three ways. The first extension performs a modified chopchop attack on Dynamic Host Configuration Protocol (DHCP) ACK packets.

The second modifies the attack to become a cryptosystem denial-of-service attack using the MIC countermeasures. As we mentioned before, if two MIC failures are detected in one minute, the AP will shut the wireless interface down completely for one minute and force a rekey when the AP reinitializes. Halvorsen and Haugen proposed using this mechanism as an attack vector and using the chopchop attack to trigger an AP shutdown.

The third poisons the ARP by reinjecting spoofed ARP packets.

The Hole 196 Attack

The Hole 196 attack's creator found a vulnerability in IEEE 802.11-2007,[2,3] where the standard explains how the GTK is a one-way key that protects multicast and broadcast traffic. All the wireless supplicants that have joined the network know and store this key. The standard mentions that the GTK doesn't provide pairwise key support to protect from spoofing or data forgery. Accordingly, a trusted insider can spoof multicast messages protected by the one-way GTK.

The Hole 196 attack can enable wireless ARP spoofing. Such spoofing monitors all traffic and executes an MITM attack that uses the AP to decode encrypted traffic. So, stations acting on behalf of a malicious insider can forge an ARP request packet that announces that their machine is the gateway. Some call this a "pick-pocket attack" (one person distracts the victim while a third party easily empties the victim's pockets).

This attack poisons the victim station's ARP cache

Table 1. Results for the Beck-Tews attack on access points.

Manufacturer	Model	Firmware version	Attack successful?*	Comments
Linksys	WRT-54GL	4.30.14	Yes	—
Linksys	WRT-54G-TM	5.00.33	Yes	—
Hostap	Hostap	0.6.10	Yes	Testing confirmed the stability problems reported by Finn Halvorsen and Olav Haugen.[9]
D-link	DIR-655	1.21	No	The attack works only on Linux-based stations, according to Halvorsen and Haugen.
D-link	DGL-4300	1.9	No	—
Apple	Airport Extreme FB763LL/A	7.4.2	No	The firmware rekeyed after one message integrity check (MIC) failure message instead of two, effectively causing a denial-of-service failure (but no Address Resolution Protocol data exfiltration).

** Unsuccessful attacks might have been successful if the hostap application didn't crash before the attack was complete (which confirms results reported elsewhere).*

for the gateway. The victim station then sends packets (encrypted with the current Pairwise Transient Key [PTK]), but the destination is now the attacker station. The AP then receives the encrypted traffic from the victim station and sees that the destination is the attacker station. The AP decrypts the traffic and reencrypts it with the attacker's PTK. The attacker can now decrypt all the victim's traffic that passes through the AP. The attacker is now the MITM and can forward, store, or stop all the victim station's traffic.

WPA Attack Experiments

To better understand BFA and Beck-Tews-type attacks, we created a simple testbed. (We didn't perform experiments with the Hole 196 attack because its source code wasn't available.) We used a standard Linksys AP with stock firmware. We set up the AP in the same manner as in the original Beck-Tews attack we described earlier. The attacking station used a virtual machine (running Backtrack 4 Final) running on OS X 10.6.4. Backtrack was in turn connected to a USB-based Broadcom wireless card with the Ralink RT73 chipset.

BFAs

We used the airodump-ng part of Aircrack-ng to capture four-way handshake packets handed over to Aircrack-ng (tailored to attack the WPA PSK). Aircrack-ng also includes airmon-ng software-monitoring capabilities used to put wireless cards into promiscuous mode (to capture all packets). The Ralink chipset's driver is built into the Backtrack distribution and allows for packet injection. Therefore, all that we needed in preparation for our successful BFA was a typical password list (dictionary file) and that the AP and victim stations were linked. When the dictionary file contained the actual PSK, the key was extracted and therefore compromised.

Beck-Tews-Type Attacks on WPA

We recreated this attack to verify that the original Beck-Tews attack code is present and working in Aircrack-ng. We also aimed to ascertain the applicability of such an attack involving an attacking station, target station, and AP.

Hardware. As we mentioned before, for the attack to be successful, the AP must run an embedded Linux kernel, such as the popular Linksys WRT-54GL hardware. Table 1 summarizes the results for the APs we tested with this attack.

Figure 4 shows a target station's typical configuration. Initially, we set up separate machines for each interface. However, for ease of use and manageability, we switched to a single laptop that simultaneously acted as both the victim and attacker and that incorporated four wireless interfaces. We attached each wireless interface to separate virtual machines running on the laptop. We used a MacBook Pro 5,5 triple-booted to run Mac OSX 10.6.4, Windows XP SP2, and Backtrack 4 Final, each attached to the USB Broadcom wireless cards we described earlier.

Software. We used Kismet, a Linux-based Wi-Fi surveillance tool,[15] to instrument the attack. The data collected with Kismet let us quickly determine whether the attack was successful. We needed the Kismet data to configure

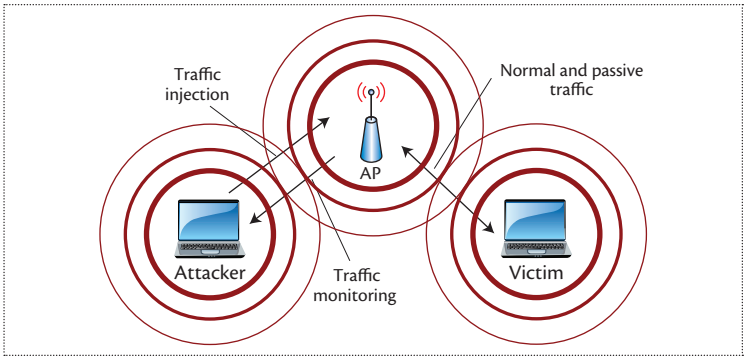

Figure 4. Our experimental setup for testing the original Beck-Tews attack on WPA is a testament to how feasibly attackers can create different attack scenarios.

the command line inputs for the attack. The attacker used Backtrack 4 Final, based on Ubuntu 8.10 LTS (long-term support).

Aircrack-ng provides a framework for wireless exploits similarly to how Metasploit provides for exploits of wired systems. We used its airmon-ng, airodump-ng, and tkiptun-ng components. Wireshark is a protocol analyzer that lets users watch packet data from a chosen network device.[13] The WPA supplicant was the IEEE 802.11i daemon software supplicant for Linux, FreeBSD, NetBSD, and Microsoft Windows. This daemon connects a station to the AP for normal operation.

Victim station setup. To attack a victim, we used Kismet and airodump-ng to ascertain the current in-use AP channel and determine whether TKIP was in effect. Knowing these facts, all we needed was the victim stations' MAC (media access control) address.

Table 2 lists the results for various OSs and chipsets; Figure 4 shows the setup. Generally, Linux and Unix supplicants were exploitable; Windows-based supplicants weren't. The tkiptun-ng attack didn't work on Android and Apple's iOS; a change (as yet unidentified) in the supplicant prevented it from working. The most common symptom of that attack failing was nondetection of the MIC failure message.

Halvorsen and Haugen's extensions. We built each of Halvorsen and Haugen's three extensions and validated their effectiveness. These extensions worked correctly.

Attacking Windows. We connected a Windows XP Service Pack 2 station to the same vulnerable Linksys AP used in some of the other attacks. We again used the Backtrack-4-based attacker to attack the victim station. Our attacking station also captured traffic to compare successful and unsuccessful attack traffic to understand the differences that cause an attack to fail. The victim used a Ralink chipset and the Ralink driver Win7:3.1.3.0 for the Edimax EW-7318USg USB wireless adapter. The tkiptun-ng attack successfully deauthenticated the victim but failed when it couldn't find an ARP packet from the victim to the AP.

Using a different chipset (Broadcom 802.11n), we continued the attack on Windows XP SP2. The tkiptun-ng attack found the ARP packets with the correct packet length. The attack proceeded to the next step, which launched the modified chopchop attack. However, this step failed because it didn't find the MIC failure frames that indicate a successful chopchop attack guess.

The next experiment attempted to shut down the compromised AP with MIC countermeasures. However, even after we ran a continuous modified chopchop attack for more than an hour, the AP didn't shut down. Our diagnosis is that during the attack, MIC failure frames from victim stations weren't broadcast.

Detecting and Preventing Attacks

As of this writing, enabling WPA2 encryption is the best way to secure a wireless network.[9] Furthermore, if you're using WPA2-PSK, then to avoid BFAs, you should use a very long shared "random" key. Naturally, longer keys are harder to crack using brute force ($2^{64} < 2^{128} << 2^{256}$ possible keys).

Airdrop-ng (part of Aircrack-ng) is a rule-based deauthentication tool. The versatility of the rule writing enables airdrop-ng to be a wireless intrusion prevention system (WIPS). Using it, defenders can actively monitor the wireless spectrum and respond to attacks on the host. They can deploy countermeasures against devices or rogue networks to prevent intrusions.

Defenders can also use airdrop-ng to single out stations or groups of stations to deauthenticate from connected APs, thereby forcing them to join honeynets. With this technique, defenders can perform reconnaissance. Of course, attackers can use the same technique to force stations to join evil-twin networks or rogue APs, to create MITM attacks.

In addition, defenders can combine Kismet's server-drone feature with Wireshark or TShark to create a distributed, low-cost wireless intrusion detection system (WIDS).[16] In this case, they configure airdrop-ng to become a WIPS to disconnect rogue stations and keep legitimate stations from connecting to nonapproved APs.

Next, we consider strategies for dealing with Beck-Tews-type and Hole 196 attacks.

Beck-Tews-Types Attacks

Let's consider the environment with respect to APs. Moving to WPA2 is the obvious and most straightforward protection. If you have to use WPA, issues of backward compatibility might exist in the smart grid or other

aging critical infrastructures. So, owing to hardware or device limitations, you should use the following tactics.

One possible protection is to turn off QoS. As we mentioned before, the original Beck-Tews attack needs QoS turned on so that it can increment the TKIP sequence counter (on each QoS channel) to bypass TKIP security. However, as we also mentioned, Ohigashi-Morii attacks don't require QoS to be activated and thus are immune to this defense.

Given that tkiptun-ng attacks depend largely on ARP packets, one way to defeat them is to simply use IPv6. IPv6 diminishes DHCP's role because it employs a stateless autoconfiguration, uses a network discovery protocol, and completely eliminates ARP packets. Unfortunately, the US has been slow in adopting IPv6, even though an urgent need exists for additional IP addresses. Major barriers include insufficient technical expertise, a lack of vendor support, and no clear picture of the commercial or economic benefits.

As our experiments showed, the tkiptun-ng attack failed with all the tested versions of Windows across a variety of common COTS wireless cards. Because most wireless stations are Windows based, this is one of the easiest existing protections. Unfortunately, it's unclear whether Microsoft deliberately chose this design feature, weighing its benefits against its impact on network performance. Knowing the answer would be useful.

For most Internet infrastructure stakeholders, wireless network security is in constant motion. The bottom line for them is to rely on best practices. But, the cost of constantly upgrading hardware, firmware, and software might be infeasible. Without practical scientifically valid cost risk analysis methods (and metrics) associated with sensible, enforceable notions of liability and cost benefit, best practices will continue to be patently insufficient. Will market-based, legal, or regulatory incentives improve our Internet infrastructure's trustworthiness? This question leads to even more uncertainty. On what basis and motives do we choose courses of action that have the highest risk-reduction return on investment for protecting against wireless networks' insecurity?

Table 2. Using the Beck-Tews-type attack on WPA to compromise a victim station.

Target station hardware*	OS	Attack successful?
Dell D630	Backtrack 4 Final	Yes
Dell D630	Windows XP SP3 X86	No†
Dell D630	Windows 7 X86	No†
Dell D630	Ubuntu 8.10	Yes
Powerbook	Mac OSX 10.5.7	Yes
Macbook Pro 5,5	Mac OSX 10.6.3	Yes
Macbook Pro 5,5	Windows XP SP2	No†
iPod Touch	iPhone OS 3.1	No†
Motorola Droid	Android OS 2.0	No†

** All victim station hardware used a Broadcom wireless card with the Ralink RT73 chipset.*
† No MIC failure frame was received.

Hole 196 Attacks

Defenders might be able to detect the Hole 196 attack with a WIPS by watching the broadcast fields of group data packets. This defense would likely change the use of a GTK to protect broadcast traffic.[12] This would require changing code for drivers in all the various OSs as well as the AP firmware. Such a change would require coordination among all Wi-Fi hardware vendors and standards committees. One downside of changing this type of broadcast traffic is poor performance.[12] Packets that were sent once for a group might have to be sent individually to each station, significantly increasing the Wi-Fi network load.

Another possibility is to use host-based intrusion detection software that looks for this type of activity. A program running on the host can tell whether the ARP cache has changed for no reason and warn the user. Two examples of host-based intrusion detection software that could serve this purpose are DecaffeintID and McAfee's EPO.[12]

Another countermeasure is to turn on the client isolation mode that some APs offer.[12] This effectively stops communication between stations through the AP. This would stop attacks that use the GTK to wirelessly poison a station's ARP cache because the AP would block the victim station's data from being sent to the attacker. The attacker still could send malicious payloads to the victim station, so this defense won't completely stop the attack. The obvious downside of turning on client isolation is that the station can no longer share files or communicate directly. Such peer-to-peer communication is routine on many corporate and residential networks. Thus, this countermeasure would greatly reduce functionality.

For the foreseeable future, the best approach to secure a wireless network (or any network) is a layered (defense in depth) approach. Besides using the right wireless encryption, a good practice is to establish a virtual private network (VPN) tunnel permitting stations to privately connect through an AP. The tunnel acts as an additional security layer making the data more difficult to decrypt. A VPN used with a WIDS and WIPS provides the fullest protection currently possible. Actively monitoring the network to identify anomalous

Smart Grids and Wireless Network Security

According to the International Energy Agency, the widespread deployment of smart grids is crucial to achieving a more secure and sustainable energy future.[1] The benefits should include greater grid reliability, integration of renewable generation and plug-in vehicles, reduced electricity peak demand, and stronger cybersecurity.

However, US energy utilities face a monumental challenge as they address compliance with the North American Electric Reliability Corporation's Critical Infrastructure Protection (CIP) standards. The increased use of wireless networking technology and its introduction into control center networks and field devices have compounded this challenge.[2] Consequently, utilities have had to establish electronic security perimeters (ESPs) to monitor, protect, and control their infrastructure.

Nevertheless, mobile devices capable of unauthorized wireless connectivity with wired and wireless interfaces could still access CIP-protected cyberassets in ESPs. For example, many utilities routinely permit the operation of cell phones, pagers, wireless bar code readers, and wireless-local-area-network connections in or near their control centers or substations.

Consequently, utilities need relevant security that helps reduce such cyberthreats, which are often quick, multifaceted, well resourced, and persistent. In addition, any efforts to modernize and protect such a critical cyberphysical infrastructure must balance the need for better performance and lower cost with better security.

A case in point is the current wave of smart meters (SMs) being deployed; more than 20 million SMs have been installed in the US.[3] Industry insiders have raised concerns that cryptographic-key-management systems only support SMs and might not scalably interoperate among other smart-grid domains. (Such domains include energy control systems; wide-area monitoring, protection, and control; synchronized phasor measurements; distributed and renewable energy resources; electric transportation; and energy storage.[4])

For situations such as this to change, stakeholders need tools, including cybereconomic incentives, for developing more adaptive and resilient Internet infrastructure. One such possible tool is the 2011 *Roadmap to Achieve Energy Delivery Systems Cybersecurity*.[5] "Increased insight from private-public collaborations will allow us to better protect the nation's energy delivery systems that keep our lights on and the power flowing," said US Department of Energy Secretary Steven Chu. "The 2011 Roadmap takes the necessary steps to strengthen the security and reliability of our country's electric grid, in a climate of increasingly sophisticated cyber incidents."[6]

Quantifying the return on investments in cybersecurity is difficult. Moreover, we know of no comprehensive plan to track the cost of losses due to failures caused by cybersecurity breaches in the energy sector. Yet, if the smart grid undergoes only a generational upgrade (that is, minimizing cost and maximizing performance) without designed-in security and resiliency (the ability, among others, to quickly and cost effectively adapt to new threats), then the cost to respond to new attacks might be catastrophic. Tracking the cost of security breaches is essential to determining inherent risk and developing effective incentives toward making cybersecurity ubiquitous.[7] Will the savings from a smarter, more efficient grid offset the costs derived from disruptive cyberintrusions?

References

1. *Technology Roadmap: Smart Grids*, Int'l Energy Agency, 2011; www.iea.org/papers/2011/smartgrids_roadmap.pdf.
2. T. Kuruganti et al., "Wireless System Considerations When Implementing NERC Critical Infrastructure Protection Standards," white paper, US Dept. of Energy, 2009; http://energy.gov/sites/prod/files/oeprod/DocumentsandMedia/NERCCIPwireless_whitepaper.pdf.
3. "Frequently Asked Questions: How Many Smart Meters Are Installed in the U.S. and Who Has Them?," US Dept. of Energy; www.eia.gov/tools/faqs/faq.cfm?id=108&t=3.
4. "Vulnerability Analysis of Energy Delivery Systems," US Dept. of Energy, Sept. 2011; http://energy.gov/sites/prod/files/Vulnerability%20Analysis%20of%20Energy%20Delivery%20Control%20Systems%202011.pdf.
5. *Roadmap to Achieve Energy Delivery Systems Cybersecurity*, Energy Sector Control Systems Working Group, Sept. 2011; www.controlsystemsroadmap.net/ieRoadmap%20Documents/roadmap.pdf.
6. "Department of Energy Releases New Roadmap to Guide Public-Private Cybersecurity Initiatives," US Dept. of Energy, 15 Sept. 2011; http://energy.gov/articles/department-energy-releases-new-roadmap-guide-public-private-cybersecurity-initiatives.
7. *Trustworthy Cyberspace: Strategic Plan for the Federal Cybersecurity R&D Program*, US Nat'l Science and Technology Council, Dec. 2011; www.whitehouse.gov/sites/default/files/microsites/ostp/fed_cybersecurity_rd_strategic_plan_2011.pdf.

behavior adds yet another layer of assurance for data confidentiality. Despite all these layers, if a phishing attack is successful, along with an elevation of access privilege on the host station, little is left in the line of defense to stop exfiltration from causing significant loss.

Understanding known attacks provides an important basis for exploring the attack surface of critical information assets (for example, data at rest) and infrastructures (for example, communication, command, control, and cyberphysical), with the goal of achieving resiliency.

Recreating known attacks is relatively easy. Moreover, the low cost of leveraging known attacks and discovering new attack vectors, and the ease with which such exploits are promulgated will remain significant factors in wireless networks' insecurity. Each new (or evolved) protocol must anticipate this problem, which exemplifies the Internet infrastructure's inherent weakness.

For a look at a real-world area in which this issue is crucial, see the sidebar. ■

Acknowledgments

This article was authored by a contractor of the US Government (USG) under contract DE-AC05-00OR22725. Accordingly, the USG retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for USG purposes. US National Science Foundation grant DUE-104394 partially supported this research.

References

1. Y. Xiao, "IEEE 802.11n: Enhancements for Higher Throughput in Wireless LANs," *IEEE Communications Magazine*, vol. 12, no. 6, 2005, pp. 82–91.
2. *IEEE Std. 802.11-200, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE, 2007; http://standards.ieee.org/getieee802/download/802.11-2007.pdf.
3. S. Frankel et al., *Establishing Wireless Robust Security Networks: A Guide to IEEE 802.11i*, US Nat'l Inst. Standards and Technology, 2007; http://csrc.nist.gov/publications/nistpubs/800-97/SP800-97.pdf.
4. J. Geier, "802.11 WEP: Concepts and Vulnerability," *Wi-Fi Planet*, 20 June 2002; www.wi-fiplanet.com/tutorials/article.php/1368661.
5. H. Cheung, "How to Crack WEP—Part 1: Setup and Network Recon," *Tom's Guide*, 2008; www.tomsguide.com/us/how-to-crack-wep,review-451.html.
6. H. Cheung, "How to Crack WEP—Part 2: Performing the Crack," *Tom's Guide*, 2008; www.tomsguide.com/us/how-to-crack-wep,review-459.html.
7. H. Cheung, "How to Crack WEP—Part 3: Securing Your WLAN," *Tom's Guide*, 2008; www.tomsguide.com/us/how-to-crack-wep,review-471-7.html.
8. F. Robinson, "Examining 802.11i and WPA," *Network Computing*, 26 Mar. 2004; www.networkcomputing.com/wireless/229622096.
9. F. Halvorsen and O. Haugen, "Cryptanalysis of IEEE 802.11i TKIP," Dept. of Telematics, Norwegian Univ. of Science and Technology, 2009; http://download.aircrack-ng.org/wiki-files/doc/tkip_master.pdf.
10. E. Tews and M. Beck, "Practical Attacks against WEP and WPA," *Proc. 2nd ACM Conf. Wireless Network Security*, ACM, 2009, pp. 79–86.
11. T. Ohigashi and M. Morii, "A Practical Message Falsification Attack on WPA," 2009; http://packetstormsecurity.net/papers/wireless/A_Practical_Message_Falsification_Attack_On_WPA.pdf.
12. "WPA2 Hole196 Vulnerability: Exploits and Remediation Strategies," white paper, AirTight Networks, 2010; www.airtightnetworks.com/fileadmin/pdf/whitepaper/WPA2-Hole196-Vulnerability.pdf.
13. G. Combs, "TShark—Dump and Analyze Network Traffic," Wireshark; www.wireshark.org/docs/man-pages/tshark.html.
14. D. Gupta, "The New Attack on WPA/TKIP: Much Ado about Nothing?," blog, AirTight Networks, 8 Sept. 2009; http://blog.airtightnetworks.com/the-new-attack-on-wpatkip-much-ado-about-nothing.
15. M. Kershaw, "Kismet Readme," Kismet, 2011; www.kismetwireless.net/documentation.shtml.
16. K. Hutchinson, "Wireless Intrusion Detection Systems," SANS Inst., 2004; www.sans.org/reading_room/whitepapers/wireless/wireless-intrusion-detection-systems_1543.

Frederick T. Sheldon is a senior research scientist at Oak Ridge National Laboratory. His research interests include developing and validating models, applications, methods, and tools for creating safe, secure, and dependable systems. Sheldon has a PhD in computer science from the University of Texas at Arlington. He's a senior member of IEEE. Contact him at sheldon@ieee.org.

John Mark Weber is a senior cyber engineer at Dynetics. His research interests include wireless security, computer network attacks, and supervisory control and data acquisition security. Weber has an MS in electrical engineering from the University of Alabama in Huntsville. Contact him at jark22@hotmail.com.

Seong-Moo Yoo is an associate professor in the Department of Electrical and Computer Engineering at the University of Alabama in Huntsville. His research interests include computer network security, wireless networks, and information assurance. Yoo has a PhD in computer science from the University of Texas at Arlington. He's a senior member of IEEE. Contact him at yoos@eng.uah.edu.

W. David Pan is an associate professor of electrical and computer engineering at the University of Alabama in Huntsville. His research interests include image and video processing and compression, channel coding, motion estimation, pattern recognition and classification, remote-sensing data processing and fusion, multimedia information security, and quantum computing. Pan has a PhD in electrical engineering from the University of Southern California. He's a senior member of IEEE. Contact him at dwpan@eng.uah.edu.