SANDIA REPORT

SAND2005-7408
Unlimited Release
Printed November 2005

INTRUSION DETECTION AND MONITORING FOR WIRELESS NETWORKS

Jamie Van Randwyk, Dimitry Averin, Ryan P. Custer, Jason Franklin, Franklin Hemingway, Dominique Kilman, Erik J. Lee, Mark Lodato, Damon McCoy, Kristen Pelon, Amanda Stephano, Parisa Tabriz, Eric D. Thomas

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.doe.gov/bridge

SAND2005-7408
Unlimited Release
Printed November 2005

INTRUSION DETECTION AND MONITORING FOR WIRELESS NETWORKS

Document Authors

Wireless Network Device Fingerprinting – Frank Hemingway
(Special thanks/acknowledgment to Ratish Punnoose)

Location-Based Authentication with Wireless LANs – Eric Thomas

OpenSource Wireless Security Tools – Dominique Kilman

Developer Notes for Sessionlogger – Jason Franklin, Mark Lodato, Dimitry Averin

Sessionlogger manpage – Jason Franklin, Mark Lodato, Dimitry Averin

Sniffer manpage – Jason Franklin, Erik Lee

Author Info

Dimitry Averin
averin@gmail.com
Polytechnic University

Damon McCoy
mccoyd@colorado.edu
University of Colorado, Boulder

Ryan Custer
rpcuste@sandia.gov
Sandia

Kristen Pelon
s1369781@cedarville.edu
Cedarville University

Jason Franklin
jfrankli@andrew.cmu.edu
Carnegie Mellon University

Ratish Punnoose
rjpunno@sandia.gov
Sandia

Frank Hemingway
fheming@unm.edu
University of New Mexico

Amanda Stephano
Amanda.Stephano@gmail.com
Indiana University

Dominique Kilman
dkilman@sandia.gov
Sandia

Parisa Tabriz
tabriz@uiuc.edu
University of Illinois at Urbana-Champaign

Erik Lee
ejlee@sandia.gov
Sandia

Eric Thomas
edthoma@sandia.gov
Sandia

Mark Lodato
lodatom@lafayette.edu
Lafayette College

Jamie Van Randwyk
jvanran@sandia.gov
Sandia

Contents

I. Introduction

Wireless computer networks are increasing exponentially around the world. They are being implemented in both the unlicensed radio frequency (RF) spectrum (IEEE 802.11a/b/g) and the licensed spectrum (e.g., Firetide [1] and Motorola Canopy [2]). Wireless networks operating in the unlicensed spectrum are by far the most popular wireless computer networks in existence. The open (i.e., proprietary) nature of the IEEE 802.11 protocols and the availability of "free" RF spectrum have encouraged many producers of enterprise and common off-the-shelf (COTS) computer networking equipment to jump into the wireless arena. Competition between these companies has driven down the price of 802.11 wireless networking equipment and has improved user experiences with such equipment. The end result has been an increased adoption of the equipment by businesses and consumers, the establishment of the Wi-Fi Alliance [3], and widespread use of the Alliance's "Wi-Fi" moniker to describe these networks.

Consumers use 802.11 equipment at home to reduce the burden of running wires in existing construction, facilitate the sharing of broadband Internet services with roommates or neighbors, and increase their range of "connectedness." Private businesses and government entities (at all levels) are deploying wireless networks to reduce wiring costs, increase employee mobility, enable non-employees to access the Internet, and create an added revenue stream to their existing business models (coffee houses, airports, hotels, etc.). Municipalities (Philadelphia; San Francisco; Grand Haven, MI) are deploying wireless networks so they can bring broadband Internet access to places lacking such access; offer limited-speed broadband access to impoverished communities; offer broadband in places, such as marinas and state parks, that are passed over by traditional broadband providers; and provide themselves with higher quality, more complete network coverage for use by emergency responders and other municipal agencies.

In short, these Wi-Fi networks are being deployed everywhere. Much thought has been and is being put into evaluating cost-benefit analyses of wired vs. wireless networks and issues such as how to effectively cover an office building or municipality, how to efficiently manage a large network of wireless access points (APs), and how to save money by replacing an Internet service provider (ISP) with 802.11 technology. In comparison, very little thought and money are being focused on wireless security and monitoring for security purposes.

Companies such as AirMagnet and AirDefense have established themselves by selling wireless security software and hardware. But the products they sell focus on office environments, network tuning and performance measurement, and existing attacks rather than novel wireless attacks. Many innovative tools have been developed in the open source community, but as the deployment of wireless networks expands, many opportunities for new tools, new areas of research, and improvements on commercial and freely available tools remain.

A quick look at a few threats that exist for wireless networks reinforces the importance of studying security and incident response in relation to these networks:

- *Rogue clients* are wireless devices (laptops, PDAs, printers, etc.) that are not authorized to use a wireless network. Additionally, many rogue clients could "spoof," or impersonate, the MAC (media access control) address (i.e., the hardware address) of an authorized client to gain access to network resources.

- *Rogue APs* are APs that are not authorized to serve a wireless network. As in the case of rogue clients, a rogue AP could spoof the MAC address of an authorized AP to launch man-in-the-middle attacks on the network.

- *AP flaws* are a broad category of threats to wireless networks:

    - Wireless APs may be (and are) manufactured and sold with errors in the implementation of the wireless specification (e.g., IEEE 802.11a/b/g/i); an attacker could exploit these errors.
    - Additionally, APs may be improperly configured or managed by the network administrator. Software patches and/or firmware updates might have been inconsistently applied or not applied at all.

- *Clients using outside APs* may not even know their connection to the outside AP exists. Whether the connection to the outside AP is made knowingly or not, this connection could put the wireless network—as well as the confidentiality of any information transmitted over the wireless network—at risk.

- *Sniffing traffic* is often the first step in attacking a network. An insider or an outsider could learn a lot about the network and the information exchanged on that network by simply listening to the data being passed freely through the air.

- *A variety of denial of service (DoS) attacks* exists for wireless networks.

    - A client (or clients) could flood the network with authentication requests.
    - Similarly, client(s) could flood the network with association requests.
    - Rogue client(s) could spoof deauthentication requests to deny service to specific clients.
    - A flood of request-to-sent/clear-to-send (RTS/CTS) messages could deny service to wireless devices on the network.
    - RF-level interference could increase noise on the wireless network to the extent that the network becomes unusable.
    - The injection of malformed frames into the wireless network could also disrupt network operations.
    - A device could repeatedly broadcast deauthenticate/disconnect frames, causing all devices to disconnect from the network.

- *Ad hoc networks* can affect infrastructure wireless networks in a variety of ways:
    - Client devices may create ad hoc networks without the user's knowledge.
    - Client devices may connect to ad hoc networks without the user's knowledge.
    - Ad hoc networks can interfere with infrastructure networks at the RF level (another DoS attack).

- *Misconfigured wireless network resources* provide additional avenues for attack:
    - The option to remotely manage the AP through a Web browser interface may not be access-controlled.
    - Weak or no encryption may be employed on the network.

During the course of this Laboratory Directed Research and Development (LDRD) project, we studied ways in which to both monitor and protect wireless networks, specifically those using the IEEE 802.11a/g protocols. We tried to think about future iterations of 802.11 and other wireless technologies such as WiMAX (IEEE 802.16), Bluetooth, etc. as we conducted our research.

The term "monitoring the network" includes activities such as intrusion detection, data capture, and visualization of network data. We studied monitoring for defensive and forensic purposes, not necessarily for network health.

Our research into wireless session/flow monitoring resulted in us writing a software tool named Wmon that can be run on embedded BSD or Linux devices, capturing wireless traffic, and then aggregating it at a central location for analysis and forensics. Wmon can weed out duplicate flows at the central aggregation point so that many sensors can be deployed to adequately cover a network. The software can also record every frame seen to a central storage facility for later forensic or research use. The aggregation point eliminates duplicate frames as well.

By leveraging earlier work in visualizing network traffic on wired networks, we built a library of network visualization functions and tools that enable us to view wireless networks in new ways. Visualization of networks allows us to detect attacks that text-based tools miss.

We also looked for new ways to protect wireless networks. This element of the research evolved during the course of the two-year project as commercial and open source tools and research were transformed over the same period. Our work in this area varied widely.

We spent time looking at wireless attacks at the RF level, specifically RF jamming attacks. We were successful in modeling these attacks and suggest several mitigations for this category of attack.

Additionally, we studied the idea of location-based authentication. The term "location-based services" [4] refers to the popular idea of tailoring network services based on a mobile device's location. We looked at the efficacy of using a wireless device's location as an additional criterion for controlling access to the wireless network. Much of the research focused on techniques to precisely and accurately locate wireless devices in three-dimensional space.

We also studied the area of wireless device fingerprinting. Fingerprinting is a technique whereby we passively or actively look at a wireless device to determine its various characteristics. A partial list of these characteristics includes the following: operating system and operating system version, wireless driver and wireless driver version, wireless device model number, and wireless chipset model number.

We spent a limited amount of time looking into fingerprinting devices at the RF layer, an area that shows a lot of promise. RF fingerprinting has been studied [5] in relation to preventing unauthorized cloning of cellular telephones, but little has been published regarding the fingerprinting of Wi-Fi devices.

We have also identified many aspects of the 802.11 MAC-layer protocols that lend themselves to enabling device fingerprinting, both passively and actively. We selected one characteristic, time deltas between probe request management frames, and developed a database of fingerprints and a tool to identify wireless drivers based on that database.

References

[1] "Firetide: Instant Mesh Networks," Firetide, 2005. [Online]. Available: http://www.firetide.com/

[2] "Motorola Canopy," Motorola, 2005. [Online]. Available: http://motorola.canopywireless.com/

[3] "Wi-Fi Alliance," Wi-Fi Alliance, 2005. [Online]. Available: http://www.wi-fi.org/

[4] R. José and N. Davies, "Scalable and Flexible Location-Based Services for Ubiquitous Information Access," in *HUC '99: Handheld and Ubiquitous Computing: First International Symposium*, 1999, pp. 52–66.

[5] T. Fawcett and F. Provost, "Adaptive Fraud Detection," *Data Mining and Knowledge Discovery*, vol. 1, pp. 291–316, September 1997.

II. Augmenting Wireless Intrusion Detection with Visualization

Abstract

Wireless networks make intrusion detection difficult because of physical network layout, complex physical interactions, and unintelligible protocols. We have addressed some of this complexity using data visualization tools to display real-time network data to human operators. This allows the operator to gain a quick understanding of the network's overall state. Using these tools, the operator can quickly isolate regions of interest in the network and drill down into them for a closer look. This capability greatly reduces the cognitive load on the operator and promotes fast and effective responses for many types of intrusions. Visualization technology has also proven useful in imparting knowledge to its users, so that they are able to better understand the network and improve the automated defenses accordingly.

Introduction

An effective Intrusion Detection System (IDS) must be able to recognize familiar threats as well as identify potentially new threats. Automated systems are currently very effective in the first of these two categories; that is, a well-designed automatic IDS can very quickly identify a network intrusion if that intrusion has a previously seen pattern of data. However, the state of the art in anomalous activity detection (which is required for recognizing new threats) is very unreliable in automated systems. These systems lack the reasoning ability that is crucial for making decisions about anomalous data that may or may not be a threat, with the typical consequence of an extremely high false positive rate. For this reason, we have designed a system that incorporates the reasoning abilities of a human operator through the mechanism of a visual network representation. Using this system in combination with a traditional automated IDS may lead to significant gains in the overall effectiveness of the IDS.

Visual representation of network data supplies a human operator with an interface and a metaphor to perform data analysis. This allows the operator to draw conclusions about patterns in network traffic, and quickly isolate the types of traffic that are of interest. The knowledge gained from this interaction can then be applied to the automated IDS and the overall network design to improve network security.

The power of visualization technology comes from its parallelism. Without some form of data visualization, an operator is required to either rely on automatic tools (with their inherent limitations) or to do a linear trace of huge amounts of largely irrelevant data. Visualization allows the operator to view the data in parallel as a dynamic graphical image. The operator can then tune the image to emphasize the aspects of the information that are of interest, so that the irrelevant data attracts less attention. This provides a holistic view of the network at large, alleviating the problem of "not being able to see the forest for the trees."

Visualization Techniques

Our visualization system uses several metaphors and data representations to convey information about the state of a network. These representations are not necessarily orthogonal to one another. We have found that allowing overlap in information between representations helps the operator develop a consistent understanding of the network.

There are many ways to aggregate network data into a graphical representation. Our approach to this problem has been to identify entities of interest and display relationships between those entities as they evolve over time. For example, a particular visualization tool may display hosts on a network as entities, and as those hosts communicate with each other it may change their relative positions and colors to indicate the discovered properties of their communication.

It is important to note that the visualized entities are purely abstract. This means that an entity could be designed to represent any type of network object, whether or not it has physical meaning. As an example, consider a network that is intended to pass only encrypted data. On this network, it is very important to monitor the connections and ensure that they are running under proper cryptographic protection. A visualization tool whose emphasis is on data elements would be most appropriate here. The tool could continuously sample the entropy of connections between machines, and display these connections as colored objects on the screen, with the color indicating the measured entropy. Using this interface, an operator could quickly identify most types of traffic being sent unencrypted because they will appear as a different colored graphical entity. After identifying the offending connection, the tool's interface would identify the endpoints of the communication so that the operator could address the problem quickly.

Another important technique we support in our visualization tools is cross-tool entity correlation. The tools support a uniform representation for entities. When one tool has an entity selected, it can inform any other tools that the selected entity currently is receiving attention. If the other tools are also displaying a representation of that entity, they then have the option of changing their displays to emphasize the same entity. This allows the operator to correlate the information provided in several formats to achieve maximal understanding.

Visualization Tool Architecture

The visualization tool suite we have developed was designed with extensibility as a primary goal. It is apparent from the ideas presented above that providing the operators with the ability to modify and extend the tools is extremely useful. To this end, the tool suite is entirely plugin-based. Core capabilities relating to rendering and data processing are provided as library functionality, and the plugins have a well defined and powerful method of communication.

The core architecture of the application is based on a data flow model of traffic analysis. Plugins are created with an interface allowing them to accept arbitrary types of data as input, and generate arbitrary types of data as output. The input/output relationship is many-to-many, so plugins have the ability to aggregate data from different sources and provide different types of processed information as output to subsequent plugins.

Each plugin also has the option of providing a graphical view using a very flexible 3D interface class provided by the application core. As data enters the system from source plugins, it is passed along chains of information analysis plugins. Each analyzer has the chance to examine its particular area of interest, extract relevant information, and pass on that information. This data flow architecture is itself implemented as a graphical drag-and-drop style interface that allows the user to select the analysis plugins of interest and connect them in any way that makes sense. A data type checking system inherent in the connections themselves prevents invalid connections.

Figure 1 shows an example configuration. In this figure, a live network sniffer is used as the source plugin that is responsible for monitoring the network and importing network data into the application. The Sniffer plugin then passes data to a series of analysis plugins. In this example, analyzer plugins correspond to layers of the OSI seven layer network model. The first analyzer plugin inspects the link layer header on each packet it receives and processes it appropriately. This includes recording Ethernet addresses from the header and tracking relationships between entities, such as sender and receiver.

Once the Ethernet header is analyzed, the header is removed and the network data is passed to the next analyzer plugin. This is not strictly necessary, and the Ethernet plugin or any plugin for that matter, could pass along the all of the network data without stripping off the header. This approach has two major benefits.

First, it reduces the amount of work of each plugin at higher levels in the protocol stack. Each plugin only needs detailed knowledge of a single protocol and does not have to manipulate the network data as much. This eliminates repetitive implementation of network data handling code or calls to library functions. As a result, the analyzers share the strengths of the OSI seven layer network model.

The second advantage is that a visualization plugin can tap into any point, or set of points, along the analysis chain. This makes it easy for a visualization to collect data from several analyzer plugins and also reduces the work each visualization plugin must do to display data of interest. The process of receiving network data, analyzing the appropriate portions, and passing processed data to higher level analyzers is followed up the protocol stack as desired. Of course, eventually raw data is reached, which may or may not have an analyzer plugin. In Figure 1, the analysis chain proceeds to the network layer of the OSI seven layer network model and stops. The endpoints are visualization plugins, which are described below.

In addition to the analysis plugins, the application provides a powerful aggregation engine for bringing together the multiple types of information gleaned from the data flow processors. Each processor can post information to this engine, and the engine automatically maintains relationships between the entities involved. These relationships can then be explored to gain further information from a higher level perspective.

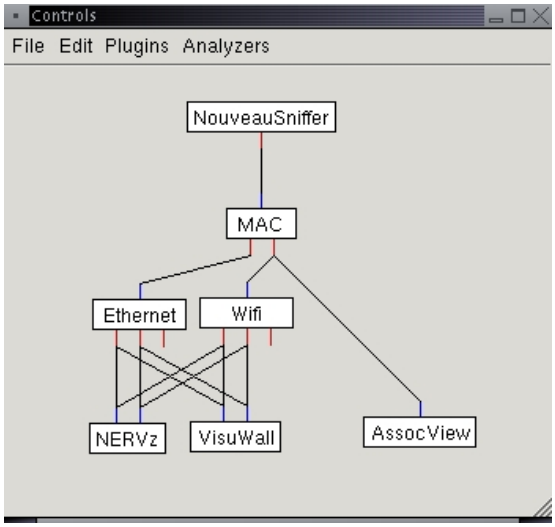


Figure 1. An example analyzer configuration.

Specific Tool Descriptions

This section describes several visualization plugins we have implemented to test these concepts. Each plugin has a slightly different emphasis to its display. Currently, visualizations have been developed for typical IEEE 802.3 wired networks and IEEE 802.11 wireless networks. Each of the plugins described below visualizes data from one of these network types. Applications that analyze protocols at or above the link layer can be used for either type of network. All of the wireless plugins focus analysis on the IEEE 802.11 protocol and rely on other visualization to provide representations of higher level protocol data.

ApLocate

ApLocate is the simplest of the visualization plugins. It was designed to give the operator an idea of the physical proximity of wireless Access Points to each other. The central entity of ApLocate is the access point, and is rendered as a sphere. The secondary entity represents a client transition between access points, and is rendered as a line that appears at the time of the event and fades away over a period of about one second. As more information is gained about the relative positions of the access points in the network, they move in relationship to each other.

The algorithm used to infer the access point positions is based on client transitions. Assuming a uniform distribution of wireless clients, access points that are close together should see more shared transitions than those that are farther apart. This information can be used to determine a logical separation of the access points, and that logical separation is displayed graphically as a distance in the application. Final positioning is done by solving a spring-mass model where the access points are masses in a viscous fluid and connected to each other by springs of varying spring constants. As clients transition between access points, the spring constant for the spring corresponding to the transition is increased, bringing the affected access points closer together. In the absence of springs, the access points repel each other with a constant force to prevent the whole mass from contracting to a point in the center.

This algorithm, while only a very rough approximation to actual physical proximity, provides valuable information about network usage and hotspots to the operator. Improbable hops are also emphasized, because the respective access points will be far apart in the view and the hop will be rendered as a line that is much longer than ordinary. This can serve as a flag to indicate the use of a high gain antenna, or a user outside of the typical usage area of the network.

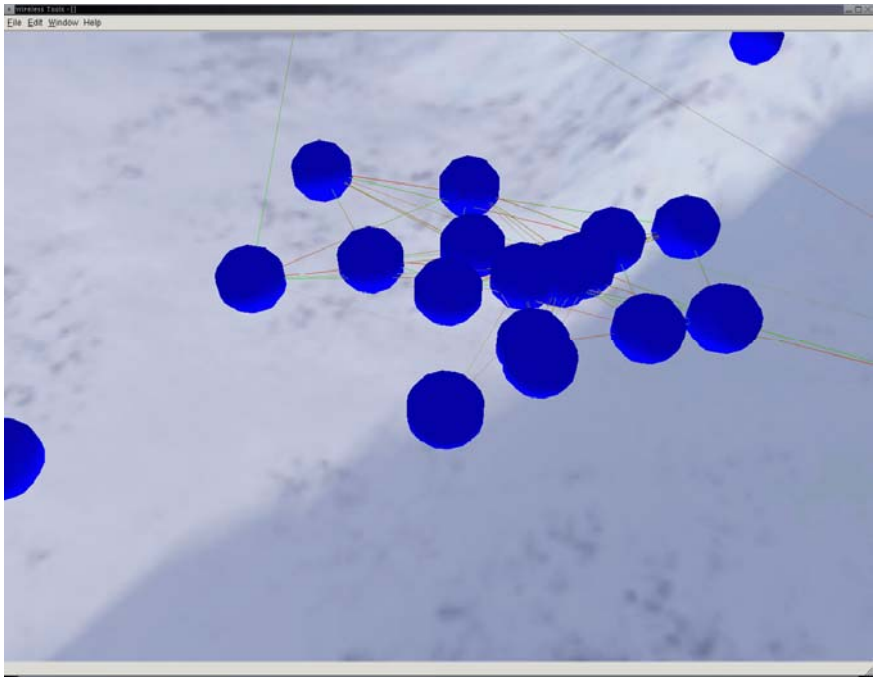


Figure 2. APLocate representation of wireless network client transition between access points.

AssocView

This visualization application concentrates on the IEEE 802.11 link layer. It represents access points as red spheres and clients as spheres of different colors. Clients that are associated to a particular AP are tan in color and are place in a cylinder under the AP with which they are associated. Clients that have authenticated, but are not associated to an AP are bright green. Client that have been detected, but without association information, are white and scattered along the “undetermined” plane. The tan clients associated to an AP also draw an opaque blue line to the AP with which they are associated currently. These hosts also draw an opaque yellow line to any AP with which they have an authentication. IEEE 802.11 data packets are represented as purple lines between two entities. Finally, packets sent in IEEE 802.11 ad-hoc mode are represented as green opaque lines drawn between two entities.

In addition to these representations, AssocView also runs a positioning algorithm to position access points to indicate that the APs share clients. If a client is associated to one access point and authenticated to several others, the APs are placed in a ring around the client. This process is repeated for each client, and each AP that has not been previously positioned is placed in the appropriate spot relative to the client. The result is a “web” of APs that have at least one client relationship in common. If there is a very strong relationship among APs, a large number of opaque, yellow association lines will connect clients of one AP to other APs nearby. As a result, an operator can tell at a glace which APs share a large number of clients and are most likely physically close together.

The algorithm described above is not perfect, and some overlap of APs and clients occurs as a result. With a more exhaustive technique, the APs could be placed to prevent overlap that obscures important relationships. This version of the tool concentrated effort on providing a reliable and responsive interface to an operator. In future versions of the tool, effort will be dedicated to perfecting the positioning algorithm to display client to AP relationships more effectively.

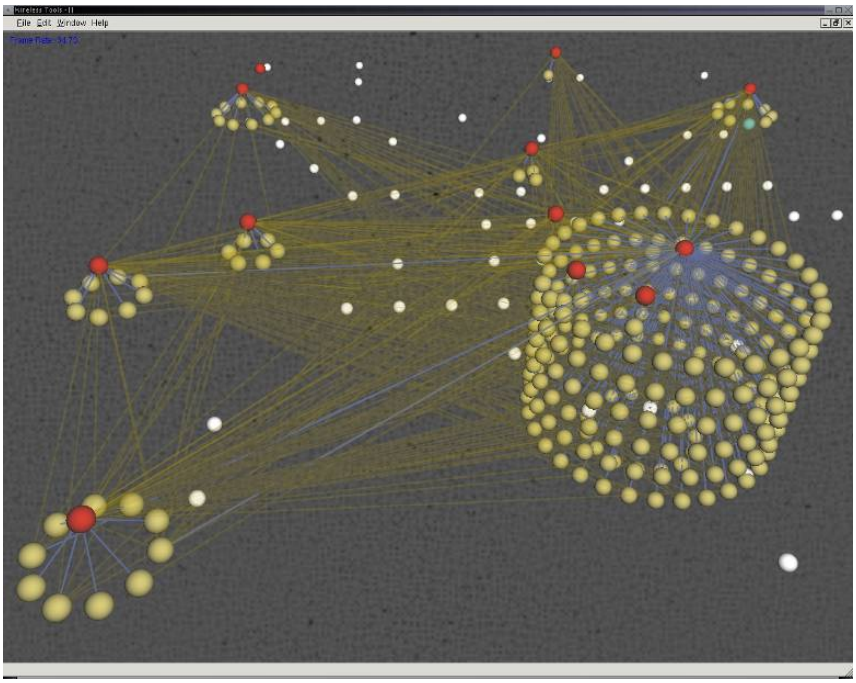


Figure 3. AssocView representation of associations and authentications among wireless clients and access points.

NERVz

NERVz is a visualization of the MAC, IP, UDP, and TCP layers of network traffic. The entities of this visualization are individual IP addresses (usually corresponding to a physical host) and packets. Packets are further split into TCP and UDP protocols, with any other packet being rendered as a cyan line. TCP and UDP packets are rendered as a two segmented line, with one end being blue and the other end red. The blue end of the line is the source end of the packet, and the red end is the destination. These packets pass through a third point between the other two, which corresponds to a port number and protocol indicator. Every TCP packet will start at a host (represented by a sphere), travel to a point on the upper plane, and then down to another host. The plane represents a mapping of all 65,536 possible port numbers, decomposed into a grid of 256 by 256. UDP packets use an identical scheme, except that they use the bottom plane. Each unique IP address discovered on the network is represented as a colored sphere. The color is determined by the manufacturer portion of the MAC address (the first three bytes). IP addresses are positioned according to their respective usage of the network. As more traffic is sent than received, the host will move toward the left side of the display. The opposite applies for receiving more than sending. The total bandwidth used by the host determines its vertical position. This position is calculated so that the hosts will be sorted from top to bottom according to their relative bandwidth usage. ARP (Address Resolution Protocol) messages are visualized as an expanding red sphere centered on the requesting host. The reply returns from the answering host as a cyan line.

This view of the network provides a tremendous amount of information to the user about the various interactions of individual hosts on the network. It is possible to quickly determine the most commonly used services, most active hosts, and general structure of the network traffic. Other important indicators of network health and security (such as the ARP notification and color-coding of the hosts) allow an operator to get a quick status report for the network. An interesting consequence of the bandwidth positioning algorithm is that servers tend to rise to the top center of the view. Using this interpretation, an operator can rapidly see when a server has failed or a different machine has taken its place (such as would happen during a man in the middle attack).

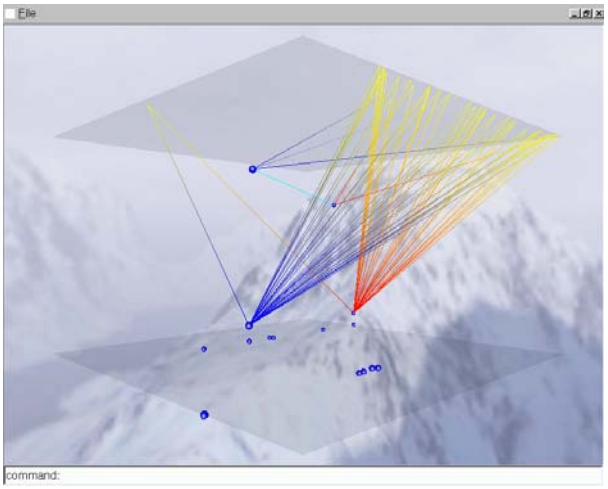


Figure 4. NERVz representation of an Nmap scan.

ServiceLakes

This visualization is concerned with the representation higher level services. Hosts are represented as randomly placed spheres on an elevated plane. As a host uses services, such as Domain Name Service (DNS), a small disk for that service appears on a lower plane. These service disks have the appearance of lakes, for which this tool was named. As the utilization of that service increases, the disk grows in diameter. As the utilization decreases, so does the diameter of the disk. In this way, the popularity and relative usage of services on a network can be seen.

If a service is used frequently, the disk will have a larger diameter most of the time, while services that are rarely used will remain smaller in diameter. This interface also has the ability to remove inactive host and services from the representation, or display only the most popular hosts and services. The interface also allows that operator to change the labels of services and to name hosts.

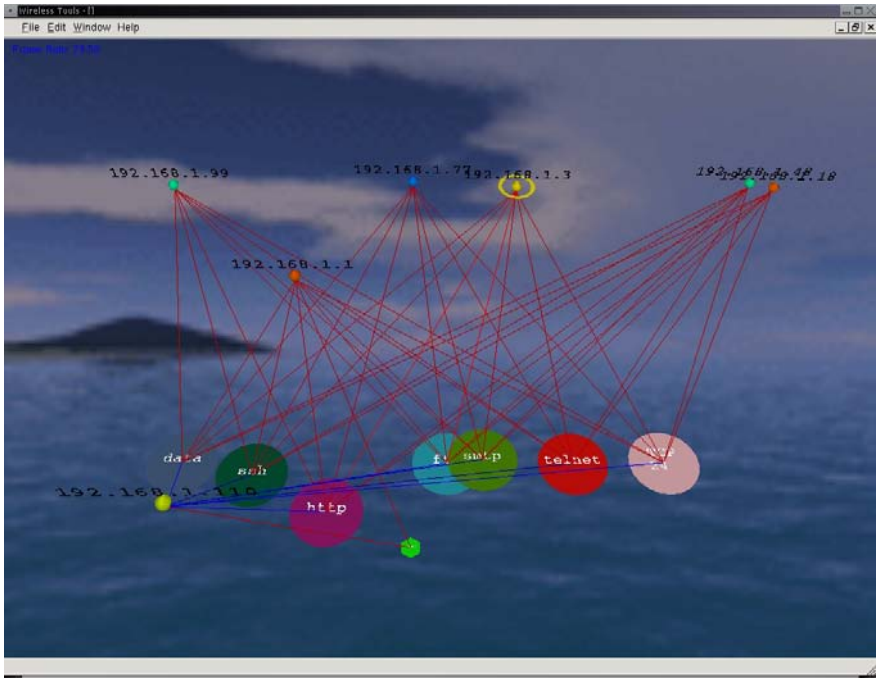


Figure 5. ServiceLakes representation of an Nmap TCP port scan of on a small network.

VisuWall

This visualization, like the NERVz application, concentrates on the IP, TCP, and UDP layers, as well as the ICMP and ARP layers. This visualization displays class B source and destination IP address spaces as rectangular planes. The third part of the dotted decimal address (e.g., xxx.xx.16.x) is used to position links along the x-axis of the source and destination address planes. The fourth part of an IPv4 dotted decimal address (e.g., xxx.xxx.xx.1) is used to position links along the z-axis of the source and destination address planes. The ICMP, TCP, and UDP layers are represented as rectangular planes with the size of the packet along the z-axis of each plane. ARP request are represented as "red flares" that originate at the sender's location on the source address plane. A large, transparent plane represents the 65,535 ports used by TCP and UDP, and is situated in a gap between the source and destination address planes. A link coming from the source address plane hits the appropriate location on the port plane (zero for protocols that do not use ports) and is "refracted" to each appropriate plane.

As an example, a TCP packet sent to port 80 would originate from the appropriate point on the source address plane, hit the port plane at the port 80 location, and "refract" to the appropriate point on the destination address and TCP planes. The reply would follow an similar process, with the roles of source and destination being switched and the reply being sent to a different port.

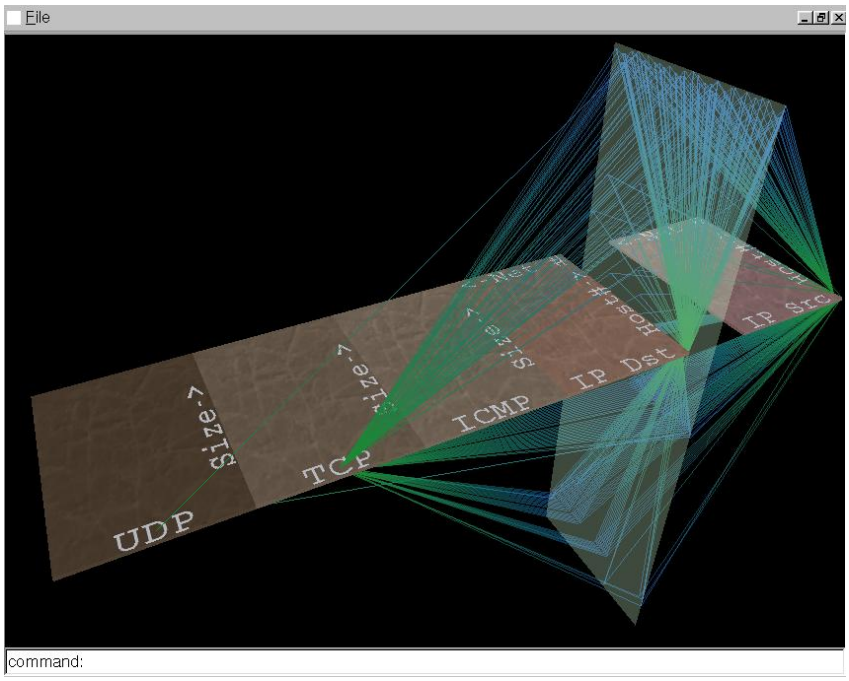


Figure 6. Figure 1 - VisuWall representation of an Nmap TCP port scan.

WState

WState is a visualization of the link layer of 802.11 wireless networks. It distinguishes between three types of entities: an access point, rendered as a spool; a client, rendered as a sphere between the ends of the access point it is associated to; and a packet, rendered as a line. This visualization was developed to monitor movement of clients on the network, and determine patterns in that movement. In addition, it maintains a Markov model of the transitions between 802.11b frame types to provide some automatic anomaly detection for link layer attacks. As clients move back and forth between access points, their respective spheres move in and out of the region of influence of the access point spools. As access points become more heavily loaded, their spools increase in height. This serves to bring attention to critical areas of the network, and possibly indicate areas where better load balancing is needed. When an entity on the network sends an improbable sequence of frames it will begin flashing red. The probability of any given frame type transition is continually updated based on the observed behavior of the network. When the host's behavior returns to normal for a sufficient period of time, it will cease flashing. This can flag both malicious behavior (such as a management frame flood) and problematic behavior (such as failure to associate to an access point).

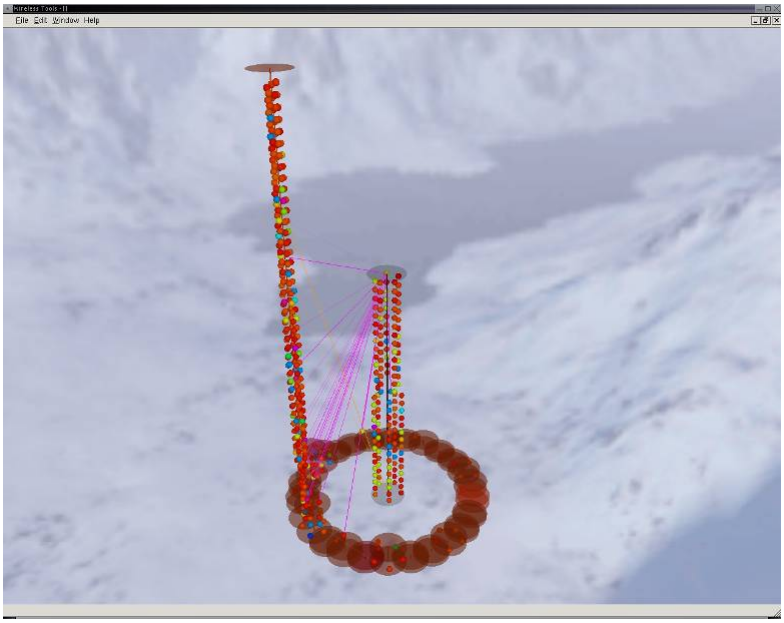


Figure 7. WState representation of associations and network use of large wireless network.

Conclusion

We have created a suite of tools that allow operators to visually analyze network data. Presenting network data through visualization reduces the time and effort needed to identify complex patterns and relationships. This caters to strengths of a human operator to recognize patterns, and uses this understanding to create signatures for traditional intrusion detection methods. This technique provides an effective way to help an operator understand the behavioral aspects of a network and supplements traditional intrusion detection methods.

The tool suite also facilitates the rapid creation of new tools to highlight network data in new ways, and as a result present easily understandable data. The quick creation of new tools can be used to create new visualizations of new protocols or network data. It is important to have multiple visualization tools, as using different visualizations in tandem can help an operator gain knowledge that would be difficult or impossible with a single tool. To this end, a relationship engine tracks data collected by each tool so that relationships can be inspected using several tools at once.

Specific descriptions of analysis tools that have been created so far were presented to give the reader an understanding of some of the different ways network data can be visualized.

This page intentionally left blank.

III. Wireless Intrusion Detection and Mitigation at the RF (Physical) Layer

RF Defined

Radio frequency (RF) refers to any frequency within the electromagnetic spectrum associated with radio wave propagation. Wireless communication uses RF by supplying current to an antenna, thereby creating an electromagnetic field that is able to propagate through space. Because the signals employed by wireless communication use space, or air, as their physical medium, the communication channel is open to any and all who might wish to listen in, or to transmit over the same medium.

RF jamming occurs when a signal disrupts the communication of wireless devices. This can happen naturally due to RF noise produced by objects like microwave ovens. A more serious concern involves RF jamming techniques used by malicious attackers. These attackers broadcast noise in order to disrupt communication for other users. An RF jamming attack results in a denial of service (DoS) for the wireless network.

The 802.11 networks (a/b/g) all use RF to communicate. The widespread use of this technology leads to widespread use of RF jamming techniques as well. Without some way to detect and mitigate jamming attacks, the utility of wireless networks may be destroyed before the full potential of the technology can be realized.

RF Vulnerabilities

Several features of wireless communication makes it susceptible to jamming attacks. First, the physical medium (i.e., air) through which wireless talks is shared and open to all. This results in a medium that can be easily eavesdropped. In addition, access to this open channel is nearly impossible to control. There is no good way to prevent someone from accessing the wireless communication channel; the only alternative is to limit the results of such access.

The decreasing cost of wireless equipment also increases the risk of jamming attacks. As the availability of hardware that allows access to the channel increases, the pool of potential attackers also increases. These attackers also create and share standard attack scripts. The few knowledgeable attackers make tools for attacking networks available to others. These “script kiddies” who would have been unable to attack even the simplest, most poorly configured networks on their own now have an easy way to attack any network they can find.

As the cost of ownership declines for wireless users, so does the availability of commodity hardware. Because the field of wireless communication is growing rapidly, the need to standardize communication also grows. This again makes life easier for the attacker who only has to learn a few standard, well-known protocols.

RF Mitigation Strategies

Several groups including Sandia National Laboratories have put time and effort into developing ways of mitigating and detecting RF jamming attacks.

In Xu et al. [1], two ideas are suggested for avoiding wireless DoS: channel surfing and spatial retreats. The first idea advocates all nodes in a network agreeing upon a channel switching strategy. If these nodes suspect they are being jammed, they can use this previously agreed upon strategy to move to a different communication channel. The second strategy makes use of the mobile nature of wireless nodes. With this strategy, the nodes in the network set up a “retreat pattern” ahead of time. If the nodes detect RF jamming, they then use this retreat path to move out of the range of the jamming nodes. This strategy is designed for a military environment where coordinated movement of nodes is possible.

In Wood et al. [2], the strategies for dealing with jamming attacks recommend that the nodes create an internal map of the jammed area. Once the nodes can determine the area of interference, messages can be routed around these areas. This allows the network to remain connected in the presence of malicious jamming attacks.

An additional research effort is pursuing ways to use RF jamming to control wireless use. This research is in its early stages and may never be adopted. Applying this type of research would allow building and campuses to control where wireless networks are made available. By jamming certain areas of a campus, the administrators of a network could create secured areas where wireless communication could not take place.

OPNET

The OPNET simulation tool shows the effects of RF jamming on a wireless network by comparing the throughput and response time graphs for a network in the presence of a jammer to the same graphs of that network without the jammer. The dramatic decrease in performance created by a single jammer node demonstrates the risk that all wireless networks should be aware of. The work done in Acharya et al. [3] shows the effects of intelligent jamming on a network. In this research, the energy usage of the jammers is an important consideration. The researchers show that by selectively targeting when and where to jam, the energy needed by the jammer is reduced significantly.

Configuration

The following OPNET simulation shows the effect of a jamming node on network communication.

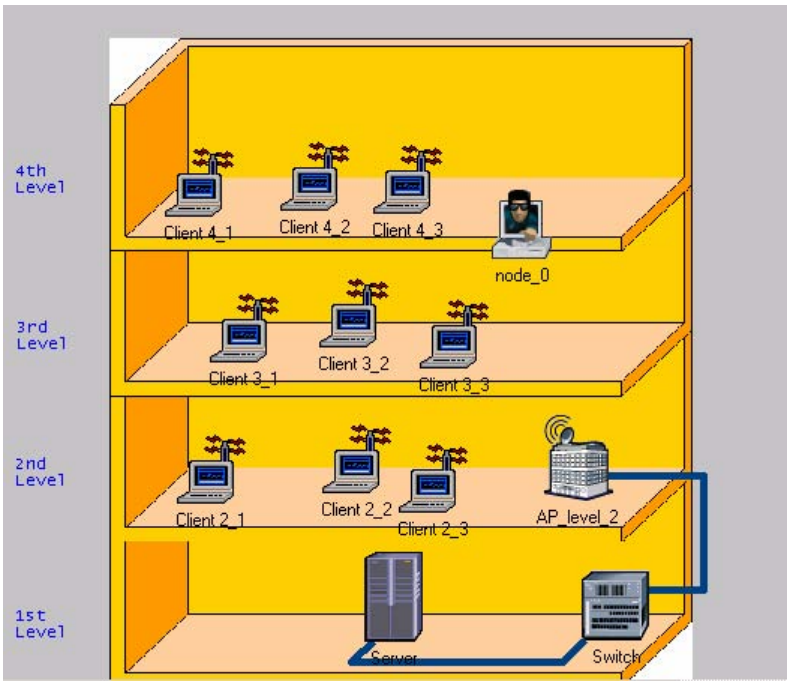


Figure 8. Jam Scenario

The scenario consists of a four-story building with a single access point (AP) and nine client machines that connect to the AP. In the baseline simulation, nodes communicate normally with one another and the base station. In the jammed scenario, a jammer node (node_0) broadcasts packets to disrupt communication.

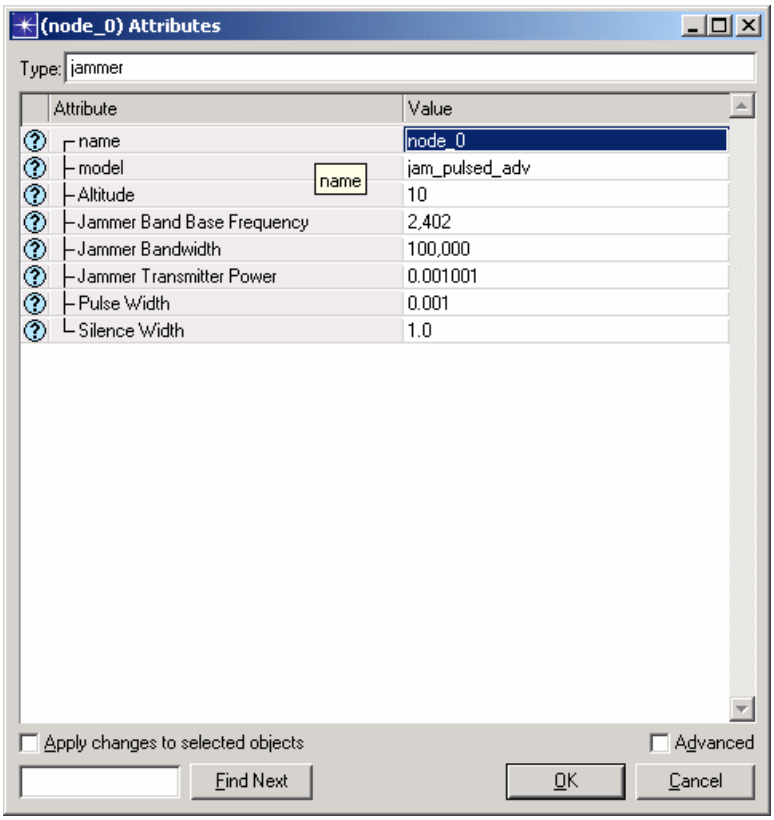


Figure 9. Jammer Configuration

The jammer node is very simple. The process consists of scheduling packets, sending a packet to last the proper duration, and then scheduling the next packet to be sent. The jammer node transmits for 0.001 seconds continuously and then waits for a period of 1 second. The packets being sent by the jammer are not received by either the clients or the AP. These packets serve as RF noise on the channel.

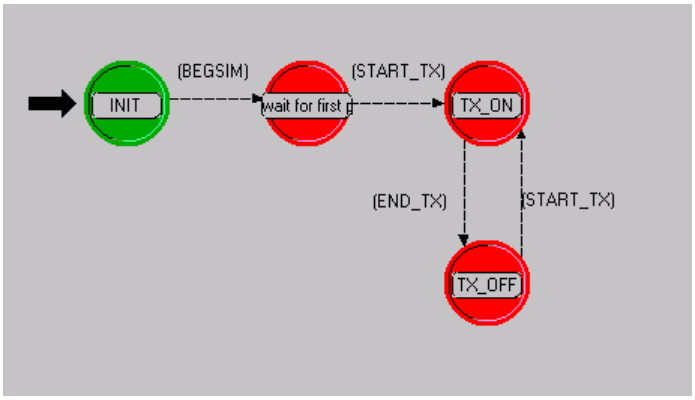


Figure 10. Jammer Process Model

Results

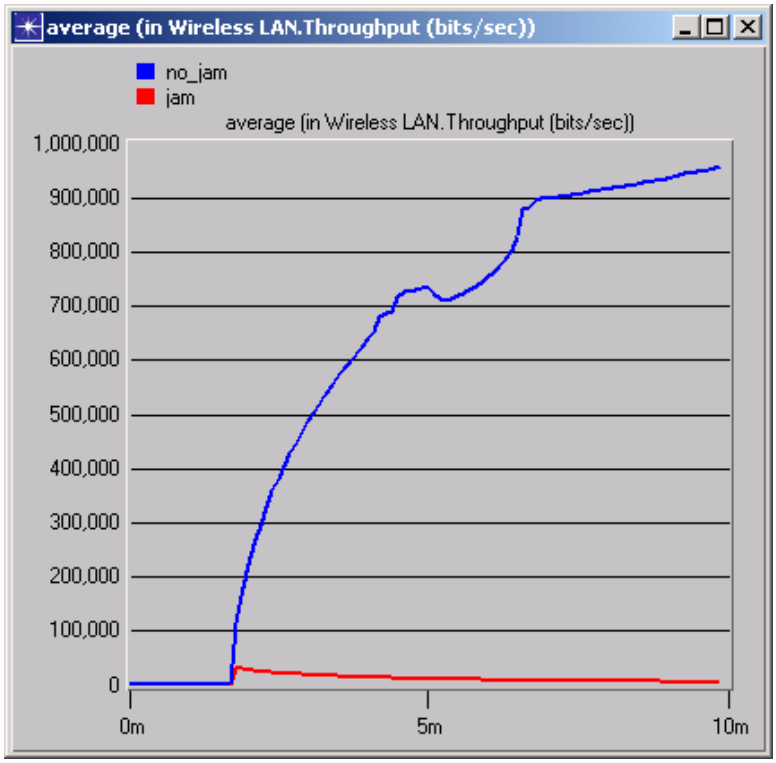


Figure 11. Throughput Comparison

As can be seen in the throughput comparison graph, having a jammer in the network significantly affects throughput. Almost no data gets through once the jammer begins to disrupt communication.

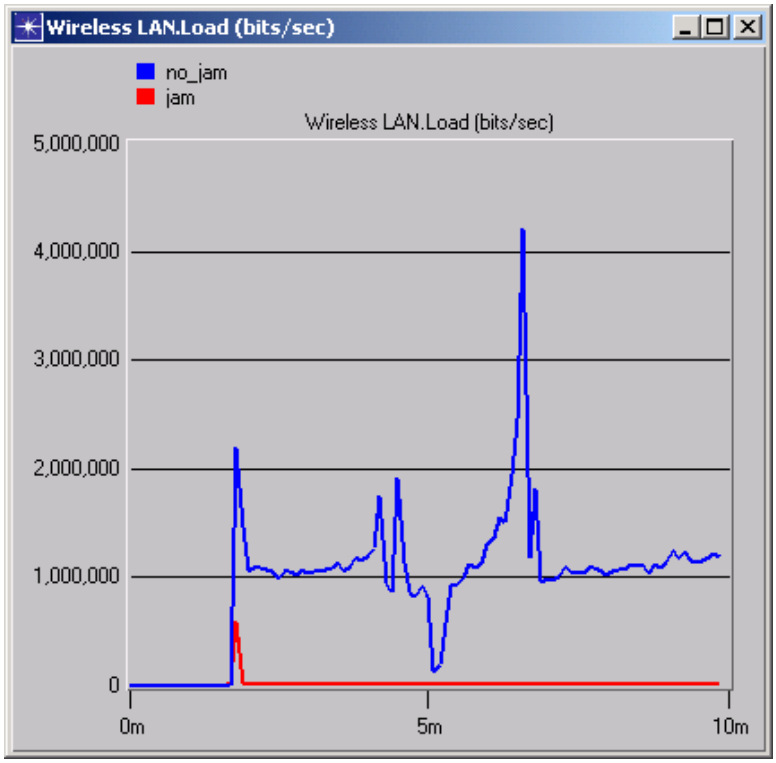


Figure 12. Load Comparison

Because real communication has stopped in the network, the load of the network drops to 0. None of the nodes are able to communicate their packets.

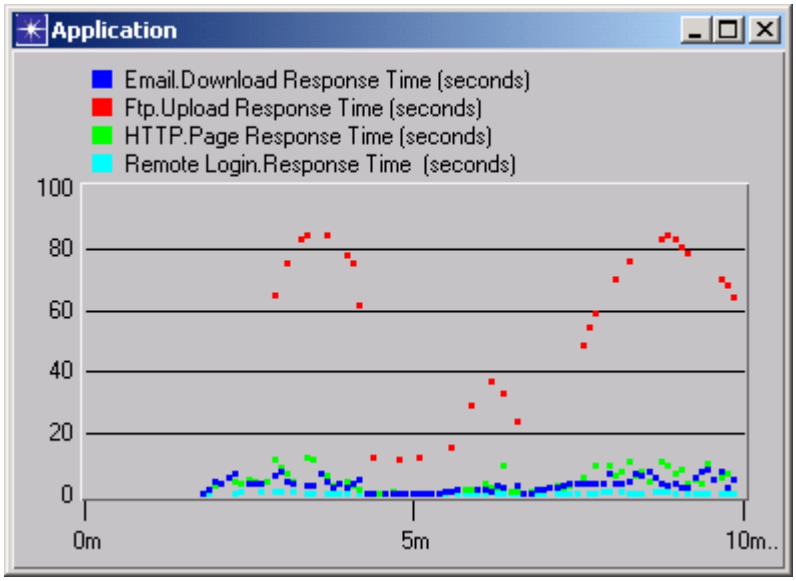


Figure 13. Application Details—No Jam

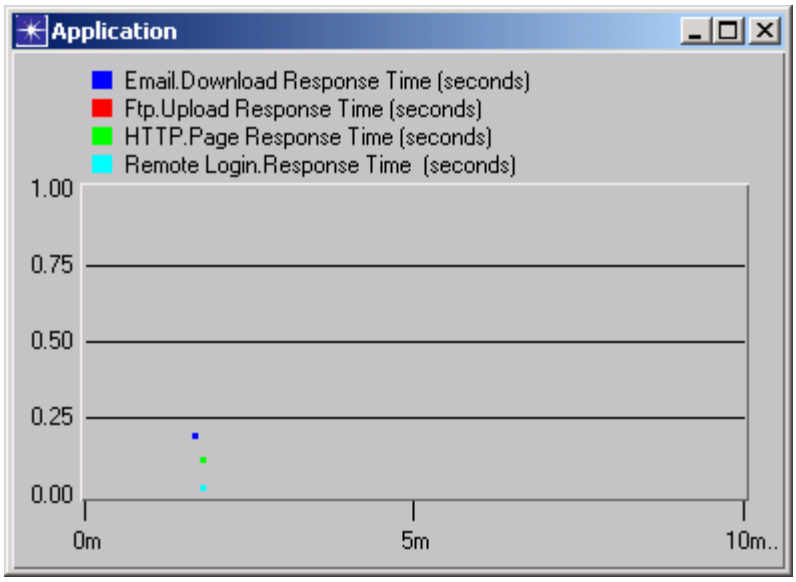


Figure 14. Application Details—Jammed

The two application detail graphs reveal the details of what should have been communicated. In the normal scenario, FTP sometimes gets a slow response, but each of the four applications receives a response. In the jammed case, FTP never responds, and the other three applications have a single transaction completed before they are also shut down.

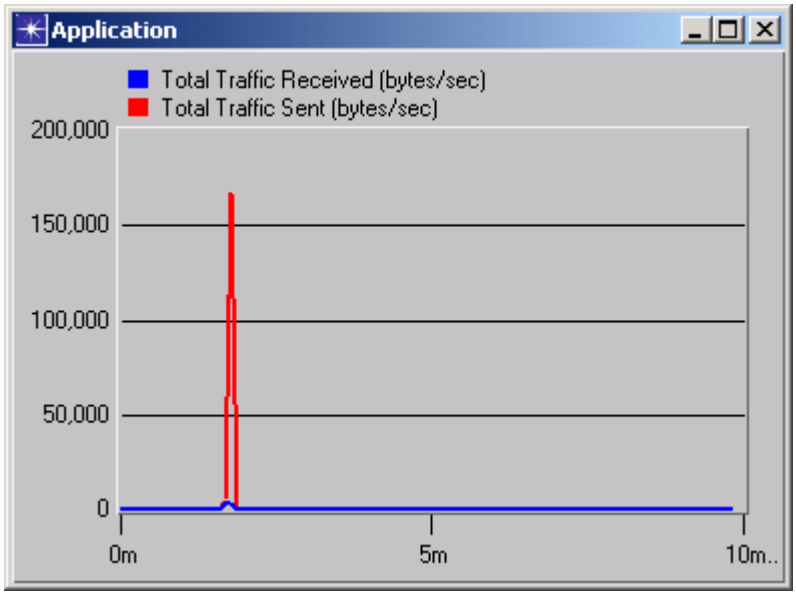


Figure 15. Total Traffic—Jammed

Another result, not seen in the graphs, is the number of errors generated during the simulation. When the jammer node is active, the number of errors jumps to 200. Most of these errors indicate that the client nodes could not establish a connection with the server. The root cause of this problem is network congestion caused by the jammer node.

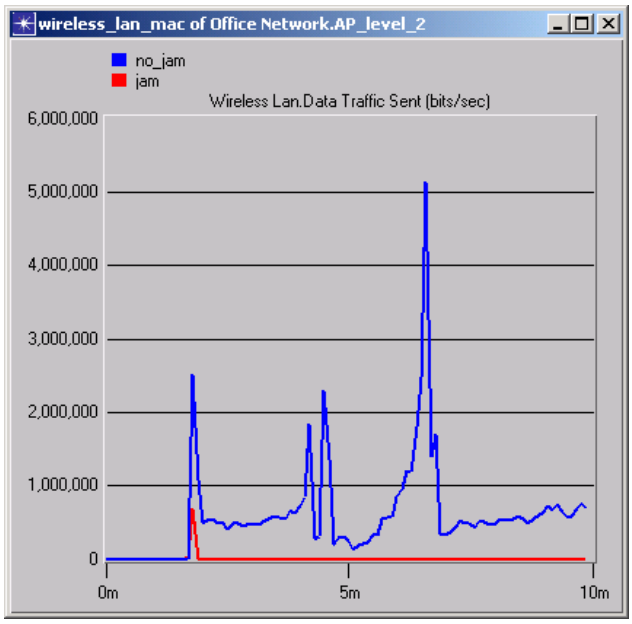


Figure 16. AP Traffic Sent

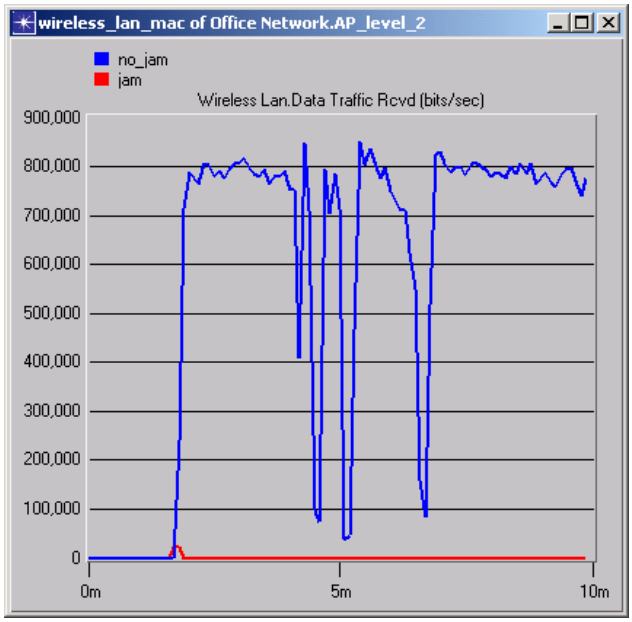


Figure 17. AP Traffic Received

Figures 16 and 17 show that the AP is neither sending nor receiving data in the jammed scenario.

Future Work

Ultimately, OPNET could also be used to test mitigation strategies. A possible technique to consider with OPNET is allowing the APs to communicate on a wired network. By using a wired communication channel, the APs would be able to identify a relative location for the node that is sending RF jamming signals onto the network.

Wireless intrusion detection capabilities can be added to the OPNET scenarios to test the effect that an intrusion detection system (IDS) would have on the jammers, as well as the effect of the IDS on other nodes within the network.

Other work with the OPNET tool could be used to show the effect of multiple, coordinated jammer nodes. Multiple jammers could negate the usefulness of some of the mitigation strategies. They may also evade detection by an IDS.

References

[1] W. Xu, T. Wood, W. Trappe, and Y. Zhang, "Channel Surfing and Spatial Retreats: Defenses against Wireless Denial of Service," in *2004 ACM Workshop on Wireless Security*, October 2004.

[2] A. Wood, J. Stankovic, and S. Son, "JAM: A Jammed-Area Mapping Service for Sensor Networks," in *2003 IEEE Real-Time Systems Symposium*, December 2003.

[3] M. Acharya, T. Sharma, D. Thuente, and D. Sizemore, "Intelligent Jamming in 802.11b Wireless Networks," in *OPNETworks 2004*, August 2004.

This page intentionally left blank.

Location-based Authentication with Wireless-LANs

By

Eric Daniel Thomas
B.S. (San Jose State University), 2002
A.S. (Las Positas Community College), 1999

THESIS

Submitted in partial satisfaction of the requirements for the degree of

Masters of Science

in

Computer Science

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

Davis

Approved:

_____

_____

_____

Committee in Charge

2005

Location-based Authentication with Wireless-LANs

by

Eric Daniel Thomas

Abstract

Location-based Authentication with Wireless-LANs

by

Eric Daniel Thomas

For some time, password-based authentication systems have become one of the major vectors of attack because password guessing and brute-force attacks are often very successful. Multiple-factor authentication has promised to supplement these systems by making it more difficult for an attacker to obtain the required credentials for successful authentication. Other authentication factors, such as the use of tokens ("what you have") or biometrics ("what you are"), are often expensive and/or highly subject to attack. A fourth factor, location-based authentication, can provide an inexpensive and high integrity supplement to other authentication methods, and can be implemented using already widely-deployed 802.11a/b/g WLAN (or Wi-Fi) networks. This thesis discusses the deployment of a location-based authentication system that uses the wireless infrastructure to provide location information. It focuses on location determination techniques, a model of location-based authentication, strategies and policies for dealing with location estimation system error, a realistic deployment of the system, attacks against the system, and possible defenses against these attacks. This thesis shows that inexpensive and high-integrity location-based authentication is a practical solution to authentication issues.

Professor Prasant Mohapatra
Dissertation Committee Chair

To my wife, who put her plans on hold so I could reach my goals.

To my mother, father, and Bonnie, who taught me the importance of self-discipline and determination.

Contents

List of Figures

Acknowledgements

I'd like to say thank you to the many people who have helped me achieve my educational goals. Thanks goes to Prasant Mohapatra, who is the Committee Chairperson for the approval of my thesis. Thanks to Matthew Bishop and Felix Wu, who were the other committee members, and also helped inspire some of the ideas for the thesis, even though the probably didn't know it. I thank Barry Hess and John Howard, who were my managers at Sandia and allowed me to spend a significant portion of my time on school-work. Barry practically made my getting my Masters degree a requirement for employment, and as such I was very happy that he was sensitive to my ambitions to get the degree. John Howard helped me get on track with my thesis a few times when I was a little lost, and he kept me focused. He, Jamie Van Randwyk, Tim Toole, and Nancy Durgin, all workmates of mine, helped me in one way or another to refine ideas and review my thesis. I also thank Sandia National Laboratories in general, whose University Part Time program funded my graduate level education.

Special thanks goes to Nina Berry. She was a mentor of mine for some time as an intern, and I learned much from her. Also, she ultimately convinced me to pursue a higher level degree beyond my Bachelors, and I'm very happy that she did.

Special thanks also goes to Fred Cohen. He sparked my interest in Computer Security early in my college education. My entire career was started down its current path because of his vision of creating a group of interns studying computer and network security at Sandia. Over the couple years I worked with him, I learned more from him than I have from any other single person (except my parents, of course). He emphasized both critical and creative independent thinking in solving computer security problems, which was just what I needed to get started in, and to be able to contribute to this field.

Chapter 1

Introduction

Defense in depth has become a new buzzword in the field of Computer Security. The idea behind it is the provision of multiple layers of defense protecting computers and networks. Technology like firewalls, proxies, intrusion detection systems, virus scanning programs, and passwords as well as actions like patching and program updating are all some of many different techniques that, when used together, provide a fairly solid method of defense against malicious behavior.

Even with these defenses in place, systems are still being attacked and exploited. Though there are many vectors of attack, one that occurs time and time again is the guessing of passwords. Even corporations with policies that require strong passwords[1] may have problems enforcing those policies. People usually have the freedom to choose their passwords because they have a high level of control over their computing systems. Studies have shown that users tend to choose passwords that are easy to remember [1, pp 316-317]. Unfortunately easy-to-remember passwords are also usually easy to guess.

Sometimes weak passwords are the default for systems, and the operators simply have not changed them. For example, the Linksys Wireless Access Point default password is *admin* [2]. Many consumers will buy the access point and deploy it without any configuration changes. This is easily recognizable to people who wish to exploit this

[1]Strong passwords are passwords that are not easily guessable or susceptible to brute-force guessing attacks [1, pp 316-317]

weak password because the default SSID[2] is *linksys* and WEP[3] is not enabled, both of which are easy to see. This increases the likelihood that the operator did not make any configuration changes, and that the default password is highly likely to succeed.

Also consider the malicious Internet bot[4] that came out in late January, 2005 that takes advantage of the *MySQL UDF Dynamic Library Exploit* [5, 6]. The bot first gains access to the MySQL system root account by brute-force guessing for weak or null passwords. After access is obtained, the bot uses MySQL to infect the system and propagate to other systems. This bot spread to surprisingly many systems (approaching 10,000 systems within 24 hours of detection) using the password guessing technique, even though, at the time, there was increasing awareness of the need for secured services [7].

These password problems are not likely to go away soon because, for many people, the services provided are far more important than the security necessary to protect the services. Such access control problems could be solved if proper authentication is used. Informally, authentication is a process that verifies that a user is who he says he is using some form of credentials. If a system can be sure that the person accessing a service is an authorized user, then the system is more secure.

Typically, authentication is based upon "what you know" (e.g. a password), "what you have" (e.g. a token), or "what you are" (e.g. a unique fingerprint). Combining two or more of these factors is what makes a multiple-factor authentication system. This is one method that could make authentication systems stronger by requiring that the entity attempting to gain access satisfy more conditions than simply knowing a password.

Unfortunately, most current systems have significant disadvantages. The high cost of token-based or biometric sensing equipment makes multiple-factor authentication a problem for personal use, small business use, or very large deployments. Furthermore, these systems need to be managed just as much, if not more, than password systems,

[2]Service Set Identity. This is often a user-recognizable character string that gives a name to the network [3].

[3]Wired Equivalent Privacy. Built-in encryption designed to provide minimal privacy and access control on 802.11 networks [3].

[4]A bot is a software program that interacts with network services intended for people as if it were a real person [4].

costing more man-hours.

Using location, i.e. "where you are", as another factor for authentication may have some advantages over traditional multiple-factor authentication systems. A system that determines the location of an entity (e.g. user, device, etc.) is called a *location determination system*. If the location determination system is deployed in a wireless network, it may be part of the normal network service infrastructure, or at worst, part of the security infrastructure (e.g. intrusion detection system (IDS) sensors, etc.); it does not require any additional hardware. Also, if this authentication system is designed correctly it may be highly resistant to replay, spoofing, and other common forms of attacks to which the other forms of authentication are susceptible. Finally, it may easily be integrated into traditional authentication systems.

As a result, using location as a means of authentication may be a logical solution for multiple-factor authentication. For most people, day-to-day computer and network operations are likely to originate from a small set of locations, like a specific office in a building, or a specific desk in a room. Simply requiring that an entity attempt to log in from a small set of locations not only greatly reduces the possibility of successful authentication by an unauthorized entity but also presents a small inconvenience to the authorized entity.

1.1 Thesis Contributions

The main contribution of this thesis is the introduction and discussion of the use of 802.11 wireless LAN location determination for the second factor of a two-factor authentication system, the other factor being a password ("what you know"). Other contributions include formalizing a model describing location-based authentication (Chapter 3), defining policies for dealing with systematic error in current wireless location determination systems (Chapter 4), discussing attacks against wireless location-based authentication systems, and specifying the use of location-based authentication in real-world applications using Pluggable Authentication Modules (Chapter 5).

1.2 Thesis Organization

This thesis gives some background into location determination techniques, authentication theory, and previous attempts to merge the two in Chapter 2. Chapter 3 discusses the location-based authentication model and what an ideal system would entail. Knowing that by nature an ideal system is not possible, Chapter 4 discusses using policy to help overcome the system errors inherent in location determination systems. Chapter 5 discusses how a location-based authentication system can actually be deployed, some attacks that are possible against the system, and what defensive strategies can be applied. Finally, Chapter 6 discusses future potential areas of research and development for wireless location-based authentication.

Chapter 2

Background

In this chapter we discuss several different techniques for determining the location of an entity. We also state a commonly referenced model of authentication. Finally, we discuss previous research and commercial projects that attempt to bind location determination and authentication.

2.1 Location Determination

Determining the location of a entity can be accomplished using many well understood methods. In the animal kingdom dolphins and bats locate prey using the reflection of sound waves off of objects, a technique called *sonar*. The closer an object is to the transmitter/receiver, the shorter the round trip time of the signal and its echo. Such techniques are distance limited both by the speed of sound and the rate of signal degradation as the signal travels farther.

Other similar techniques use radio waves, such as aircraft radar and radar guns used by police officers [8]. The propagation speed of the signals, approaching the speed of light in a vacuum, allows these systems to locate objects that are much farther away.

In a wilderness environment, experienced people can self-locate using a compass, a topographic map, and unobstructed views of visual references, such as mountain tops, ridges, buildings, etc. The process, called *orienteering*, involves finding multiple bearing

lines leading toward these references on a map, the intersection of which is the person's location [9].

Though there are many other techniques for determining the location of an entity (e.g. using infra-red, ultrasound, and so on) the remainder of this section focuses on location determination techniques that utilize various characteristics of radio signals where both the location determination infrastructure and the entity act as both transmitters and receivers.

2.1.1 Non-WLAN-based Location Determination

A significant amount of technical work has been geared toward determining the location of an entity utilizing radio frequency technology. For example, the Global Positioning System (GPS) developed by the military uses dozens of earth-orbiting satellites, each of which sends radio-frequency (RF) messages to the surface [10]. Receivers on the surface temporally synchronize with the received signal to find a propagation delay, which is converted to the distance between the satellite and the receiver. Obtaining and synchronizing signals from four or more satellites allows a receiver to triangulate its location to within 100 meters (95th percentile), though accuracy is usually much better, especially when using additional sources of information like differential-GPS (DGPS, 5-10 meters) [11] and Wide Area Augmentation System (WAAS)-enabled [12] satellites (less than 3 meters). Unfortunately, the use of GPS is usually limited to outdoor environments because signal attenuation through building materials reduces the strength of the received signal so that it is indistinguishable from background noise.

An alternative location determination method uses the existing cellular phone infrastructure. The *Global System for Mobile Communication* (GSM) technology is currently the most popular form of cellular phone infrastructure world-wide, which in the USA usually operates in the 1900 Mhz frequency range [13]. A cell phone communicates with the rest of the global telephone network via messages to and from a nearby base station. All base stations continually monitor the radio frequency signal strengths of cell phones within range. Base stations will communicate with one another to determine

who is obtaining the greatest signal strength from a cell phone, commanding the cell phone to start communicating with a different base station (on a different frequency band) when the other base station is observing a better signal [14]. This "hand-off" technique indicates the assumption of a functional relationship between signal strength and proximity.

This assumption corresponds to the free space attenuation principle of RF signals, as expressed in Equation (2.1), where r is the travel distance of the signal of wavelength λ, and L_{fsl} is the free-space signal loss [15].

$$L_{fsl} = \frac{r^2(4\pi)^2}{\lambda^2} \tag{2.1}$$

When Equation (2.1) is expressed in decibels (dB) and we assume the standard frequency of the popular frequency of GSM networks (1900 Mhz), the equation simplifies to Equation (2.2).

$$L_{fsl} = 38 + 20 * log(r) \tag{2.2}$$

Appendix A describes how this signal-loss equation is obtained, as well as the similar equation for 802.11 networks operating in the 2.4 Ghz frequency band.

Equation (2.2) indicates that as the distance r between the RF transmitter and receiver increases, the loss due to free space signal attenuation increases logarithmically. Thus, if one considers a client station C (e.g. a cell phone) and two base stations $S1$ and $S2$ (e.g. cell towers) as shown in figure 2.1, with distance $a > b$, one would expect that the signal strengths of client messages observed from base station $S2$ would be greater than those of $S1$. Hence C will send messages to $S2$ under the premise that the free space attenuation principle indicates that $S2$ is closer.

This mechanism provides a very coarse estimate of the client station's location. It narrows down the possible locations to the area covered by the base station[1] through which the client is communicating.

[1] In cellular networks, the area covered by a base station is called a cell, and is usually hexagonal in shape [16].

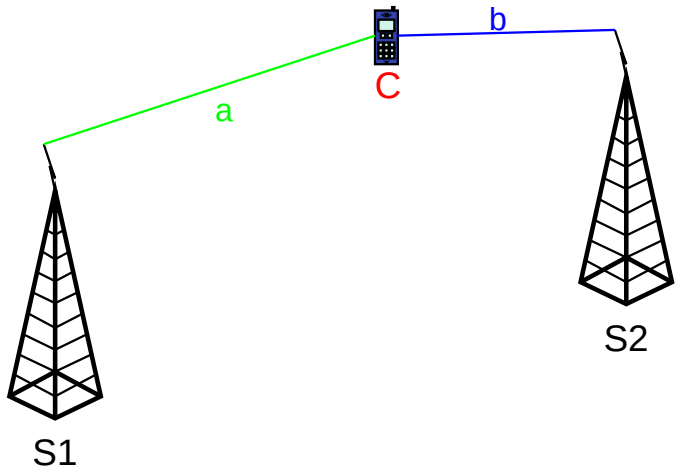


Figure 2.1. **Signal path loss increases with distance**. Client C is closer to tower S2 than tower S1, therefore the signal path loss between the C and S2 would be less than that between C and S1. In that case, C should send messages to tower S2.

This area could possibly cover tens of square kilometers [17], and for some applications better accuracy is required. For example, the Federal Communications Commission (FCC) imposed a requirement that all cellular network providers have the capability of determining the location of a cell phone to within 125 meters (67th percentile) by October 2001 [17]. The motivation behind such a system, called E-911, is to allow emergency services to locate a cellular user who needs help, but who is unable to talk with an operator to provide his or her location. In this application, narrowing down the location to within a few square kilometers in not sufficient, so the requirement imposes stricter guidelines.

More advanced techniques were developed to provide more accurate location claims[2], which take advantage of different characteristics of RF signal propagation[3]. One technique uses what is referred to as the Angle-of-Arrival (AOA) technique, which takes advantage of the fact that RF signals travel along a relatively straight path (assum-

[2]A *location claim* is an estimate of an entity's location produced by a location determination system, regardless of the method used to derive the location estimate.

[3]For the remainder of this section and much of the remainder of this thesis, we consider only computations and location claims in two dimensions for simplicity. The discussion does apply in three dimensions. Furthermore, the curvature of the Earth's surface is ignored, though it is obviously important in a real-life situation.

ing there is little interference from physical objects). Two or more base stations could measure the relative angles from which the signals of a client station are received. Correlating these measurements could be used for a central location determination system to locate the client station by computing the intersection of the bearing lines to the client, a process called triangulation.

Two other techniques can take advantage of the fact that RF signals travel at predictable speeds through standard atmospheric conditions, speeds approaching the speed of light in a vacuum ($3.0 * 10^8 \frac{m}{sec}$). These techniques are call Time-of-Arrival (TOA) and Time-Difference-of-Arrival (TDOA)[17]. The TOA technique measures the one-way time it takes for a signal to propagate from the client to the base station, usually by timing the cumulative round trip time of a request-reply exchange[4]. Multiple base stations could perform this measurement (relying heavily upon client cooperation), and use the speed of signal propagation to estimate the distances from the client to each base station. These distances relate to circles[5] centered at the base station with radii being the distance from that base station to the client [17, pp 8-9]. The intersection of the circles is the estimated location of the client. This system is very similar to the system employed by GPS, though in GPS it is the client that does all the computational work [10].

In the TDOA technique, multiple time-synchronized base stations report the time in which a single specific message was received from the client at each station. A base station that is closer to the mobile client will report having received the message earlier than a base station that is farther away, as a result of propagation delay. A central processor can compute the time difference between the observation times of two base stations, converting the difference to a distance by assuming the signal travels at a constant propagation speed. This distance is assumed to be the difference in distance between the client and one base station and between the client and the other base station.

[4]The cumulative round trip time includes the transmission times of the messages, the propagation delays, and the processing delays, all of which the base station must account for.

[5]Many location determination systems locate in two dimensions, hence the assumption of a two dimensional circle as opposed to a three-dimensional sphere.

The plot of all points (in Cartesian coordinates) that satisfy this difference results in a half-hyperbola[6] [17, pp 13-15]. If the central processor performs this computation for multiple base station pairs, multiple half-hyperbolas can be plotted, the common intersection of which is assumed to be the location of the client (see figure 2.2).

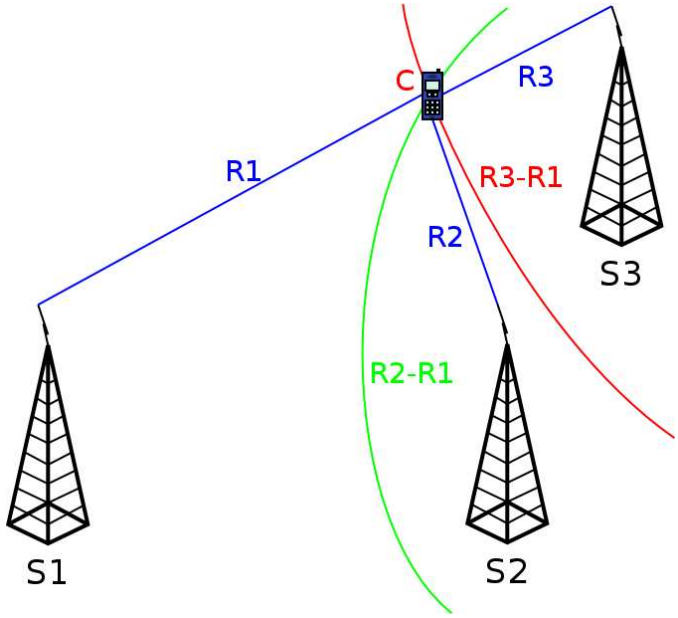


Figure 2.2. **TDOA technique for location determination**. R2-R1 is the hyperbola formed as the constant distance difference between the client and each of S1 and S2. R3-R1 is similarly defined. The intersection of the hyperbolas, C, is the location of the client.

All the aforementioned techniques and infrastructures are susceptible to large errors. The distances between the location infrastructure and client can range from a few kilometers (in the case of cellular networks) to tens of thousands of kilometers (in the case of GPS), and the farther a signal has to travel, the more the opportunity for the signal to be affected by environmental conditions. Radio Frequency (RF) signals are susceptible to attenuation, reflection, interference, and multi-path fading, and thus errors are introduced as signals traverse the environment. As previously mentioned, GPS is only available outdoors. The other techniques will work in indoor environments but are unlikely to provide enough accuracy to meet the needs of certain applications, for

[6]The actual shape is a hyperbola, but half of it can be eliminated because it is known which base station is closer to the client

example authentication (described in section 2.2), which may require accuracy to within a few meters.[7].

2.1.2 WLAN-based Location Determination

The growth in the number of recent research papers indicates there is interest in using 802.11 wireless network infrastructures for a means of location determination. Wireless-LAN (WLAN) infrastructures can be cheap to deploy and are in relatively close proximity (within 100 meters) to the entities which are to be localized. They are also usually deployed in indoor environments to accommodate the audience of the technology, usually users with laptops and other mobile devices. Though RF signals in indoor environments are susceptible to the same RF signal problems described above, and in some circumstances more-so[8], engineers have been able to overcome the many challenges and routinely obtain location accuracy measures to within five meters[9].

Though differential or WAAS-enabled GPS is able to achieve this level of accuracy outdoors, WLAN location determination systems can be deployed anywhere where power and communication lines can be run to the access points, may not require specialized and expensive hardware, and may use the already-deployed wireless infrastructure without interfering with normal operation[10]. These characteristics may make WLAN-based location determination attractive to commercial and government organizations.

Many research papers in WLAN location determination have involved the measurement of signal strength of RF signals in an 802.11b (Wi-Fi) deployment. The first of these was Bahl et al. concerning a system called RADAR [18]. The RADAR system consisted of two phases. The first is the *off-line phase*, during which a model of the signal strength is obtained. The authors took a laptop equipped with a wireless network

[7]Location determination in cellular networks for E-911 purposes has a maximum error of 125 meters (67th percentile) [17], and GPS can have an error of as much as 100 meters [10]

[8]Indoor environments have many walls, doors, equipment, and people, all of which affect RF signal characteristics in a somewhat unpredictable manner.

[9]The five meters measure is the author's estimate based upon results from cited literature [18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32].

[10]A fairly cheap, easily deployable, highly programmable, and customizable access point may be created using commodity hardware (e.g. Soekris devices [33]), an open source operating system (e.g. Pebble Linux [34] and LinuxAP [35]), and access point software (e.g. HostAP [36]).

card to numerous locations in a hallway, recorded the signal strengths of messages from the client as observed from multiple access points (APs), and created a lookup table. The lookup table consisted of a mapping of signal strength vectors to physical locations. The granularity of the model was defined as the distance between physical locations from which measurements were taken. The higher the granularity, the smaller the space between points.

During the *online phase* a signal strength vector was created by combining the signal strengths received for a message from each of the access points in range, and a lookup in the table was performed. RADAR computed a *distance metric*, which was the Euclidean "distance" (root-mean-square) between two signal strength vectors. The two vectors used in estimating the client's location were the observed signal strength measures obtained in the online phase, and a signal strength vector stored in the lookup table. The location associated with the lookup table entry with the smallest distance metric was assumed to be the location of the client.

Subsequent location determination algorithms follow a similar structure ([19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31]). All of these systems consist of an offline phase, during which a model of the signal space is created, and an online phase, during which location claims are computed from the model. Obviously, the more accurate the model, the more accurate the location claims would be.

Based upon the author's survey of the cited literature, some characteristics of WLAN-based location determination have emerged. There are at least two orthogonal axes that can be used to classify different location determination algorithms (refer to Figure 2.3). The two axes are 1) the active party during the online phase, and 2) the principal algorithm employed by the location determination algorithm.

On one side of the active party axis (depicted on the vertical axis in Figure 2.3) are systems where the client is attempting to locate itself based upon signals received from the APs. The APs (and the rest of the WLAN infrastructure) are possibly unaware of the localizing attempt[11]. On the other side, the infrastructure is performing the location

[11]Though it is necessary for the client to obtain access to the signal model, which may need to be

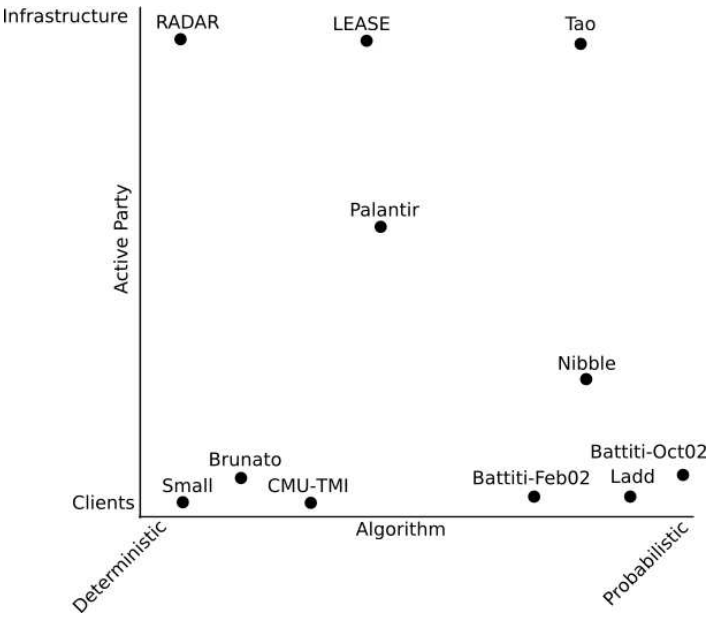


Figure 2.3. **Location Determination System Classification**. This graph depicts the classification of radio map-based location determination systems, including a general idea of how the following specific systems are classified: RADAR [18], Small [37], CMU-TMI [20], Nibble [21], Battiti-Feb02 [22], Battiti-Oct02 [23], Brunato [38], Ladd [30], Tao [25], Palantir [31], LEASE [32].

determination without the knowledge of the client. It is also possible to have a system that involves both an active client and an active infrastructure, which indicates that the active party axis is a continuum.

A comparison of research paper results indicates that client-based location determinations systems (only the client is active during the online phase) tend to be more accurate[12]. A major contributing factor for this is that access points are often very static devices, both in terms of physical location as well as signal space properties. An access point (AP) is unlikely to have different output power levels, which directly affects the signal strength model. A further reason for the preference of client-based systems is that privacy advocates agree with this approach because the client is able to keep its location to itself without having to involve the infrastructure [20].

However, client-based systems can have some major drawbacks, especially when

obtained from a server on the infrastructure.

[12]This is not a rigorous comparison as environmental conditions affect the accuracy, not just location algorithm.

it comes to network security. It may be in an organization's interests to be aware of the physical location of a device that is using its network. The infrastructure should not have to rely on the client to accurately and truthfully report its location because, with respect to security, the client cannot be assumed to be trusted. Even if the client is trustworthy, the problem of heterogeneous hardware still exists. One can expect to find a wide variety of wireless network interface cards (NICs), each of which is likely to have varying power output levels and different received signal strength measurement errors.

Infrastructure-based techniques try to resolve this by assuming the client is bad, and therefore do not involve it in the location determination process. Still, there are various techniques that rogue clients[13] can use to try to confuse infrastructure-based location determination systems, such as varying output power levels on their NICs. This is likely to significantly degrade the accuracy of most location determination systems. Some systems try to account for devious behavior and heterogeneous hardware, but they tend to be less accurate ([25, 31, 32]).

Youssef et al. was the first to classify radio map-based location determination research into categories based upon the "technique" used by the location algorithm [39]. Early research used *deterministic techniques* that modeled the signal strength space as a lookup table or some similar mechanism ([18, 19, 37, 38, 24, 25]). Nearest neighbor, k-nearest neighbor, and m-vertex polygon algorithms are considered to be deterministic algorithms. The other category was *probabilistic techniques* which employ the use of a trained probabilistic or statistical model of the signal strength space ([21, 22, 23, 30, 26, 27]). Neural networks, bayesian (belief) networks, support vector machines, and statistical models are examples of probabilistic algorithms.

Though Youssef et al. categorized algorithms discretely into one of these two broad categories, it is unclear whether this is a proper classification. It may be more accurate to describe the algorithm on a continuum. For example, whereas a nearest

[13]Rogue clients are network clients that have malicious intentions. The actual intentions of the legitimate user are unimportant, as he or she may have accidentally downloaded and installed malicious software, or the client computer may have been compromised and be under the control of someone with malicious intent.

neighbor technique might be considered a deterministic technique, a k-nearest neighbor technique, which includes some elements of statistics, might not fall neatly into the deterministic category. This algorithm continuum is our second axis (depicted on the horizontal axis of Figure 2.3) of location determination system classification.

Figure 2.3 depicts the two axes classification, active party and algorithm, as well as where some previous location determination systems may fit within the classification. A more rigorous taxonomy might be an interesting topic for future research.

There are many interesting and useful results from the many papers devoted to WLAN location determination. Examples are as follows:

- Orientation of the antenna affects received signal strength, especially when a human is present and may be a source of attenuation [18].
- Current parametric signal propagation models cannot model the indoor signal strength environment to a high degree of accuracy [18, 20].
- Continuous tracking of users helps to alleviate aliasing problems[14] [19].
- Environmental conditions (e.g. temperature, humidity, etc.) change at different times of the day, and these changes affect the accuracy of the signal map [19].
- Over time the physical environment (e.g. furniture, equipment, etc.) tends to change, affecting the signal space in an unpredictable manner [21, 32].
- Probabilistic methods tend to be more computationally intensive [23].
- Increasing the granularity by a factor of two resulted in an order of magnitude more computation time, though the increase in accuracy was marginal (13%) [24].
- Mixing granularity by increasing granularity in more hard-to-model areas improved performance more than marginally [24].

[14]The aliasing problem is when more than one location claim, separated by relatively large distances, are likely estimates given by the location determination system.

- Modeling the difference in signal strength provides better accuracy and resilience in the face of rogue clients than measuring signal strength itself [25].
- Clustering techniques (where each cluster is an independent model) can significantly reduce computation required without sacrificing much accuracy [26].
- Creating a signal model by using sophisticated interpolation techniques can significantly reduce the number of training points required without sacrificing much accuracy [27, 32].
- Theoretically ideal probabilistic techniques are provably more accurate than theoretically ideal deterministic techniques [39].
- The use of a trained neural network to fuse the location claims from multiple location determination algorithms results in more accurate location claims [28].
- The offline phase can be eliminated as long as there are dedicated tamper-resistent stationary emitters that are used to build an interpolated signal map [32].
- The signal model can be constantly updated during the online phase, eliminating errors caused long-term environmental changes [32].

The study of location determination is a large project unto itself. Many works have been omitted for brevity's sake. The results of the cited works above indicate that it is indeed practical to determine the location of an 802.11b wireless device to within a few meters accuracy using signal strength characteristics of RF communications. There are many other avenues of research that have been and are currently being explored, for example, involving location determination within ad-hoc wireless networks (e.g. [40, 41, 42]). Certainly there are current projects seeking to improve upon the work that has already been done. For example, Youssef et al. continues to explore probabilistic location determination techniques [43].

It is also important to note that location determination has some exposure in the commercial arena. Three products, WiFiWorkplace, Ekahau Positioning Engine,

and AeroScout ([44, 45, 46])[15] all use 802.11b signal characteristics to determine the location of entities on the wireless network. WiFiWorkplace [44] and AeroScout [46] use the TDOA technique, and therefore require specialized hardware sensors that can synchronize on the scale of nanoseconds[16]. Ekahau uses a signal map based upon a manual survey [45]. These commercial products emphasize both inventory/personnel tracking and security applications. The success of these products indicate that location determination within WLANs has a positive future outlook for commercial and government institutions.

2.2 Authentication

Determining the location of the entity is only a small part of the location-based authentication process. This section describes a model that defines authentication in general [1], which will later be used to describe location-based authentication with wireless-LANs.

Key definitions:

- **principal** - a unique entity
- **identity** - specifies a principal (e.g. username)
- **authentication** - a process that binds an identity to a principal

In the authentication process, authentication information is obtained from an entity, is analyzed, and is then used to determine if that information is associated with the entity. The following describes the model for an authentication system[17]:

1. The set A of *authentication information* is the set of specific information with which entities prove their identities.

[15] These three products, are trademarked by Newbury Networks, Ekahau, and AeroScout respectively, and are the three initial non-RFID-based products that the author is aware of that do location determination based exclusively upon 802.11b signals.

[16] An RF signal traveling at the speed of light will travel 0.3 meters in 1 nanosecond, and 3 meters in 10 nanoseconds. Thus for good location accuracy, clock precision and synchronization on the order of tens of nanoseconds is required.

[17] This definition is obtained directly from *Computer Security, Art and Science* by Matthew Bishop [1]

2. The set C of *complementary information* is the set of information that the system stores and uses to validate the authentication information.

3. The set F of *complementary functions* that generate the complementary information from the authentication information. That is, for $f \in F$, $f : A \rightarrow C$.

4. The set L of *authentication functions* that verify identity. That is, for $l \in L$, $l : A \times C \rightarrow \{$ **true, false** $\}$.

5. The set S of *selection functions* that enable an entity to create or alter the authentication and complementary information.

An instance of an **authentication system** is described by the 5-tuple $\Pi = \{A, C, F, L, S\}$.

In the security community it is often believed that there are three main factors of authentication. That is, an element a of the set A can be classified into one or more of the following classifications [1, pp 309]:

- What the entity knows (e.g. password)
- What the entity has (e.g. badge)
- What the entity is (e.g. fingerprint)

A fourth factor has also become relevant: *Where the entity is* (e.g. an office or building) [1, 47]. This fourth factor is the basis for any location-based authentication system. The justification for using location as authentication information is that many locations have physical limitations for public access, with the exception of a few individuals. If an entity is at a location with restricted access, and the physical limitations have not been compromised (a standard assumption), then the entity is assumed to be one of the few individuals with access to that location. In this case, using the location as a factor for authentication is justified, though not sufficient alone, for binding an identity to the entity.

2.3 Related Work

Much of the motivation for this thesis comes from Denning and MacDoran's paper, *Location-Based Authentication: Grounding Cyberspace for Better Security* [47]. This paper was the first public paper to describe using location as a factor for authentication. It described the benefits to security and how such a mechanism could be implemented.

Their system used the Global Positioning System (GPS) to obtain a geographical location signature which was forwarded to a verifier. Access to systems or information was only allowed if the location signature met the criteria set up *a priori*. A main concept introduced was that of an *unforgeable location signature*. The *unforgeable* qualifier is important. A signature that can be spoofed or replayed undermines the authentication system by allowing an entity to say it is in a location that it is not. Denning and MacDoran specifically stated that generic GPS is not suitable for authentication because a false location signature can be forged. But with certain technical additions to the system a high degree of unforgeability is achieved [47].

Though at least one commercial product was derived from this system, the idea did not progress far, partially due to unreasonable expenses (a GPS receiver was much more expensive then), and partially because of the limitations of GPS, namely the inability to work in indoor environments. Since much work that requires access controls is performed indoors, (e.g. offices, conferences rooms, etc.) GPS is not a viable solution. Certainly GPS can be used to locate a person to a particular building, but higher granularity is often required. Presence of the entity in a building is not sufficient to bind an identity to the entity, because it is often the case than many people have access to the building.

Other location-based authentication systems pick up where GPS leaves off. By verifying that a device is in the right location, which may be an indoor location, that device is assumed to be authorized to perform a certain task or use a certain service. Gabber et al. described four mechanisms for verifying that a *set top terminal* (STT; e.g. a satellite television receiver) is in the location it is supposed to be [48]. The

problem encountered was that people would sell STTs on the black market, resulting in the provision of satellite television to non-subscribers. The system used at the time required the STT to dial home via telephone land-lines and a caller ID function. If the number used by the STT matched the number on record, the base station would send the STT a decryption key that allowed the STT to continue obtaining satellite data. The main assumption of this system was that the STT would be physically located at the location associated with the source phone number, which was assumed not to be transferable to another location. Unfortunately, this system is highly susceptible to a proxy attack, where another system relays the communication between the STT and the base station from the appropriate location, even though the STT could be located anywhere.

The second STT location mechanism employed the use of GPS to create a location signature. This can defeated by a man-in-the-middle attack between the tamper-resistent secure module (which communicates with the base station) and the GPS unit, or more simply, by faking signals from the GPS satellites in order to make the GPS unit give the wrong, but expected answer.

The third mechanism used the cellular network's E-911 location determination system. This can be attacked by using a proxy system where the main cellular communication antenna resides in the correct location, and the proxy relays data between the cellular network and the STT.

The fourth mechanism, and the one which Gabber et al. preferred, used signals sent from satellites and a synchronized clock. Satellites would transmit an unpredictable non-repeating signal. The base station would tell the STT to start and stop recording the signal at a specific time (hence the synchronized clock). When the complete signal was received, the STT would relay the received signal to the base station. The mechanism relied upon the concept of phase-shift. If two STTs in different locations started recording at the exact same time, they would observe a different signal because it takes longer for the signal to reach the STT that was farther away. The base station would be able to compute what signal should have been received at a specific location for any time

duration. By comparing the received signal given by the STT to the computed signal, the base station could verify that the STT was at a specific distance from the satellite. By correlating the information from multiple satellites, the location could be even more precisely verified. Assuming the satellite antenna was tamper-resistant, this system would be quite difficult for an attacker to circumvent.

Kindberg et al. used a concept called *constrained channels* to verify that an entity is in a specific context[18] [49]. In a constrained channel, communication is only allowed to take place if a contextual predicate is true. The authors give an example using a WLAN – there is a receive constrained channel between an access point and an associated client, where the contextual predicate is *the receiver is in close proximity to the access point*. If the client is associated with the access point, then it must be somewhere near the access point (say, within 100 meters), otherwise it would be too far away to receive signals[19].

The paper then goes on to describe some protocols that use constrained channels. In their scenarios, a client is attempting to access a server, and the server only allows access if it is sure the client is in the specific context (location). The server employs the use of a trusted proxy, which shares the constrained channel with the client. If the server can communicate to the client through the proxy, then the constrained channel must be in place, and so the client must be in the appropriate context. Using symmetric and asymmetric key cryptography and nonces, the protocols are able to thwart many common attacks, like man-in-the-middle attacks, spoofing attacks, as well as loss of privacy (when the client wants to keep his location private).

There are issues with the use of constrained channels. First, it cannot always be assumed that there is a trusted proxy that can verify the context of the client. Consider the case where an employee is working from home and the company has given the employee a proxy system to use. The employee may be able to tamper with the proxy so that he can maintain the "constrained channel" even when contextual predicate is false

[18] Location is a specific type of context. Other contexts include temperature, orientation, mood, time, etc.

[19] Assuming that no high-gain antennae or amplification techniques are used.

(e.g. he is not at home). Second, creating a constrained channel might be very challenging. What if the contextual predicate needs to be more precisely defined to be *the client is in the same room as the access point*. Now associating with the access point is not enough. The client location has to be determined by the proxy before the constrained channel can be valid, and the mechanism for doing this might not be available.

Sastry et al. provide another solution that uses both RF and ultrasound transmitters and receivers to estimate a client's location [50]. A verifier system V will verify location claims that reside within a circular region R surrounding V. A client makes a location claim l to V. If the location claim is outside of R, then the claim can be rejected immediately. But if the claim is inside region R, then the verifier uses the *echo protocol* to verify the location, as follows. The verifier sends a request to the client over the RF channel, and the client sends the response over the sound channel. V measures the round trip time of this communication. Since the speed RF and sound signals travel is predictable (and processing delay is also considered), V can determine whether the client is within the particular distance as specified in l.

This design is susceptible to a relay attack where a relaying device is placed at the claimed location, and the actual client is somewhere else, with some high propagation speed link between the two. This works because sound signals travel much slower than signals on the back-channel communication link. Another problem is that the client and verifier devices both need to be equipped with both RF and ultrasound communication devices. Finally, if the location needs to be more precisely determined (so that it is an actual location, not a distance from the verifier that is checked), this system will not work.

Waters and Felton designed a system that was not susceptible to the relay attack described above because the system used the round-trip time of RF signal exclusively [51]. Since RF signals already travel at the fastest possible speed, any delay caused by relaying communication to another location will be significant, and will be caught by the verification system. Furthermore, the authors described the potential for physically locating the system beyond simply determining the distance by using TOA

techniques with multiple verifiers. However, the paper discusses the protocol and techniques only, and does not attempt an experimental setup. This is likely due to the fact that hardware that could measure round-trip latency at the desired precision is difficult to find and prohibitively expensive.

Commercial products claim to provide WLAN location-based authentication services, but more often than not, these claims are imprecise and filled with marketing hype and buzzwords. WiFi Watchdog [44] creates a logical perimeter, outside of which clients cannot obtain access to the wireless network. Newbury Networks [44] claims their comparable service is location-based authentication, when in fact this coincides much better with location-based access control - the service provides no binding of an identity to a principal.

A WLAN location-based authentication service would try to help bind an identity to an entity only if the entity is in one of the locations allowed. The current location of the entity would be determined through a location determination system configured with a wireless network (Section 2.1.2). Though a trustable location signature is still necessary, it might be enough to simply trust the location determination system to provide a good location claim.

2.4 Summary

In this chapter we have looked at location determination using several methods and infrastructures. We have discussed location determination using TOA and TDOA techniques with a cellular network infrastructure. We then looked at location determination using Wi-Fi infrastructures, showing that fairly accurate location claims were possible. We reviewed a theoretical model for describing authentication, then discussed previous work that looked at using location of devices as a factor in authentication.

Chapter 3

An Ideal Location-Based Authentication System

In this chapter we define a formal model of location-based authentication. Then we examine a location-based authentication system that is ideal, meaning that the system has no false positives and no false negatives.

3.1 Location-based Authentication Model

Location-based authentication can be described using the authentication model discussed in Section 2.2. The following model assumes that the client supplies a username u, and the location determination system provides a location w. The generic location-based authentication system is described as $\Pi = \{A, C, F, L, S\}$, where

1. $A : \forall a \in A$, $a = \{u, w\}$, where A is the set of authentication information, u is the claimed username, and w is a location supplied by the location determination system.

2. $C : \forall c_u \in C$, where C is the set of complementary information, and c_u is a set of locations. Each set is associated with a username u, and is used for authentication of an entity claiming to have the identity u.

3. $F : F = \{f\}$, where F is the set of complementary functions, and $f(a) = a$ (the identity function).

4. $L : L = \{l\}$, where L is the set of authentication functions, $l(a, c_u) \equiv w \in c_u$, and $a = \{u, w\}$.

5. S : where S is the set of selection functions by which the entity changes c_u through an authorized program or an administrator.

This model describes a generic location-based authentication model that binds an identity (username) to an entity using only location authentication information and a username[1]. The advantage of this model is that it does not specify exactly how the location is determined or how the complementary set (sets of authenticating locations) is specified. Thus the model provides for some flexibility in the implementation.

3.2 Ideal Authentication System

There are a few definitions that allow us to discuss location-based authentication:

authorized area (AA) - the set of locations among where an entity should be for successful authentication (i.e. complementary information).

false positive - successful authentication when the entity's location is not within the AA.

false negative - failed authentication when the entity's location is within the AA.

In the authentication model, the authorized area is what the location claim w is checked against. Therefore, for an entity u, the complementary information $c_u = AA_u$, which is the set of locations where authentication should succeed.

In an ideal location-based authentication system, an entity is always correctly authenticated when within the predefined authorized area and correctly not authenti-

[1]It is simple to construct a model that uses passwords in addition to location and username, the implementation of which would provide a two-factor authentication system.

cated when outside the predefined authorized area.[2] The system is ideal as long as the following two premises hold true:

Premise 1: The location w presented to the authentication system is always accurate and correct down to the level of granularity of the location determination system.

Premise 2: The granularity of the location determination system is such that it allows the authentication system to correctly determine whether a location resides inside or outside the AA.

A location determination system that incorrectly claims the true location of the client may report that the client is inside the authorized area (AA) when it is outside, or outside the AA when it is inside (see figures 3.1,3.2). So in order for the authentication system to have no false positives or false negatives, it must be the case that the location determination system correctly claims the true location.

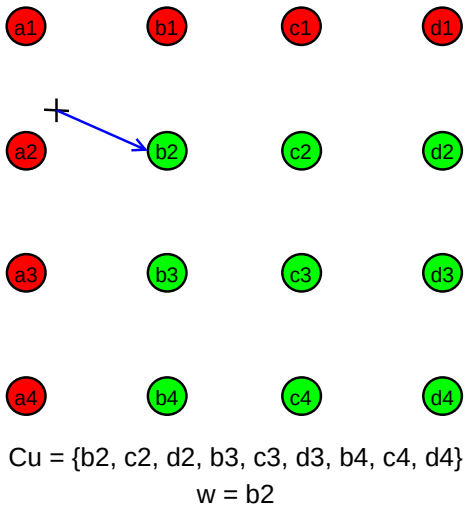


Figure 3.1. **Non-ideal authentication by premise 1 with a discrete system**. The true location is incorrectly claimed to be b2, thus premise 1 is false. The true location should be claimed to be a2, which would result in failed authentication. Since authentication succeeds ($b2 \in c_u$), this is a false positive; the system is non-ideal.

However, even if the location determination system always claims the true location correctly, problems arise if the granularity of the location determination system

[2]Note the system does not discriminate whether or not the entity is using a proxy system to remotely authenticate. But, whichever system is requesting the authentication must reside in the AA for successful authentication.

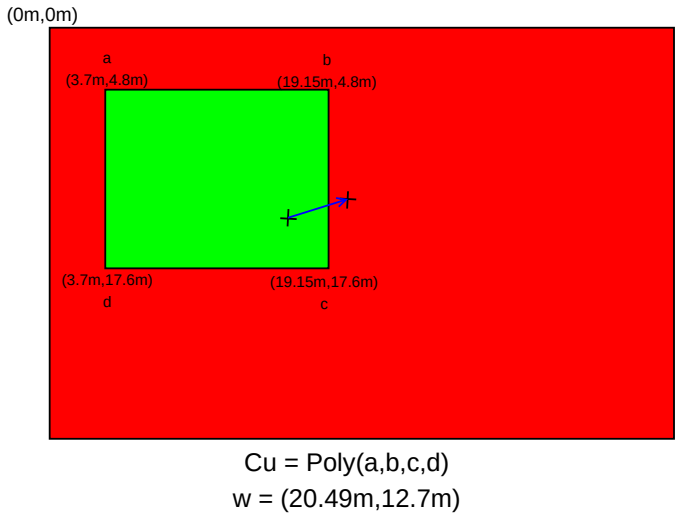


Figure 3.2. **Non-ideal authentication by premise 1 with a continuous system**. The true location is incorrectly claimed to be (20.49m,12.7m), thus premise 1 is false. The true location should be claimed to be (16.482m,13.11m), which would result in successful authentication. Since authentication fails ($w \notin c_u$), this is a false negative; the system is non-ideal.

and the authorized areas (AAs) are different. Consider two types of systems: *discrete* and *continuous*. In a discrete system, locations are claimed to be one of a set of specific points, where a correctly functioning location determination system will select the closest point to be the true location. In a continuous system, locations are claimed to be the specific coordinates of the location (e.g. (x, y)), and the system is working correctly if the coordinates selected through location determination are always that of the true location.

Consider the case that the granularity of the location determination system and the granularity of the authentication system are different. In one case, you can have a continuous location determination system and a discrete authentication system (denoted a C-D system[3]). In the other case, you can have a discrete location determination system and a continuous authentication system (denoted a D-C system).

[3]The first symbol, in this case C for continuous, denotes the granularity of the location determination system. The second symbol, in this case D for discrete, denotes the granularity of the authentication system.

If the two systems do not match up (a C-D or a D-C system), the location claim cannot be compared to the AA, thus premise 2 is false. False positives or false negatives may occur, so these systems are non-ideal (see figures 3.3,3.4).

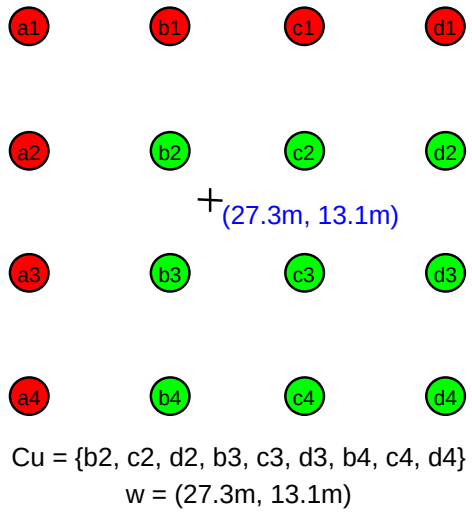


Figure 3.3. **Non-ideal authentication by premise 2 with a C-D system**. The true location is correctly claimed to be (27.3m,13.1m), but this cannot be compared in the discretely defined AA, so premise 2 is false. The comparison might result in the conclusion that the location is not in the AA, which is a false negative; the system is non-ideal.

So in order for the authentication system to have no false positives or false negatives, not only must the location determination system correctly claim the true location, but also the location determination system and authentication system granularities must match. These are the two premises that are sufficient and necessary for the authentication system to be ideal, with no false positives or negatives (see figures 3.5,3.6).

3.3 The Simple Authentication System

The simple ideal authentication system is straightforward. First, both premises described in section 3.2 must be true. That is, the underlying location determination system must be correct and accurate all the time, and the granularity of the location determination system must match up with the granularity of the authentication system.

The complementary information c_u is pre-specified as the set of locations where

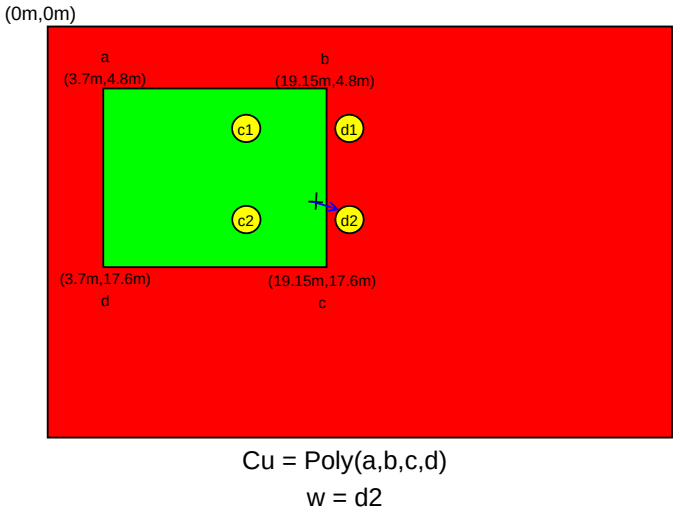


Figure 3.4. **Non-ideal authentication by premise 2 with a D-C system**. The true location is correctly claimed to be d2, but this cannot be compared in the continuously defined AA, so premise 2 is false. The comparison might result that the location is not in the AA, which is a false negative; the system is non-ideal.

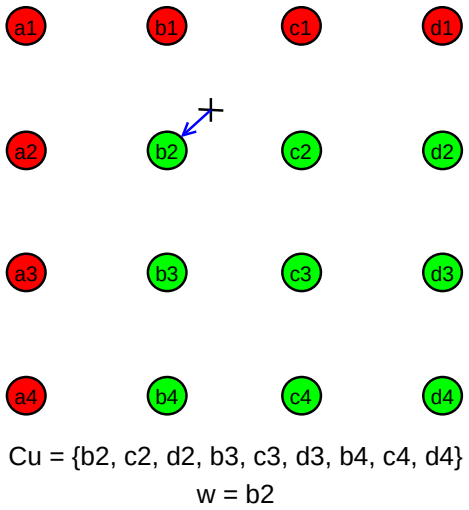


Figure 3.5. **Ideal authentication with a D-D system**. The true location is correctly claimed to be b2, which can be checked with the discretely defined AA. Both premises hold true, thus this system is ideal.

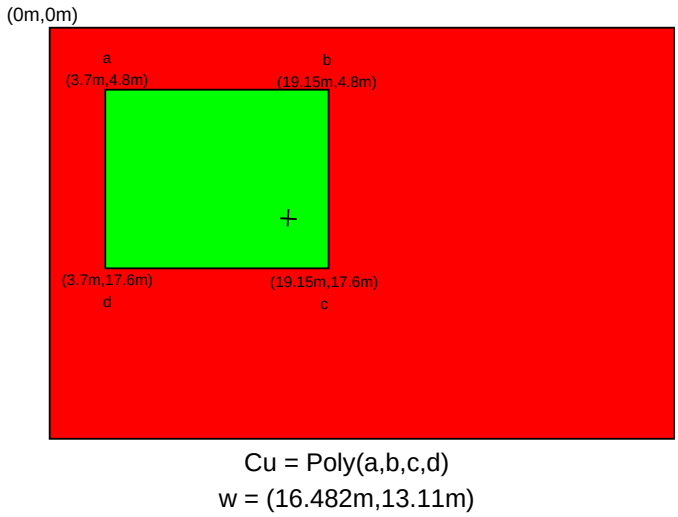


Figure 3.6. **Ideal authentication with a C-C location determination system**. The true location is correctly claimed to be (16.482m,13.11m), which can be checked with the continuously defined AA. Both the premises hold true, thus this system is ideal.

authentication should succeed. Authentication is successful if the given location of the entity is among the locations specified in c_u and unsuccessful otherwise.

3.4 Summary

In this chapter we have discussed a model for location-based authentication, discussed what it would take to have an ideal location-based authentication system, and briefly how such a system would work.

Chapter 4

Using Policy to Handle Location Error

The premises that are required to create an ideal location-based authentication system do not seem to hold true in real-life situations. No location determination system has yet been able to achieve perfect accuracy all the time, so the **Premise 1** has not been shown to be true in practice. Thus in order to make a usable system it has to allow for some false positives and some false negatives[1].

To make a useable system, the authentication system needs to take the *location claim error (LCE)* into account. Many of the location determination papers report the accuracy of their systems by showing the statistics of the location claim error, calculated by the distance difference between the location claim and the *true location* [18, 19, 38, 22, 30, 23, 26, 25]. The amount of error of each system is not constant - there is a distribution of error distances resulting from experiments. For the most part, the actual distribution is a function not only of the environment in which the system is tested but also a function of the algorithm used for computing the location. Generally speaking, these papers tend to report that the error distribution resembles a half-normal or log-normal distribution under their experimental setup (see figure 4.1).

[1]The discussion of authentication systems henceforth will be concerned only with C-C systems, though the terms can easily be ported to the D-D system without loss of generality.

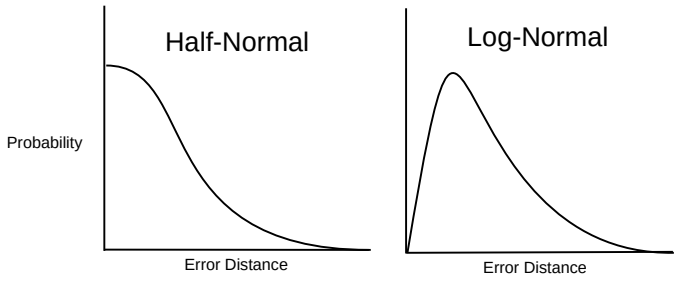


Figure 4.1. **Common probability distributions for location claim error.** These half-normal and log-normal curves closely resemble the error probability distributions reported by most WLAN location determination papers.

4.1 Definitions and notations

4.1.1 True location distribution

The error information about the location determination system can be used to define a *true location distribution (TLD)*. The TLD can be expressed graphically as a three-dimensional surface. Given a location claim, the vertical projection of the TLD is the area consisting of all locations which might be the true location. Clearly, if the LCE is expressed in terms of a distance distribution (which it typically is), the vertical projection of the TLD is the area enclosed in a circle, whose center is the location claim and whose radius is the upper bound of the location claim error (see figure 4.2).

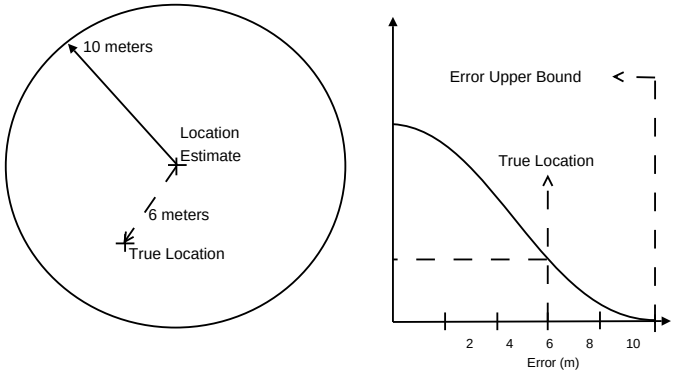


Figure 4.2. **Description of a simple true location distribution**. The upper bound on this half-normal error distribution for an example location determination system is 10 meters. The true location is somewhere within the circle centered around the location claim with a radius of 10 meters. In this example, the true location is 6 meters away.

Though the circle defines where the true location might be, given a location claim, it does not define how likely it is that the true location is at a certain position. The height of the surface above this circle is used to define this. The height of the surface over the projection is the probability that the true location is at that projected point on the two-dimensional area. The greater the height above the zero probability plane, the more likely the true location is at that point. Figures 4.3, 4.4, and 4.5 show what the surface looks like given uniform, half-normal, and log-normal true location distributions, respectively.

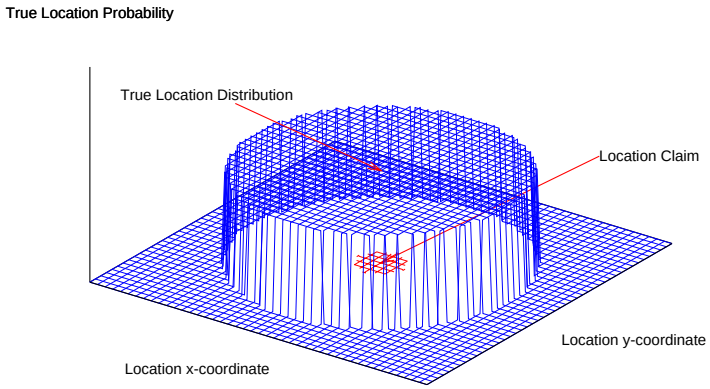


Figure 4.3. **The shape of a uniform TLD**. This TLD shape might be used when only the maximum error distance and mean error distance measures are available.

4.1.2 Authentication error threshold

Now that the true location distribution has been defined, it needs to be put to good use. Consider an example where the location claim is inside a large, rectangular authorized area (AA). In such an example, the entire TLD surface with non-zero height is confined within the AA. This means that there is no way for the true location to be outside of the AA, hence the authentication system should return a result of successful authentication. Likewise, if the location claim and the entire TLD surface with non-zero height is outside of the AA, there is no way for the true location to be inside the AA,

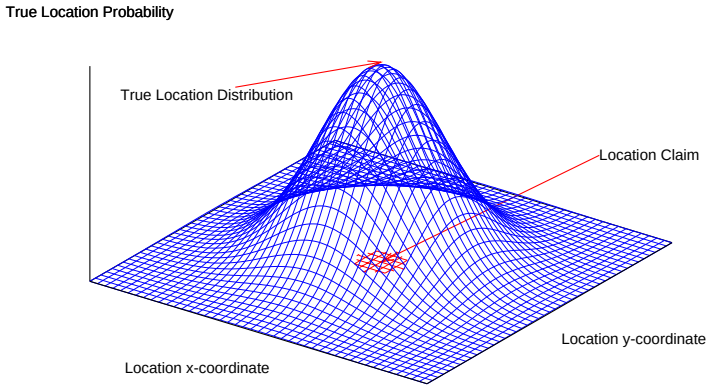


Figure 4.4. **The shape of a half-normal TLD**. Some location determination systems present location claim error results whose distribution resembles a half-normal distribution, which was used to construct this TLD.

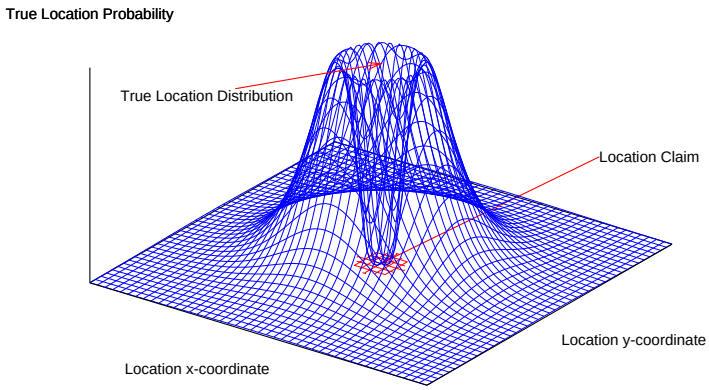


Figure 4.5. **The shape of a log-normal TLD**. Some location determination systems present location claim error results whose distribution resembles a log-normal distribution, which was used to construct this TLD.

hence the authentication system should return a result of failed authentication.

The decision is not so easy when the location claim is near the border of the AA, with part of the TLD surface over the AA and part not over it. If the location claim in inside the AA, there is still a chance that the true location is not. If the location claim in outside the AA, there is still a chance that the true location is inside. The authentication system still needs to make a decision in this case: is the entity authenticated or not? If the authentication system does not make the correct choice, this is a false positive or false negative. Like many other instances where a decision is not clear-cut, the deciding process must be dictated by policy.

A possible metric to use for policy is the fraction of the volume underneath the surface of the TLD. In particular, the system would be interested in the volume, V_0, that is underneath the surface of the TLD and above the intersection of the vertical projection of the TLD surface and the AA (this intersection is hehencefortheferred to as the *intersection area*). Strictly speaking, this volume, called the *intersection volume*, will always be less than or equal to the total volume, V_t, underneath the entire surface. The *fraction of location probability volume (FLPV)* is defined as the fraction $\frac{V_0}{V_t}$.

The closer the FLPV gets to 1, the more likely the true location is inside the AA. The closer it gets to 0, the more likely the true location is outside the AA. Notice how this fraction helps solidify the case where the entire TLD is inside the AA; the FLPV in this case is 1. If the entire TLD is outside the AA, the FLPV is 0. To handle border cases, the authentication system defines a policy called the *authentication error threshold (AET)*. The AET is defined similarly to the FLPV; it is also measured as a fraction of the total volume underneath the entire surface of a TLD. The FLPV is a measure for a specific location claim and AA, whereas AET is used for policy decisions for a generic location claim.

The execution of the policy could be as follows: given a location claim and a predefined authorized area (AA), if the associated FLPV is greater than or equal to the AET, authentication succeeds, otherwise authentication fails. For example, if the AET is 0.5, then, for authentication to succeed for a given location claim, the FLPV must be

at least 0.5. That is, at least 50% of the volume underneath the TLD surface must be above the AA for successful authentication.

Clearly, this policy definition promotes some false positives and some false negatives. But since ideal conditions for an authentication system do not exist, such allowances are necessary. Allowing the administrator to set the policy gives the site more flexibility to balance the requirements for availability (fewer false negatives) and integrity (fewer false positives).

The authentication error threshold not only provides a policy mechanism, it also provides some useful insight into the effectiveness of the authentication system. If the AET is set to 0.75, then for any AA the upper bound on the rate of both false positives and false negatives is 25%[2]. This upper bound is *only* approached when location claims are close to the authorized area border. Thus, depending upon the size of the AA, the true rate of false positives and false negatives of a random sampling might be much less.

4.1.3 Authorized Claim Area

Given the authentication system described thus far, the first of three steps in authentication would happen during the offline phase. First, is the definition of the *authorized area* (AA) and *authentication error threshold* (AET) for an entity. The second step, during the online phase, would be when the system computes the *fraction of location probability volume* (FLPV) given the *true location distribution* and the *location claim*. The authentication decision, the third step, relies solely upon the comparison between the FLPV and the AET.

One problem with this design is that there are some heavy computations that must be performed during the online phase. Specifically, calculating the FLPV can be computationally expensive, especially when the intersection area is not a regular shape, like a rectangle. If the intersection area is an irregular polygon, or portions of the perimeter are curved lines, the volume computation could take a long time.

[2]Assuming that the true location distribution is accurate for the location determination system deployment.

Another problem with this design is that it does not fit neatly into the authentication model defined in section 3.1. In the ideal system, the complementary information c_u is defined as the AA. But for a realistic system the AA will not work because the authentication functions (L in the model) do not take the location claim error (LCE) into account.

Instead of changing the model to fit the authentication system, the system could be changed to fit the authentication model, while at the same time reducing the amount of online phase processing required. During the offline phase, the authentication system could create an *authorized claim area (ACA)*, which is the set of locations claims that would result in successful authentication for a specific entity. The rationale is simple: given two location claims X and Y, if $X = Y$, then $FLPV_X = FLPV_Y$, assuming that neither the true location distribution (TLD) nor the authorized area (AA) for the entity changes. This means that the FLPV for a specific location will not change over time, so precomputation is possible.

Regardless of the specifics of the volume computations, the ACA can be used to make the system fit the authentication model. More precisely, the complementary information can be defined as the ACA (that is, $c_u = ACA_u$). Now, during the online phase, the authentication system would work by determining if the location claim is an element of the ACA, which is precisely the authentication function defined by the authentication model (Section 3.1).

One way to perform the FLPV computations offline is for the authentication system to simulate the authentication process for all possible location claims, and those location claims that result in successful authentication become elements of the ACA. Other more efficient techniques, like interpolation and extrapolation, can be used to reduce the number of operations required.

During each of the simulated computations, the volume under the TLD surface over the intersection area must be computed. The computations themselves are not necessarily difficult. The normal distribution with mean 0 and standard deviation $\frac{1}{\theta}$ can

be expressed as in equation 4.1 [52][3].

$$P(x) = \frac{2\theta}{\pi} e^{\frac{-x^2\theta^2}{\pi}} \tag{4.1}$$

This curve is 2-dimensional, but the 3-dimensional surface can be created from it by rotating the curve about the vertical axis. The resultant curve is equation 4.2, which was shown in figure 4.4.

$$z = \frac{2\theta}{\pi} e^{\frac{(-x^2-y^2)\theta^2}{\pi}} \tag{4.2}$$

Multiple integration techniques can be used to find the volume underneath this surface, but that is actually overkill for at least three reasons:

- Not all location claim error distributions (in fact probably none) can be easily expressed in the form of an equation.
- Only portions of the entire volume under the surface are going to be computed in order to build the authorized claim area.
- Exact volume measures are not necessary, only close approximations.

For these reasons, it makes sense to use approximation techniques, for example *Riemann Sum Approximation*[53], which uses 3-dimensional boxes. Of course, if the distribution is not in the form of an equation, the height of each box needs to be determined by other means. For example, if the distribution is in the form of a lookup table, it might be necessary to apply interpolation techniques to obtain the correct height approximations for points in between entries in the table. Another option is to try to fit a curve to the location claim error (LCE) data.

A further advantage to this approximation technique is that varying the ranges of the x and y coordinates in the computation is straightforward. Therefore, computing only a partial volume under the distribution is easy. The process can even be sped up by using a table to store the volumes of Riemann boxes at particular positions relative to

[3]The mean of 0 indicates that the apex of the curve is at $x = 0$, which is the half-normal distribution.

the location claim, because they will always be the same since the distribution does not change. So the only computations that really need to be done for each potential location claim is looking up the volumes in the table and summing them up.

4.2 Interesting Setups and Policies

In order to analyze different situations involving different LCE distributions, AETs, and AAs, the notation $F - X - Y$ will be used to identify the situations, where:

- F - describes the shape of the TLD for the location determination system,
- X - defines the AET for the situation as a percentage, not a fraction,
- Y - describes the general shape of the AA involved.

The number and type of distributions described is unlimited, but there are a few that are interesting and instructive:

- Uniform - the non-zero height of the TLD surface is level, meaning all possible true locations are equally likely (figure 4.3),
- Half-normal - the TLD surface resembles a half-normal distribution curve (figure 4.4),
- Log-normal - the TLD surface resembles a log-normal distribution curve (figure 4.5).

The general shape of the AA, denoted Y above, need not be precisely defined. For example, $Y = rectangle$ might mean that the AA is rectangular in shape, but does not give an indication of the dimensions.

In addition to the setup notation, a two dimensional depiction of the setup will be used to improve readability and understandability. Naturally, representing a three dimensional depiction (in this case the third dimension is location probability) in two dimensions is not straistraightforwardure 4.6, which corresponds to the half-normal TLD (figure 4.4), shows how the use of gradients will help to make this transformation.

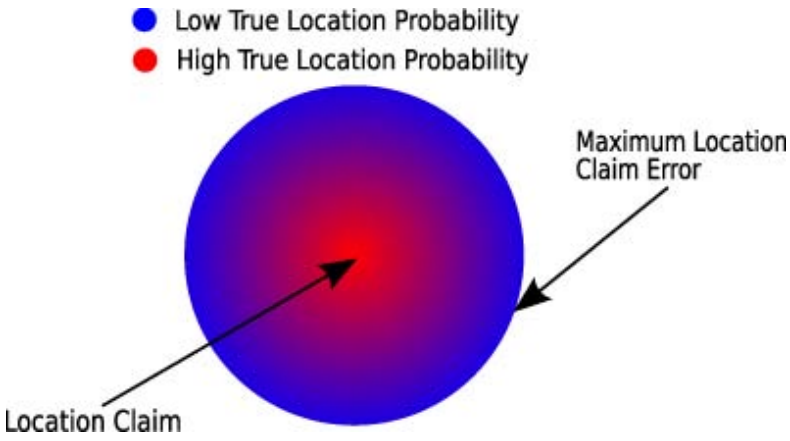


Figure 4.6. **The half-normal distribution depicted in two dimensions**. The red parts of the figure are areas with high probability for the true location. Blue parts are areas with low probability for the true location. Purple areas are areas of moderate probability.

The blue portions of the figure represent areas of low true location probability. Red indicates that the probability is high. Refer to figures 4.3 and 4.5 for an idea of how the two-dimensional depictions would look for the uniform and log-normal distributions, respectively.

Following are some example setups of F, X and Y that have instructive or interesting properties.

4.2.1 Any-100-Any: No False Positives

Consider an environment where it is vitally important that there are no false positives. That is, if a client is outside the AA, authentication will fail. Assuming that the TLD is correct for the deployment, such a system is possible as long as the AET is 1 (see figure 4.7). Since all possible true locations reside inside the AA, there are no false positives.

Unfortunately, this system will also have a lot of false negatives, particularly along the outside portions of the AA.

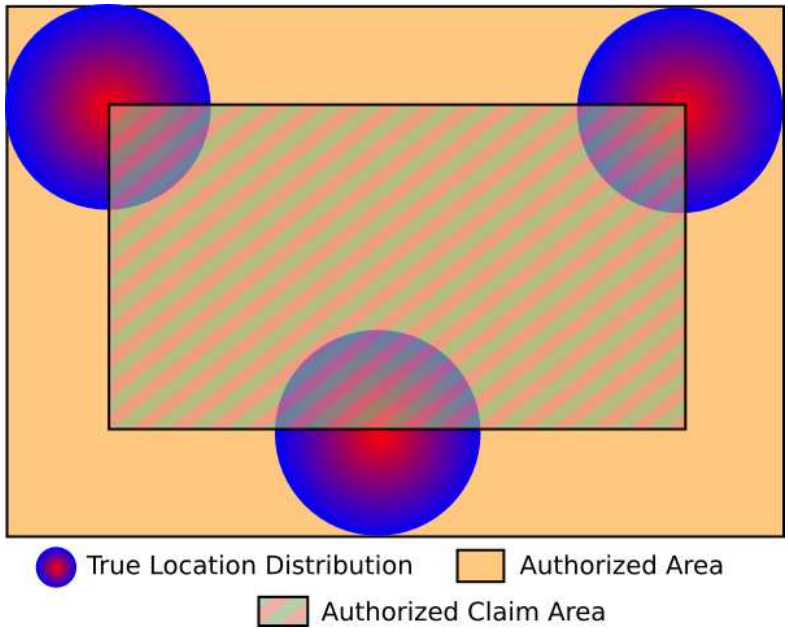


Figure 4.7. **Any-100-Any: No False Positives**. The entire ACA is enclosed within the AA. It is not possible for a true location to be outside of the AA when a location claim is within the ACA. However, there will be a lot of false negatives when location claims are outside of the ACA, but inside the AA.

4.2.2 Any-1-Any: Almost No False Negatives

A system that does not allow for any false negatives is actually not practical using the authentication system described. Having no false negatives means that the AET would be 0. But, if you have such a policy, say Uniform-0-any, all possible location claims have a volume greater than or equal to the AET. Therefore, authentication will *always* succeed regardless of the location claim. In other words, the ACA covers all possible location claim locations, and there will be a very large number of false positives.

To fix this problem, it is necessary to choose a different goal: almost no false negatives. If there is a low probability, say 0.01, that the true location is inside the AA when the location claim is outside, then most of the time there will be no false negatives. Additionally, the rate of false positives is bounded, as the ACA extends a little outside the AA, but does not cover the entire location claim area (see figure 4.8).

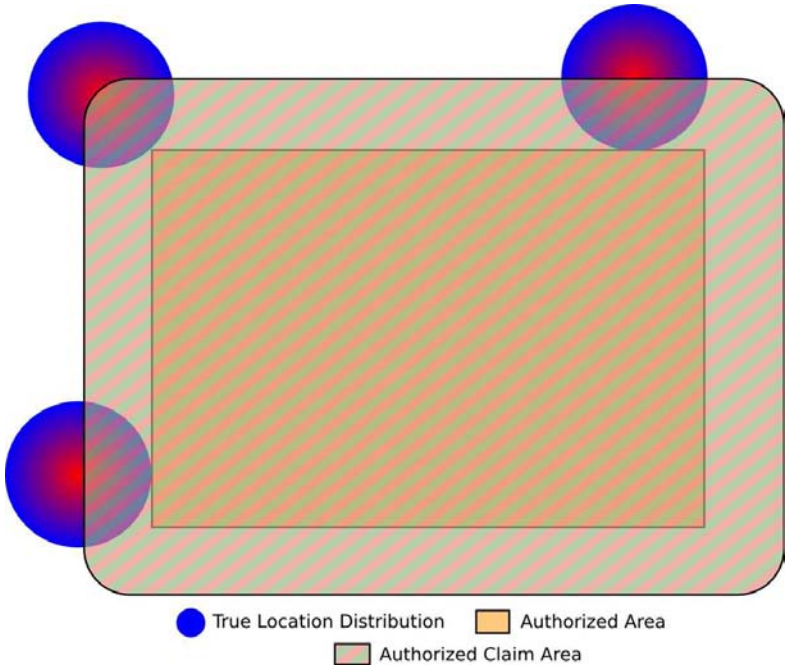


Figure 4.8. **Any-1-Any: Almost No False Negatives**. The very low AET eliminates almost all false negatives as the ACA extends beyond the AA. However, there will be a lot of false positives when location claims are just outside of the AA.

4.2.3 Any-50-Rectangle: Close Rectangular Fit

The authentication error threshold is a measure of the upper bound on the false negative rate, R_{FN}, for location claims just outside the ACA. This measure is also related to the upper bound on the false positive rate, R_{FP}, for location claims just inside the ACA. That is, $R_{FP} = 1 - AET = 1 - R_{RN}$[4]. If a deployment is to balance false positive and false negatives in a usable manner, the administrator can manipulate the AET to obtain the desired balance. It might be advantageous to try to fit the ACA into close to the same size and shape as the AA, which may provide a good balance between the upper bound on false positives and false negatives.

An interesting policy is obtained if an AET of 0.5 is selected for a large rectangular AA, as shown in figure 4.9. When the AET is 0.5, the ACA fits very closely to the AA, especially along the straight edges. Along the corners, the ACA has a rounded edge

[4]It is important to keep in mind that the upper bounds on the false positives and false negative rates are only approached near the borders of the ACA.

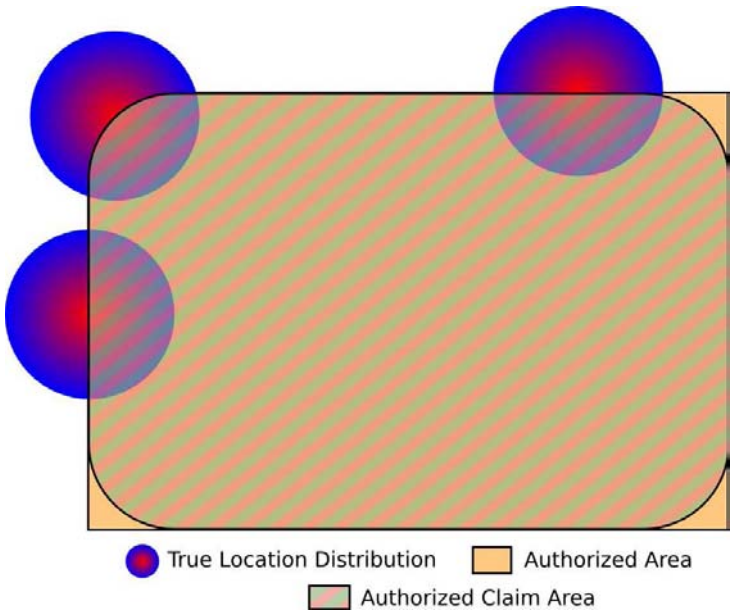


Figure 4.9. **Any-50-Rectangle: Close Rectangular Fit**. With an AET of 0.5, the ACA closely fits the size and shape of the AA, particularly along straight edges. A close fit represents a good balance between false negatives and false positives. In this figure and those following, the TLDs overlap the AAs, which is why the upper-left AA corner is not visible (NOTE: The upper left corner is not shown because the TLD overlaps it).

inside the AA. A higher or lower AET will result in the ACA being outside or inside (respectively) the AA along straight edges.

4.2.4 Half-normal-25-Rectangle: Covering Right-angle Corners

Sometimes it is desirable for the ACA to completely cover the AA. Section 4.2.3 showed how an AET of 0.5 will make the ACA cover most of a large rectangular AA, with the exception of the corners. To fully cover the corners, an AET of 0.25 is necessary, as shown in figure 4.10. If the corner of the rectangular AA is at the center of the TLD projection circle, the intersection area and the intersection volume is 25% the total area and volume. Of course, a policy with an AET of 0.25 also means that the false positive rate when the claim is at the border of the ACA is 75%, which may be too large for certain deployments.

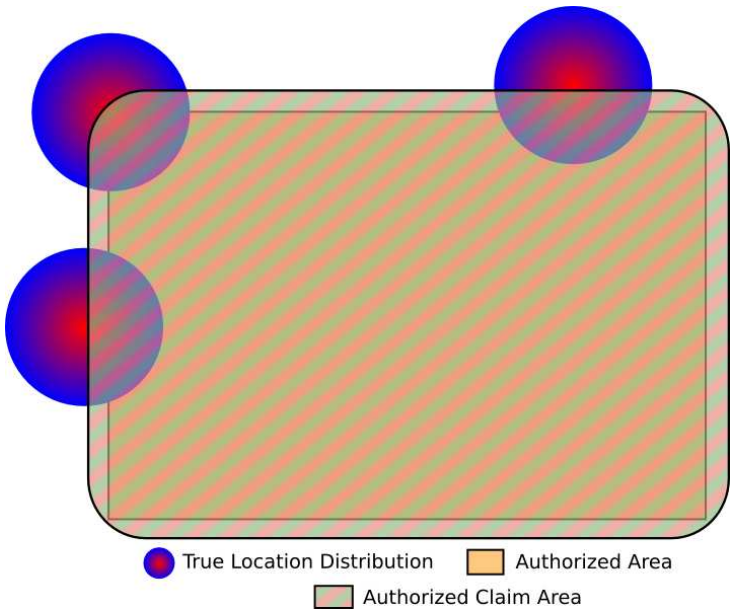


Figure 4.10. **Half-normal-25-Rectangle: Covering Right-angle Corners**. Right angle corners are not well covered when an AET of 0.5 is used, which will result in a high rate of false negatnegatives incorners. If corner coverage is important, an AET of 0.25 will allow for better corner coverage at the expense of a higher rate of false positives along straight edges.

4.2.5 Uniform-75-Rectangle vs. Half-normal-75-Rectangle: Importance of a Weighted Distribution

Figure 4.11 shows how the shape of the ACA compares for two systems, one with a uniform distribution, and one with a half-normal distribution. For the half-normal distribution, the ACA conforms much more closely to the boundary of the AA. Assuming that the TLDs accurately depict the distribution of location claim error, the closer fit and larger ACA signifies that there will be fewer false negatives for the half-normal distribution, particularly near the borders of the AA. Interestingly, the upper bounds of the false positive and false negative rates are not affected by the improved fit. This verifies that there is a direct correlation between the AET and these rates.

The significance of this is that it is much more desirable to have a location determination system that more accurately estimates the client's location. In addition to a smaller vertical projection area, a higher true location probability close to location

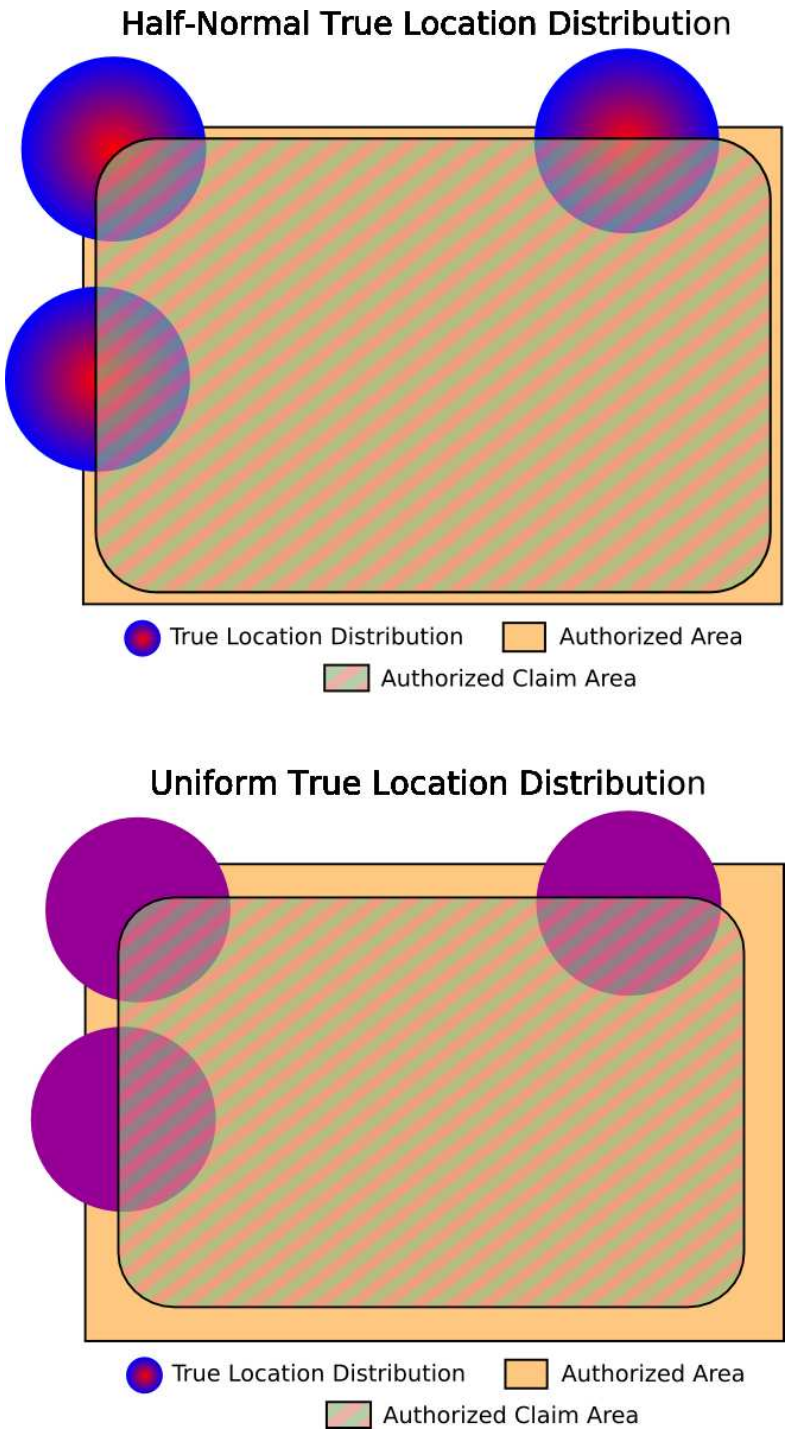


Figure 4.11. **Uniform-75-Rectangle vs. Half-normal-75-Rectangle: Importance of a Weighted Distribution**. A centrally weighted TLD yeildyieldstter fitting ACA, which reduces false negatives compared to a uniformly distributed TLD.

claims will significantly reduce false positives and false negatives for points near but not directly on the ACA border. This becomes increasingly important as the size of the AA relative to the maximum location error decreases.

Figure 4.11 also illustrates the pitfalls that are encountered if the TLD assumed for a particular authentication system deployment is not accurate. Consider a system that assumes a half-normal TLD, but the actual TLD is uniform. This will result in an increase in the upper bound on the false positive rate, closer to 50%. However the half-normal assumption also decreases the false negative rate that would be experienced if the system had assumed a uniform TLD.

4.2.6 Uniform-50-Irregular: Fitting an Irregular Authorized Area

Several conclusions can be drawn from the irregularly shaped AA shown in figure 4.12. First, the authentication system described can be used to fit non-standard shaped AAs with reasonable success. Rounded boundaries on the AA do not present a significant problem to the authentication system, nor do concave corners or corners that do not form right angles.

The second observation is that obtuse turns and corners in the AA are covered more closely by the ACA than are sharper angles. Third, whereas convex corners of the AA result in the containment of the ACA, concave corners result in the ACA overlapping the AA.

4.2.7 Any-50-Holes: Handling Holes in the Authorized Area

In addition to irregularly shaped AAs, this authentication system can handle holes in the AA fairly well in some cases. Figure 4.13 gives an admittedly contrived example of an AA with a hole in the center. Such a layout might be used in buildings with a central elevator.

A system with a half-normal true location distribution would account for the hole, but this may not be the case for the uniform distribution. The size of the hole is also relevant, as a larger hole might be accounted for by a uniform TLD.

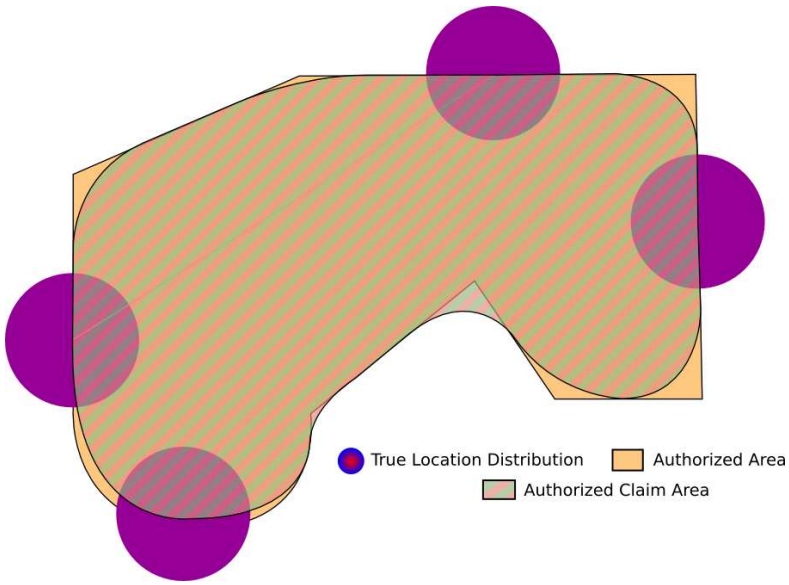


Figure 4.12. **Uniform-50-Irregular: Fitting an Irregular Authorized Area**. The described authentication system easily adapts to irregularly shaped authentication areas. Obtuse turns and corners result in closer ACA coverage than sharper turns and corners. Concave turns and corners result in an overlapping ACA, whereas convex turns and corners have an enclosed ACA.

4.2.8 Uniform-50-SmallRect: A Relatively Small Authorized Area

Serious problems arise when the AA is relatively small compared to the maximum location error distance (see figure 4.14). This situation can happen if the AA itself is small, or if the maximum location error distance for a location determination system is large, or both. Even with an AET of 0.5, the AA is poorly covered by the ACA, which means that there will be a lot more false negatives than for a larger AA. Furthermore, if you consider the area that contains all possible location claims that may result in positive authentication, that area is proportionately much larger than the AA than it would otherwise be for a larger AA.

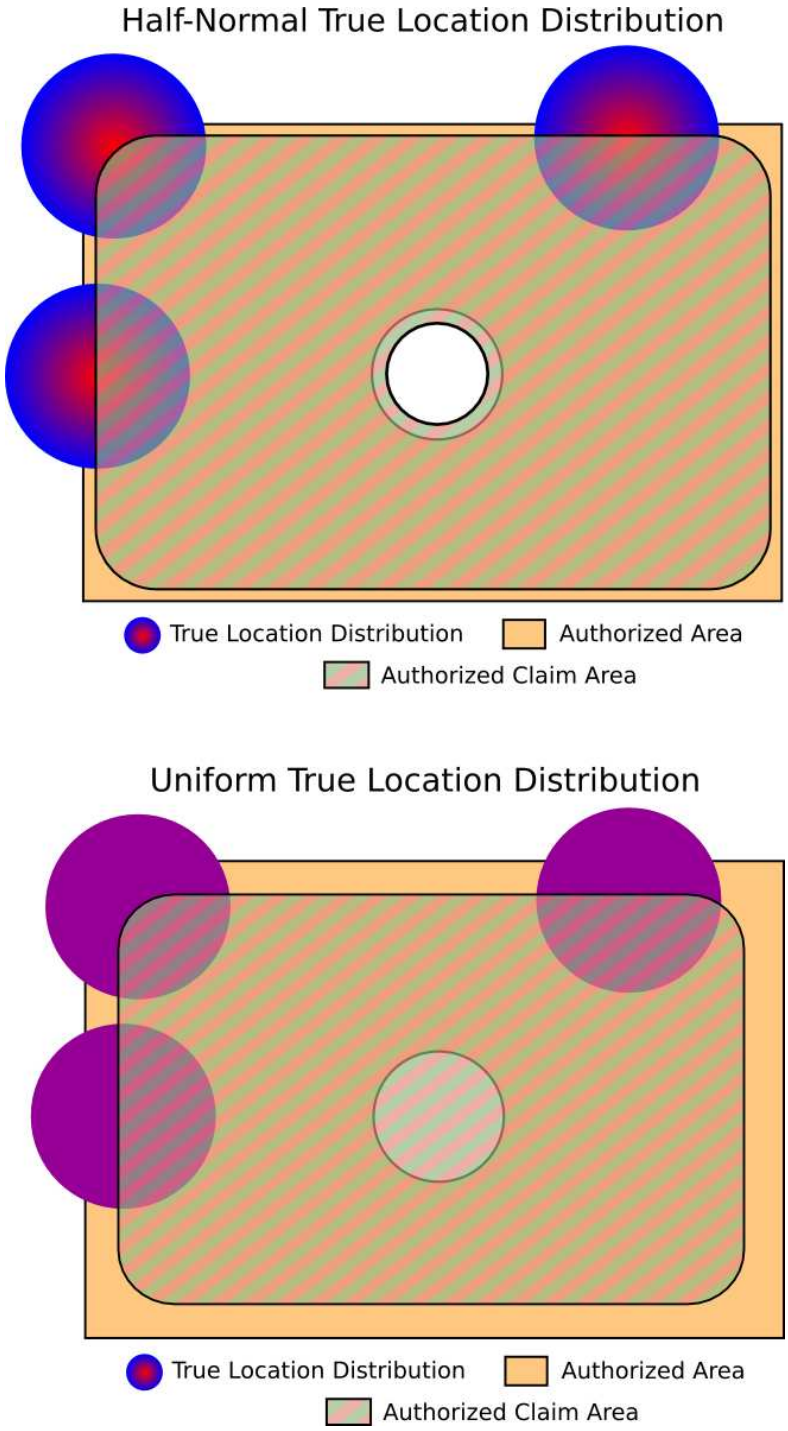


Figure 4.13. **Any-50-Holes: Handling Holes in the Authorized Area**. Centrally weighted TLDs are much better at handling holes in the AA than are uniform TLDs.

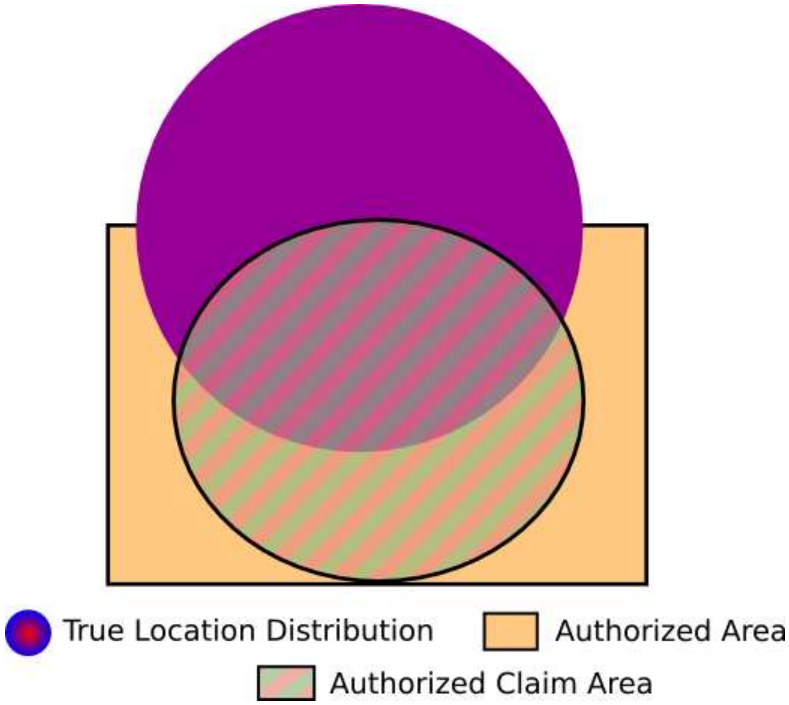


Figure 4.14. **Uniform-50-SmallRect: A Relatively Small Authorized Area**. Authorized areas that are relatively small compared to the size of the TLD result in significant issues concerning the size and shape of the ACA. Additionally, the area consisting of the potential true locations that might result in successful authentication is quite large when compared to the area of the AA.

4.3 Summary

This chapter has discussed how policy can be used in a location-based authentication system to handle the fact that current location determination systems produce inaccurate location claims. An administrator selects an authorized area (AA) of any shape and size and an authentication error threshold (AET), which is a policy which measures the strictness of the authentication system. Then the system can compute the ACA in the offline phase. During the online phase, the authentication system obtains a location claim for an entity from the location determination system and successfully authenticates the entity only if the estimate is within the ACA.

The administrator needs to select a policy that is appropriate for the deployment of the authentication system. A high integrity deployment protecting important information might be much more restrictive (higher AET) than a system with publicly available information. This authentication system allows a variety of different policies to be used depending upon the need.

Chapter 5

A Realistic Authentication System

In this chapter we go into the details of how a two-factor authentication system could be deployed using the results of previous chapters. Chapters 3 and 4 discussed location-based authentication in terms of a one-factor authentication system. However, not all authentication decisions can be made using location as the only factor. For example, in an office building environment, there are soft physical controls for individual offices, meaning that, though the office owner is the most likely occupant, others may be able to enter the office unrestricted.

For this reason, location-based authentication makes more sense as part of a two-factor authentication system. The additional factor (other than location) may be any accepted method, be it password, token, biometric, or something else. In this chapter we discuss a two-factor authentication where an entity needs to provide the appropriate username/password pair, and must be in an authorized location; authentication is based upon "what you know" and "where you are".

5.1 General Discussion

The authentication system can be described in fairly general terms, allowing a possible future deployment to fill in specific details. The realist authentication system would include a set of *entities* that use *clients* to try to log on to a *login server*. The server provides some sort of service which requires that the entity be authenticated. A client has a *wireless network card*, which has associated and authenticated with the *wireless network*. This allows the *location determination system* to locate the client. The login server communicates with a client to obtain authentication information. Some of this information is used by the login server to query the location determination system for the location claim of the client. The login server supplies the authentication information, which now includes the the location claim, to the *authentication server*. This server compares the supplied authentication information to the complementary information (see Section 3.1) and determines whether authentication succeeds or fails. The result is sent back to the login server, which in turn forwards the result to the client. Refer to Figure 5.1.

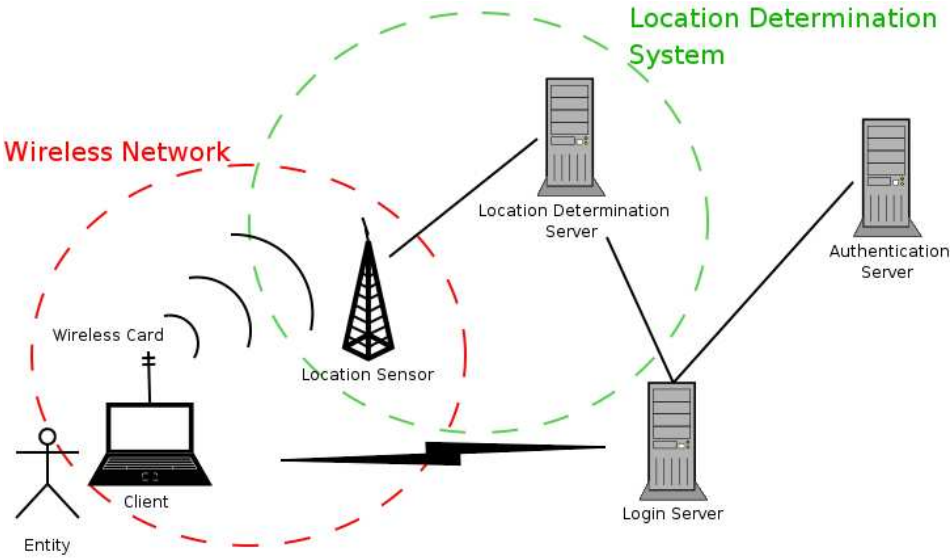


Figure 5.1. **General authentication setup**. This figure depicts the logical setup of a generic location-based authentication system, which may or may not have additional factors in for authenticating.

There are some important aspects to note about this general setup. First, it does not specify the medium over which the client and login server communicate. In general, the two could communicate over any type of infrastructure, be it over wired LAN, another wireless network, the same wireless network used for determining location, the Internet, or any combination of these. In fact, the client and the login server can be the same computer. The attempt to log on to the computer might be the initial logon to the system (i.e., the console), or it might be an attempt to access a local service.

Second, it is also possible that the different servers might be grouped into one single server, split into multiple servers, or anything else in between. The generality of the authentication process does not require a specific implementation.

5.2 Authentication Specifics

5.2.1 Authentication Initialization

The first step in implementing any authentication system is getting the infrastructure in place. This may involve getting the appropriate servers on the network and setting up certificates and/or trust relationships among them. When location is used in authentication, additional infrastructure must be in place, including the location determination system and any sensors or access point modules that are used by the location determination system. Much of this infrastructure may already be part of the deployed wireless network. Next is the offline phase of the the location determination system, which includes the building of location models and tuning[1] (see Section 2.1.2). Finally, the location determination system must be tested in order to determine the distribution of location errors (required to describe the shape of the TLD, as explained in Section 4.1.1).

The next step is for an authentication administrator to create user accounts and initialize complementary information for each one. In a system that involves passwords,

[1]Some systems do not have an offline phase for model building and tuning, as is the case in [32], where the model is built continuously during the online phase

the administrator may create temporary passwords and distribute them to each user. In a system that involves location, the administrator would need to select an appropriate AET and use it and the TLD to compute ACAs for each user based upon where users would be required to be located to access a service (see Sections 4.1.2 and 4.1.3).

5.2.2 Authentication Algorithm Implementation

When a user wishes to access a service that requires authentication, there must be protocols to facilitate communication and access control among all of the necessary parties. An authentication system can be extensible by providing a flexible framework in which different authentication methods are used. This is advantageous because these systems allow for new authentication methods, like location, to be added without having to redesign or rewrite the entire system. The authentication frameworks discussed here are 802.1X and PAM.

IEEE 802.1X: Network Port Authentication

IEEE 802.1X is a authentication framework built on top of the Extensible Authentication Protocol (EAP) [3, Chap. 6]. Its primary function is to control client access to a network, allowing data sent from the client to pass into the network only after the client is authenticated. The protocol works just above the MAC layer, not at a network layer like the Internet Protocol (IP). This means that the authentication and access control system must be accessible on the local network; remote authentication is not possible.

The advantage to 802.1X is that a client has no means of communication with other servers until it is authenticated. This is a default deny policy that provides fairly strong protection against network attacks. Another advantage is that the client need only be authenticated once in order to obtain access to all of the services on the network to which it was granted access. However, this is also a disadvantage because it makes finer-grained access control more difficult, requiring an additional layer of authentication. Not having remote authentication capabilities is another disadvantage.

In 802.1X, the client (called the *supplicant*) attempts to access the network, but is initially denied access by an *authenticator*. The authenticator requests authentication information from the supplicant, and once received forwards this information to an *authentication server*. The authentication server may request more authentication information from the supplicant, so the authenticator acts as a relay of authentication requests and responses between the two. If the authentication server indicates that authentication was successful, it notifies the authenticator, which opens up the "network port", allowing traffic from the supplicant to pass through to the rest of the network.

The authentication server may implement any form of authentication framework. The RADIUS system is commonly used because it can be a front-end for many other types of authentication services including LDAP directories, Kerberos realms, Microsoft Windows user accounts, and even PAM system.

PAM: Pluggable Authentication Modules

Pluggable Authentication Modules (PAM) is a standard authentication system used by many UNIX-like operating systems. PAM provides an authentication service to the system by providing a standard interface to applications, and a centralized location for policy management [54]. It also allows programmers to develop and use new forms of authentication without requiring any application rewriting or recompiling.

Because PAM works through an application to provide an authentication service, it does not have the same remote authentication restriction that 802.1X has. However, it does not prevent clients from communicating with hosts on the network, as 802.1X does. Instead PAM may be used solely to control access to restricted services (unless PAM is wrapped around by another protocol, say 802.1X with RADIUS).

The PAM framework allows for high flexibility in terms of what methods can be used to authenticate different services. The policy definition files for PAM generally reside in the `/etc/pam.d` directory, and each file is named after the process for the service that requires authentication. Refer to Appendix B for an example standard `/etc/pam.d/ftpd` file, which would be used to describe the authentication policy for an

FTP server. The example policy allows standard UNIX password authentication (using `pam_unix.so`).

An implementation of PAM-controlled location-based authentication would consist of a new PAM module (`pam_location.so` in the example PAM policy file). This file would need to be a dynamically loadable module compiled from source code, which would need to be written to provide the wireless location-based authentication service. In general, this module would need to initiate the comparison of location claims to ACAs stored on the authentication server and act accordingly based upon the comparison results.

The example PAM configuration file (Appendix B), once all the modules are in place, would be able to provide two-factor authentication to protect access to the FTP service. The PAM framework is used for the remainder of this chapter in the discussion of two-factor authentication using passwords and location.

5.3 Attacks and Design Decisions

A two-factor authentication system is designed to protect against unauthorized access to resources when one of the factors might be exploitable or weak in specific instances. In order to design how the authentication algorithm could be performed, we must first discuss different algorithm designs and attack scenarios.

For the following discussion, it is assumed that an attacker has obtained the username and password that would otherwise result in successful authentication were location-based authentication not also in place. It is also assumed that the attacker has not obtained physical access to the authorized claim area (ACA), nor has it obtained control of any computing system inside the appropriate ACA. These two assumptions are necessary for the discussion, otherwise the discussion would involve more than the analysis of the location-based authentication system.

5.3.1 Obtaining the MAC Address

The location determination system needs some static key in order to track and store the location of wireless clients. The MAC address of the client wireless network card communicating with the wireless network with location determination capabilities serves this purpose well. It is usually unique, rarely changes, and is easily accessible at the MAC layer, which is the layer at which the wireless network operates. Using a key at a higher layer, like an IP address, may also be possible but problems may occur if, for example, the network uses the Dynamic Host Configuration Protocol (DHCP) to assign IP addresses dynamically. In that case, the key used for looking up the location may change frequently. Even if it did not change frequently, the wireless network may not be able to obtain this higher layer information from frames. For these reasons, when the login server needs to obtain the location of a client, the MAC address should be used as the static key.

However, whereas the wireless network (and the location determination system) can easily obtain the MAC address of the client, it is not a simple process for the login server. The client may be logging into the server over a different medium, like wired Ethernet. Furthermore, the communication may be traversing several routers and gateways along the way, so the original source MAC address information in the MAC layer header of packets would be destroyed. Only the wireless network to which the client associates (and its location determination system) are capable of determining the MAC address of the wireless card. Thus, it would be a difficult task for the login server to look up a client's location based upon MAC address.

Client Supplies MAC; Authentication server maps MAC to ACA

Consider a scenario where the authentication server maintains a mapping of MAC addresses to ACAs and the client application supplies its wireless network card MAC address as credential information to the login server. In this case, the login server must supply the MAC address to the location determination system and obtain the

location claim. Then the login server presents the MAC and location claim to the authentication server. Authentication succeeds if the supplied MAC address has a location claim that resides within its associated ACA.

This scenario is insecure because information from the client cannot be trusted. It is a fallacy to assume that the client, which may be under the control of an attacker, is supplying the actual MAC address that is using the wireless network. Indeed, an attacker need simply supply (spoof) a MAC address that he knows is currently sitting within its ACA. Simple observation of traffic on the wireless network may be sufficient to obtain such a MAC address.

Client Supplies MAC; Authentication Server Maps username to ACA

This scenario is similar to the previous one only now the authentication server uses a username to obtain the ACA. Now, if an attacker simply supplies just any MAC address that resides within its ACA, authentication will probably fail. This is because the login server uses the MAC to obtain the location claim but supplies the username (not the MAC address) to the authentication server to match the location claim with the ACA.

Though slightly better, this system is still quite insecure. Now, the attacker needs to supply a MAC address that means more strict conditions. The MAC address must be registered to the user whose username and password the attacker stole, and that MAC address must be associated with the wireless network and located within the user's ACA. When correct information is supplied, the authentication server will obtain a location claim for the client that is truly associated with the user, which resides within the ACA; then authentication will be successful.

Registered MAC; Authentication Server Maps username to ACA

The two previous scenarios showed how it is a fallacy for the system to assume that the client will supply the true MAC address of the wireless card associated with the client. Consider the case where a user has previously registered a MAC address which

he or she will always use on the wireless network. Now the login server supplies the username to the location determination system (instead of a MAC address) in order to obtain a location claim. The location determination system maps the username to the registered MAC addresses and returns the location claim of that specific MAC address. The username, password, and location claim are then sent to the authentication server.

Now there is no longer a reliance on the client to supply the appropriate MAC address, which should alleviate some of the problems of the previous scenarios. Unfortunately, this system is actually even less secure. Now, the attacker does not even need to come up with a MAC address. He or she simply needs to wait until the true user of the account has associated the true wireless card (with the registered MAC address) with the wireless network inside the ACA. Then the client issues the username and password, the login server uses the username to obtain the location claim (which will return the location of the true user's client), and the location claim is sent to the authentication server, resulting in successful authentication. The system now relies only on the supplied username and password, and on the timing of the attacker; the location-based authentication is rendered useless.

Note that it is not necessary that an offline static mapping of usernames to MAC addresses be set up *a priori*. Even if the system creates the mapping using data from an EAP[2] wireless authentication system (i.e., there is a secure dynamic mapping of username to MAC address), the attacker still only needs to wait to log on until the real user is communicating with the wireless network from within the ACA using the registered wireless network card.

Challenge-Response

The main problem with the above design scenarios is that none of them can obtain the MAC address of the wireless card actually being used by the attacker's client (if any). In fact, in none of the above scenarios does the attacker even need to have a

[2]The *Extensible Authentication Protocol*, defined in RFC 2284 [55]. Other wireless authentication systems (e.g. LEAP, PEAP, 802.1X, etc.) are also possible [56] [3, Chap. 6].

wireless card. But many of these security issues can be solved if the system can somehow be confident that it has obtained the actual MAC address of the wireless network card used by the user's computer[3].

To obtain the MAC address, the login server could issue a challenge to the client. The client must issue the correct response *over the wireless network* in order to authenticate. This process would serve two purposes. First, as the client sends the response over the wireless network, the location determination system could generate a location claim for that MAC address. Second, if the correct response was reasonably unique, the location determination system could generate a mapping of username to MAC address. To do this the login server must remember what response should be obtained by the username, and if it receives that response, the MAC address of the response packet could be bound to the username.

5.3.2 The High-Integrity Authentication Process

The entire process would be as follows, as depicted in Figure 5.2. First, the user issues a request to access a service on a login server. The login server replies with a request for the username, password, and a response to a challenge. As an example challenge, the server could send a pseudo-random 64-bit nonce to the client, and the client must echo the nonce over the wireless channel as the response. Upon seeing the nonce response packet over the wireless network, the authentication server associates the source MAC address seen in that packet with the username[4] and generates the associated location claim.

After the client supplies the appropriate responses, the login server supplies the username to the location determination system to obtain a location claim. The username, password, and location claim are presented to the authentication system, and finally authentication either succeeds or fails.

[3]The use of the term *computer* is a generalization for any type of networked communication device, be it a laptop, desktop computer, PDA, cell phone, or anything else.

[4]This assumes some back-channel communication between the login server and the authentication server.

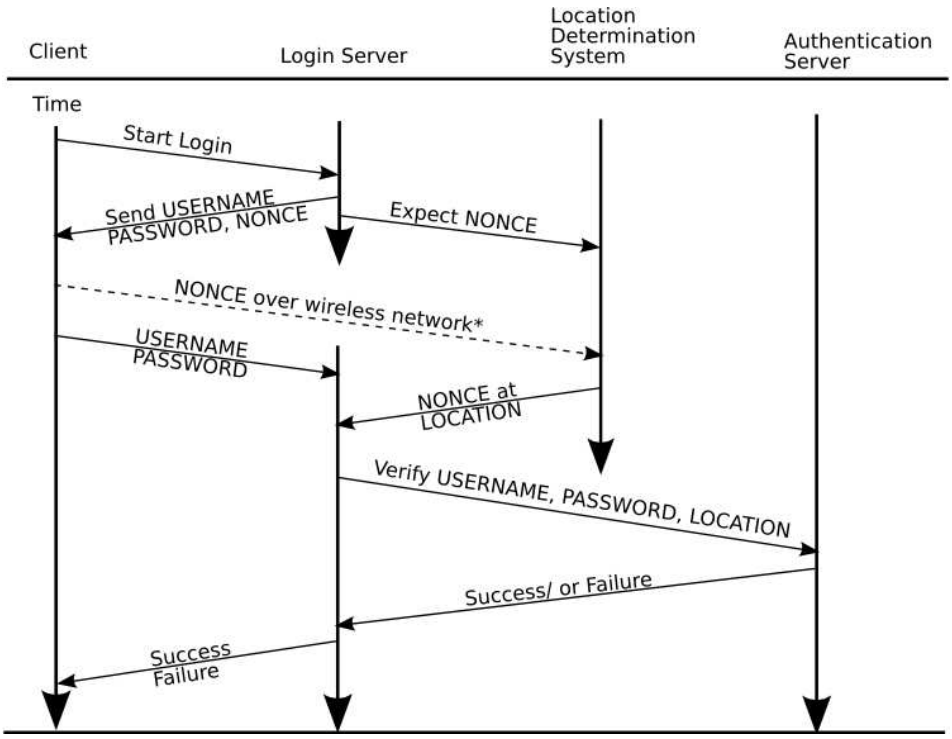


Figure 5.2. **High-Integrity Authentication Process**. This protocol graph describes the communication process in the authentication algorithm. All communication between the login server, location determination system, and authentication server occurs over a private and secure channel. *The client sends the nonce to the Location determination system through the wireless network, not through any other channel.

5.3.3 The Futility of Spoofing

Many protected wireless networks implement a technique called *MAC address filtering* which permits a wireless client to associate with an access point only if the MAC address of the network card used by the client is in an explicitly defined list [3]. A common attack in wireless networks where MAC address filters are in place is for the attacker to change the MAC address of his or her wireless network card. The result is that the attacker's client is indistinguishable from the truly authorized client, from a network perspective. Fortunately for the location-based authentication system, by changing the MAC address to an authorized MAC address the attacker gains nothing. This is because, when the login server issues the challenge the network card will still need to send the response. Since the attacker, even with his or her modified MAC address, is not within the ACA, the attack will fail.

5.3.4 Attacks that Attempt to Obtain Access

Challenge Hijacking

The challenge-response protocol is vulnerable to a time-based man-in-the-middle attack that may allow a sophisticated attacker to be authenticated while not inside the ACA of the user. An attack, as shown in Figure 5.3, could be as follows:

1. The attacker issues a request for authentication from a client outside the ACA.
2. The login server issues a challenge and expects a response over the wireless network.
3. The true user issues a request for authentication from a client inside his ACA. This request is passively intercepted by the attacker by some form of network tap along the communication line between the login server and the client.
4. The attacker hijacks[5] the true user's authentication session. The true user's client is expecting a challenge, and the attacker supplies his received challenge.

[5]Hijacking is a technique whereby the attacker assumes unauthorized control over a communication that he or she is not initially involved in. The attacker assumes the role of one or both sides of the communication by *spoofing* packets. For example, in hijacking a Transmission Control Protocol (TCP) session, the attacker must create packets that have the correct source and destination IP and port

5. The true user's client sends the response to the challenge over the wireless network.

6. The login/authentication system relates the nonce received to the attacker's authentication request. Since the source of the nonce (the true user's client) is inside the proper ACA, the authentication system successfully authenticates the attacker's client.

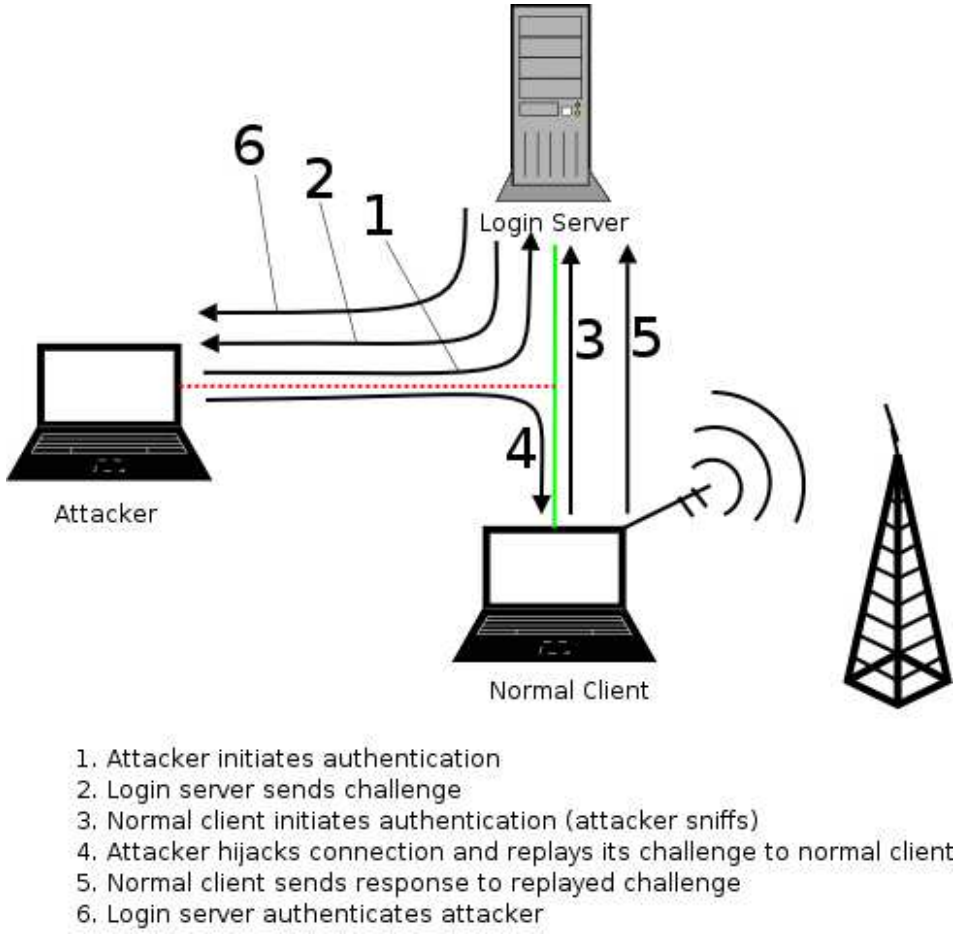


Figure 5.3. **Challenge Hijack Attack**. The attacker can successfully authenticate if he or she has obtained significant privileges.

This attack would result in successful authentication of the attacker. Surely, there are numerous variations on this attack theme, some of which are more easily

pairs (the same as those in the original communication), and the correct sequence and acknowledgement numbers (to match the state of the TCP session). If the attacker assumes the role of both sides of the communication, acts as a relay of packets (with possible modifications) between the two, and neither of the original parties is aware of the hijacking, this is a man-in-the-middle attack [1, pp 252].

executed than others. In order for this attack to be successful, the attacker must have obtained significant capabilities. First, the attacker must be ready to initiate the hijack attempt as the true user is attempting to log on, which requires some ability to anticipate or detect the actions of the true user. Second, he must be able to passively observe traffic between the true client and the login server. Third and finally, he must be able to hijack authentication sessions between the true client and the login server.

Changes to the authentication session protocol may be able to mitigate this form of attack. For example, if in the protocol the client sends the response both over the wireless network and the authentication session medium, and the server terminates authentication if they do not match, the attacker must gain further capabilities to succeed. The attacker needs to be able to actively intercept the challenge and the response over the authentication session medium, preventing these messages from reaching their destination. Physical access to the medium might be required to obtain this capability. Other strategies, for example encryption or protocol changes, are also possible.

Location Deception: Throwing one's Wireless Voice

Location determination systems rely on static radio frequency characteristics in order to work (see section 2.1.2). The system usually builds a model of the signal environment in an offline stage, then refers to the model in the online stage to locate devices. Attackers can try to take advantage of this design in order to deceive the location determination system into believing the device is somewhere it is not (as discussed in Section 2.1.2).

Consider a location determination system that builds a model based upon received signal strength of sample devices and uses interpolation techniques to fill in the blank spaces. For a given location w, the model stores a tuple consisting of the signal strength observations of all access points that received[6] the signal originating from w. If the attacker wishes to make the system believe it is located at position w when it is actually in position v, it may use one or more of several techniques.

[6]Or would have received if real samples were taken instead of using interpolation techniques.

One technique is to vary the output level of the network card, possibly using amplifiers. This could make the device seem as if it is closer to or farther from access points, because there is a functional dependence between signal strength and distance. Another similar technique uses a directional antenna to obtain the desired results. Other techniques include making use of reflection, absorption, and attenuation of signals. Though it may be difficult to form the deception precisely, it is certainly not impossible for a powerful, knowledgeable, and highly determined attacker.

The ability of the attacker to "throw his wireless voice" to another location presents a significant problem in a location-based authentication system. If an attacker can successfully deceive the system such that the system believes the attacker's client is within the ACA of the user when it is not, the integrity of the system has been breached. To thwart these attacks, location determination systems need to be able to handle such malicious types of attacks. Some research groups have already taken a preliminary look at this ([25, 31, 32]).

5.3.5 Attacks that Attempt Denial of Service (DOS)

Numerous denial of service (DOS) attacks are possible if a location determination system is built into the authentication system. However, if a client is found to be behaving badly, as would be the case in the following scenarios, the offending device could be located and disabled, meaning that location determination systems offer some level of security in themselves.

Spoofing Response Packets on the Wireless Network

Consider an authentication system that will de-authenticate a user if two or more challenge responses are received from the user's MAC address over the wireless network. This is a plausible scenario which on the surface seems secure because it prevents attackers from piggybacking on a valid user's authentication session. However, an attacker can take advantage of this by spoofing response packets while valid user's are attempting to authenticate. The valid user will be denied access.

Traditional Wireless DOS techniques

An attacker could deny service to valid users by generating noise, disrupting the wireless network as a whole, or by physically disabling the wireless infrastructure. These forms of denial of service are usually found by the wireless infrastructure, but the location determination system might be able to aid in the process by quickly locating the offending or disabled device.

5.4 Summary

This chapter discussed how a real two-factor authentication system, using passwords and location claims obtained from a wireless location determination system, could be deployed. Such a system needs to be designed properly in order to mitigate possible attacks and prevent unauthorized access to protected resources.

Chapter 6

Conclusion

Deploying the location-based authentication system appears possible using current technologies and research results. But there is some work that would help such systems in terms of integrity and manageability and would also make these systems more appealing as a viable security solution.

6.1 Future Work

Account management applications would need to be developed. The open-source community could easily create PAM modules that support location-based authentication. Developers could also create modules and applications that work with non-UNIX-like platforms. The *pGina* system is a project that allows Windows clients to communicate with PAM servers [57]. These modules and applications are necessary not only for the authentication process but also for account management (for example setting a user's authorized area (AA)). However, since many of today's computer-based maintenance tasks are graphically oriented, graphical front-end applications or plugins should be created. A good solution may be to create a web-based script that acts as an interface to the authentication back-end, a model that seems to be becoming more common as remote systems management appears to be gaining in popularity.

A further area of research involves finding ways to securely verify a user's wire-

less MAC address. Chapter 5.3 described several possible attacks that an adversary may use to successfully authenticate by deceiving the authentication system. Attacks that involve supplying the MAC address of a wireless client that resides within a user's authorized claim area (ACA) are examples of such deceptions. A thorough analysis and design of authentication protocols is required to help minimize the threat of a successful attack.

Finally, the general usability, integrity, and management of location-based authentication systems would all benefit from further research into increasing the precision and accuracy of wireless location determination systems. With higher accuracy, administrators could define more finely specified authorized areas, like offices (improved usability), and reduce the probability of false positives and false negatives (improved integrity), while not having to spend much time fine-tuning the authentication error threshold (improved management). Current research, particularly research using statistical and probabalistic methods, are likely to help in this arena. In addition, more research needs to be done in using location determination for security, which involves improvements to infrastructure-based location estimation, and the handling of heterogeneity and anomylous behavior of wireless devices.

In the near term, test implementations should be developed in order to test and refine the ideas in this thesis.

In summary, research should continue in

1. Secure verification of a client wireless MAC address
2. Design and analysis of secure location-based authentication protocols
3. Analysis of location-based authentication, in terms of usability and security
4. More accurate WLAN location determination systems
5. Using WLAN location determination systems for security purposes

6.2 Wrap-up

Using wireless location estimation as part of two-factor authentication may have potential in real-life environments.

It can be inexpensive. Because the location determination system can be implemented on the same infrastructure as the production wireless network, the only startup costs are associated with training the location determination system (see Section 2.1.2). In some cases (e.g. [32]), even that is not necessary. In the online phase, costs are only associated with account management, which must be done even if two-factor authentiction were not used.

It can be high-integrity. Two-factor authentication provides a means of mitigating the threat of password-based attacks. The authentication and challenge-response protocols can be designed to protect against common network-based attacks such as replay, man-in-the-middle, and spoofing attacks (see Section 5.3). Furthermore, the location determination system can be designed to take into account deception attacks against the location determination system (e.g. [25]). Finally, the organization can control the policy of the authentication system (see Chapter 4) to be more or less restrictive about location claims to handle the fact that location determination systems are not 100% accurate.

This thesis has provided some groundwork in the use of location-based authentication. It described a model for location-based authentication (Section 3.1) and described the conditions under which an ideal location-based authentication system could exist (Section 3.2). It discussed how to use policy mechanisms to handle the fact that current location determination systems have some inherent system error in location claims (Chapter 4). Finally, this thesis explained the architecture and protocol design of a two-factor authentication system using passwords and location (Chapter 5). It showed that such a system could be created and deployed in a real-world environment, and that location-based authentication may have the potential to improve the security of many computing systems.

Appendix A

Free Space Signal Loss Computations

This Appendix describes some free space signal loss computions referenced in Section 2.1.1. The equation describing free-space signal loss is:

$$L_{fsl} = \frac{r^2(4\pi)^2}{\lambda^2} \tag{A.1}$$

where r is the distance between the transmitter and receiver, and λ is the wavelength of the transmission [15]. Expressed in decibels (dB), a logarithmic measure the ratio of two power levels (in this case the signal level and the background noise level) [58], the equation becomes [59]:

$$L_{fsl} = 20 \log \left(\frac{4\pi r}{\lambda} \right) \tag{A.2}$$

When considering the ideal transmission of radio signals through the atmosphere, the wavelength λ is inversely proportional to the frequency [59]:

$$\lambda = \frac{c}{f} \tag{A.3}$$

where the frequency is f and the speed of light in a vacuum is c. The free-space equation

in decibels becomes:

$$L_{fsl} = 20 \log \left(\frac{4f\pi}{c} r \right) \tag{A.4}$$

Using a multiplication property of logarithms, the equation can be expanded:

$$\begin{aligned} L_{fsl} &= 20 \left[\log \left(\frac{4}{c} f\pi \right) + \log(r) \right] && \text{(A.5)} \\ &= 20 \log \left(\frac{4f\pi}{c} \right) + 20 \log(r) && \text{(A.6)} \end{aligned}$$

If we assume a value of $c = 3.0 * 10^8 m/s$, and consider the popular GSM frequency used in the United States ($f = 1.9 * 10^9$ Hz), then the equation simplifies (approximately):

$$\begin{aligned} L_{fsl} &= 20 \log \left(\frac{4f\pi}{c} \right) + 20 \log(r) && \text{(A.7)} \\ &= 20 \log \left(\frac{4 * 1.9 * 10^9 * \pi}{3.0 * 10^8} \right) + 20 \log(r) && \text{(A.8)} \\ &= 38 + 20 \log(r) && \text{(A.9)} \end{aligned}$$

This equation is that in Equation 2.2. In a similar computation for 802.11 networks on the 2.4 Ghz frequency band, the equation simplifies (approximately) to:

$$\begin{aligned} L_{fsl} &= 20 \log \left(\frac{4f\pi}{c} \right) + 20 \log(r) && \text{(A.10)} \\ &= 20 \log \left(\frac{4 * 2.4 * 10^9 * \pi}{3.0 * 10^8} \right) + 20 \log(r) && \text{(A.11)} \\ &= 40 + 20 \log(r) && \text{(A.12)} \end{aligned}$$

which is referenced in [15].

Appendix B

The Example /etc/pam.d/ftpd file

The following constitutes a standard `/etc/pam.d/ftpd` file for ProFTPD on SuSE 6.2, with the addition of location-based authentication. The remainder of the policy is fairly standard, and there are usually few differences (other than filenames) among PAM implementations on different UNIX-like platforms. Refer to PAM documentation [54] for detailed descriptions of the file organization.

```
auth    required       /lib/security/pam_listfile.so item=user sense=deny \
                                file=/etc/ftpusers onerr=succeed
auth    sufficient     /lib/security/pam_ftp.so
auth    required       /lib/security/pam_unix.so
auth    required       /lib/security/pam_location.so
auth    required       /lib/security/pam_shells.so
account required       /lib/security/pam_unix.so
session required       /lib/security/pam_unix.so
```

The `pam_listfile.so` line makes sure that the username supplied is not in the `/etc/ftpusers` file. The `pam_ftp.so` line checks to see if the user is logging in under the anonymous account, and if so whether the account is enabled, in which case login is successful. Anonymous login is generally not allowed when protecting sensitive files. The `pam_unix.so` lines perform standard username and password checking against the `/etc/passwd` and `/etc/shadow` files on the local system. In a larger infrastructure, this

line might be replaced with another module, for example a module to authenticate with a RADIUS server or the Kerberos authentication module. The `pam_location.so` line implements location-based authentication, which is the primary focus of this thesis. The `pam_shells.so` file makes sure that the shell specified in the `/etc/passwd` file for the user is a valid shell (i.e., is found in `/etc/shells`).

Bibliography

[1] M. Bishop, *Computer Security: Art and Science*. Boston, MA: Addison-Wesley, 2003.

[2] J. T. L.L.C., "Default Passwords List," 2005. [Online]. Available: http://www.jacksontechnical.com/article.htm?id=1

[3] M. S. Gast, *802.11 Wireless Networks: The Definitive Guide, 2nd Edition*. Sebastopol, CA: O'Reilly Media, Inc., 2005.

[4] "Internet bot," Wikipedia, 2005. [Online]. Available: http://en.wikipedia.org/wiki/Internet_Bot

[5] S. M. Kerner, "Weak Passwords Leave Win MySQL Vulnerable," 2005. [Online]. Available: http://www.internetnews.com/dev-news/article.php/3465791

[6] "CERT/CC Current Activity Archive," CERT Coordination Center, 2005. [Online]. Available: http://www.cert.org/current/archive/2005/02/25/archive.html#MySQLUDF

[7] "Current Infosec News and Analysis," The SANS Institute, 2005. [Online]. Available: http://isc.sans.org/diary.php?date=2005-01-27

[8] "How Radar Works," HowStuffWorks, Inc., 2005. [Online]. Available: http://electronics.howstuffworks.com/radar.htm

[9] "Orienteering," Wikipedia, 2005. [Online]. Available: http://en.wikipedia.org/wiki/Orienteering

[10] M. Alouini, "Global Positioning System: An Overview," Tunisian Scientific Magazine. [Online]. Available: http://citeseer.ist.psu.edu/31114.html

[11] "Differential gps," Wikipedia, 2005. [Online]. Available: http://en.wikipedia.org/wiki/DGPS

[12] "Wide area augmentation system," Wikipedia, 2005. [Online]. Available: http://en.wikipedia.org/wiki/WAAS

[13] "GSM Frequenty Asked Questions," WirelessGalaxy.com. [Online]. Available: http://www.wirelessgalaxy.com/mobilephones/gsmworldphonefaqs.html

[14] "How Cell Phones Work," HowStuffWorks, Inc., 2005. [Online]. Available: http://electronics.howstuffworks.com/cell-phone5.htm

[15] "RF Propagation Basics," Sputnik, Inc., Apr. 2004.

[16] "Cellular network," Wikipedia, 2005. [Online]. Available: http://en.wikipedia.org/wiki/Cellular_network

[17] R. I. Reza, "Data Fusion for Improved TOA/TDOA Position Determination in Wireless Systems," Masters of Science in Electrical Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA, July 2000.

[18] P. Bahl and V. N. Padmanabhan, "RADAR: An In-Building RF-based User Location and Tracking System," *IEEE Infocom 2000*, vol. 2, pp. 775–784, Mar. 2000.

[19] P. Bahl, V. N. Padmanabhan, and A. Balachandran, "Enhancements to the RADAR User Location and Tracking System," Microsoft Corporation, Redmond, WA, Tech. Rep. MSR-TR-2000-12, Feb. 2000.

[20] A. Smailagic, D. P. Siewiorek, J. Anhalt, D. Kogan, and Y. Wang, "Location Sensing and Privacy in a Context Aware Computing Environment," *Pervasive Computing*, 2001.

[21] P. Castro, P. Chiu, T. Kremenek, and R. Muntz, "A Probabilistic Room Location Service for Wireless Networked Environments," in *Proc. of Ubicomp*, Sept. 2001, pp. 18–24.

[22] R. Battiti, T. L. Nhat, and A. Villani, "Location-aware Computing: a Neural Network Model for Determining Location in Wireless LANs," University of Trento, Trento, Italy, Tech. Rep. DIT-02-0083, 2002.

[23] R. Battiti, M. Brunato, and A. Villani, "Statistical Learning Theory for Location Fingerprinting in Wireless LANs," University of Trento, Trento, Italy, Tech. Rep. DIT-02-0086, 2002.

[24] M. Wallbaum, "Wheremops: An Indoor Geolocation System," *IEEE International Sumposium on Personal, Indoor, and Mobile Radio Communications*, Sept. 2002.

[25] P. Tao, A. Rudys, A. M. Ladd, and D. S. Wallach, "Wireless LAN Location-Sensing for Security Applications," *Wireless Security Workshop*, 2003.

[26] M. Youssef, A. Agrawala, and A. Shankar, "WLAN Location Determination via Clustering and Probability Distributions," *IEEE International Conference on Pervasive Computing and Communications*, Mar. 2003.

[27] A. Howard, S. Siddiqi, and G. S. Sukhatme, "An Experimental Study of Localization Using Wireless Ethernet," *International Conference on Field and Service Robotics (FSR 03)*, 2003.

[28] Y. Gwon, R. Jain, and T. Kawahara, "Robust Indoor Location Estimation of Stationary and Mobile Users," *IEEE Infocom*, Mar. 2004.

[29] T. Roos, P. Myllymaki, and H. Tirri, "A Statistical Modeling Approach to Location Estimation," *IEEE Transaction on Mobile Computing*, vol. 1, pp. 59–69, Jan. 2002.

[30] A. M. Ladd, K. E. Bekris, A. Rudys, G. Marceau, L. E. Kavraki, and D. S. Wallach, "Robotics-Based Location Sensing using Wireless Ethernet," *The Eighth ACM International Conference on Mobile Computing and Networking (MOBICOM)*, Sept. 2002.

[31] S. Ganu, A. S. Krishnakumar, and P. Krishnan, "Infrastructure-Based Location Estimation in WLAN Networks," *IEEE Wireless Communications and Networking Conference (WCNC 2004)*, 2004.

[32] P. Krishnan, A. Krishnakumar, W. Ju, C. Mallows, and S. Ganu, "A System for LEASE: Location Estimation Assisted by Stationary Emitters for Indoor RF Wireless Networks," *IEEE Infocom*, Mar. 2004.

[33] "Soekris Engineering," Soekris Engineering, 2005. [Online]. Available: http://www.soekris.com

[34] "NYCwireless," NYCwireless, 2005. [Online]. Available: http://www.nycwireless.net/pebble

[35] "LinuxAP Wireless Access Point," LinuxAP, 2005. [Online]. Available: http://linuxap.ksmith.com

[36] "Hostap driver for intersil prism2/2.5/3 wireless lan cards and wpa supplicant," Jouni Malinen, 2005. [Online]. Available: http://hostap.epitest.fi

[37] J. Small, A. Smailagic, and D. P. Siewiorek, "Determining User Location For Context Aware Computing Through the Use of a Wireless LAN Infrastructure," Dec. 2000.

[38] C. K. M. Brunato, "Transparent Location Fingerprinting for Wireless Services," in *Proc. of Med-Hoc-Net*, 2002.

[39] M. Youssef and A. Agrawala, "On the Optimality of WLAN Location Determination Systems," *Communication Netwoks and Distributed Systems Modeling and Simulation Conference*, Jan. 2004.

[40] D. Nisculescu and B. Nath, "Ad Hoc Positioning System (APS) using AoA," *INFOCOM 2003*, 2003.

[41] N. Patwari, A. O. H. III, M. Perkins, N. S. Correal, and R. J. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Transactions on Signal Processing, Special Issue on Signal Processing in Networks*, Nov. 2002.

[42] L. Evers, W. Bach, D. Dam, M. Jonker, H. Scholten, and P. Havinga, "An iterative quality-based localization algorithm for ad hoc networks," *Pervasive Computing 2002*, Aug. 2002.

[43] "Moustafa a. youssef's web page," University of Maryland, 2005. [Online]. Available: http://www.cs.umd.edu/ moustafa

[44] "Newbury Networks Wireless LAN Security and Intrusion Prevention," 2004. [Online]. Available: http://www.newburynetworks.com

[45] "Ekahau - The Most Accurate Wi-Fi Positioning," 2005. [Online]. Available: http://wwwekahau.com

[46] "AeroScout - Homepage," 2004. [Online]. Available: http://www.aeroscout.com

[47] D. E. Denning and P. F. MacDoran, "Location-Based Authentication: Grounding Cyberspace for Better Security," *Computer Fraud and Security*, Feb. 1996.

[48] E. Gabber and A. Wool, "How to Prove Where You Are: Tracking the Location of Customer Equipment," in *CCS '98: Proceedings of the 5th ACM conference on Computer and communications security*, 1998, pp. 142–149.

[49] T. Kindberg and K. Zhang, "Context Authentication using Constrained Channels," Hewlett-Packard Labs, Tech. Rep. HPL-2001-84, 2001.

[50] N. Sastry, U. Shankar, and D. Wagner, "Secure Verification of Location Claims," in *WiSe '03: Proceedings of the 2003 ACM workshop on Wireless security*, 2003, pp. 1–10.

[51] B. Waters and E. Felton, "Proving the Location of Tamper-Resistent Devices." [Online]. Available: http://www.cs.princeton.edu/ bwaters/research

[52] "Half-Normal Distribution - from MathWorld," Holfram Research, Inc., 2005. [Online]. Available: http://mathworld.wolfram.com/Half-NormalDistribution.html

[53] "Riemann Sum - from MathWorld," Holfram Research, Inc., 2005. [Online]. Available: http://mathworld.wolfram.com/RiemannSum.html

[54] "Linux-PAM - Pluggable Authentiction Modules for Linux," Kernel.org, 2000. [Online]. Available: http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html

[55] "PPP Extensible Authentication Protocol (EAP)," Merrit Network, Inc., 1998. [Online]. Available: http://www.ietf.org/rfc/rfc2284.txt

[56] "What are EAP, LEAP, PEAP and EAP-TLS and EAP-TTLS?" Free Internet Media, 2005. [Online]. Available: http://www.tech-faq.com/eap-leap-peap-eap-tls-ttls.shtml

[57] "pGina: Making the big boys play nice," XPA Systems, 2004. [Online]. Available: http://pgina.xpasystems.com

[58] "Decibel," Wikipedia, 2005. [Online]. Available: http://en.wikipedia.org/wiki/Decibel

[59] "How far will my radio transmit?" Radio Innovation, 2003. [Online]. Available: http://www.radioinnovation.com/Howto/how_far.htm

This page intentionally left blank.

Wireless Network Device Fingerprinting

IEEE 802.11 Physical Layer Radio Frequency (RF) Analysis

Project Team:

Jason Franklin
Frank Hemingway
Damon McCoy
Kristen Pelon
Amanda Stephano
Parisa Tabriz

Sandia National Laboratories
Livermore, CA

Center for Cyber Defenders
Summer 2005

Project Mentors:

Steve Hurd
Ratish Punnoose,
Jamie Van Randwyk,

Introduction

There are two paths for remotely fingerprinting wireless network devices. One path is analyzing the contents or timing of wireless frames at level 2 of the OSI networking model (the data link layer). The other path is analyzing actual electromagnetic radiation at level 1 of the OSI model (the physical layer). Both paths examine information leakage that can potentially fingerprint devices—that is characterize devices with a unique signature. This report focuses on the second path by documenting the early efforts to RF Fingerprint wireless network devices at Sandia.

Overview

The process of RF fingerprinting is not formulaic; instead, there is much room to experiment. We examine the signal from different wireless cards and look for distinguishing patterns. The goal is to determine what part of the signal is unique from one card to the next and determine how that difference should be analyzed. Most likely, the distinguishing features will be found by visually inspecting a plot of the signal. These differences can then be quantified mathematically by correlation and other statistical/probababilistic methods. The plots examined to date relate the following information:

1.) power vs. time;
2.) phase vs. time; and
3.) frequency vs. time

It remains to be determined which plot(s) will be most useful, but the power plot is the easiest way to identify frame transmissions because a signal's power spikes dramatically when frames are transmitted (Figure 1). Fingerprinting devices based on their inter-frame signal may still be possible, but sensitive equipment would be needed to detect the low signal levels.

Cabling

We obtain a clean voltage signal via cable from each wireless card under test. The cable is one of three types, depending upon what type of card is tested.

Card Type	Connection on Card	Cable Used
PCI	Reverse SMA socket (male pin)	Reverse Polarity SMA --> SMA Male
PCMCIA	Standard MMCX female	MMCX Male right --> SMA Male (gold)
MiniPCI	U.FL (IPAX) socket	U.FL (IPAX) plug --> SMA Male

The following links provide useful pictures for identifying each type of connection:

http://www.seattlewireless.net/index.cgi/ConnectorsAndCable
http://www.solwise.co.uk/wireless-cable.htm

Like the termination of our custom cables, the lead from the signal analyzer is also an SMA male connection. To accommodate this incompatibility, we used an SMA female-female adapter to make the connection. (Thus, if more cables are ordered, they should terminate in standard SMA female as opposed to standard SMA male. The U.FL (IPAX) cable already terminates in standard SMA female because the manufacturer simply added a standard SMA male-male adapter to the end.

Data Collection

We test the cards one at a time on the same computer to minimize variables. We use a Windows computer because it has the greatest amount of driver availability to support various cards. Each driver is downloaded from the manufacturer's website and installed ahead of time. The driver install files are located on the desktop of the Windows PC in case they need to be reinstalled. Because there are multiple wireless drivers installed on the same machine the user should be careful to avoid software conflict.

The data collection process proceeds as follows: The first card is installed into the CPU; the appropriate cable is attached from the card to the signal analyzer system; and the CPU is booted up. The signal analyzer cannot accommodate the bandwidth of all 11 channels, so we must choose a specific channel. If the cards are tested without the presence of an access point or without an ad hoc network, then the person conducting the test should set the card to only operate on a single channel—otherwise the card will scan the various channels, looking for an access point, and it will go out of range of the signal analyzer. Or, if multiple stations on an ad hoc network or a station and an access point are to be observed on their own network, then the devices will stay on the channel of that network.

To set a lone card to operate on a single channel, open the network connection preference in the Windows control panel and view the properties of the wireless connection. Click on the “Configure” button and go to the “Advanced” tab. This tab shows the user-adjustable options for the card. If it is possible to select an operating channel for the card, select channel 1. Channel 1 is chosen somewhat arbitrarily, but the selected channel should correspond to the capture frequency of the signal analyzer (e.g. 2.412 GHz is the center frequency for channel 1). Each channel is 22 MHz wide. Not all wireless card drivers allow the user the ability to stay on a single channel. The Senao PCMCIA card and the Linksys PCI card are two cards tested that are known to accommodate this feature (shown in the last row of the table in the *Cards Tested* section of this report).

The signal analyzer should be triggered to start recording on the rising edge of the signal. The operator should right click on the Wireless Network Connection icon in the Windows Control Panel and enable the connection. The signal analyzer records the data, and the user saves the data file to the hard disk. The data is then exported to a .mat file, where it can be loaded as a Matlab struct. The contents of a sample one-second RF trace struct are shown below:

```
SampleStruct =
```

```
        InputZoom: 1
      InputCenter: 2.4120e+09
       InputRange: 0.1000
    InputRefImped: 50
           XStart: 0
           XDelta: 2.1701e-08
          XDomain: 2
            XUnit: 'Sec'
            YUnit: 'V'
     FreqValidMax: 2.4306e+09
     FreqValidMin: 2.3934e+09
                Y: [46080081x1 single]
```

The data points are stored in the variable 'Y', which is a one-dimensional matrix containing the complex data points recorded from the signal analyzer. The rest of the struct contains useful information pertaining to the data set—information such as frequency range recorded, the sample rate (1/XDelta), and the units pertaining to the data. One second of complex data with double precision real and imaginary components entails 46 million data points and a size of about 360 MB. If we want to examine the first four data points of these 46 million we would type the following line at the Matlab command prompt.

```
>> SampleStruct.Y(1:4)

ans =

  -0.0003 + 0.0002i
  -0.0056 + 0.0031i
  -0.0059 + 0.0074i
  -0.0045 + 0.0090i
```

Working with a smaller amount of data at a time improves application performance and stays within memory limitations. Reducing the length of Y to 1000 data points to study a specific part of the signal, such as in Figure 2, Figure 3, and Figure 4 reduces the struct to a size of tens of kilobytes.

Traversing the Data

Managing the large amount of data is challenging. Finding the relevant parts of the signals is directly dependent upon one's ability to traverse the data. Matlab can be scripted to accomplish much of this searching, and built-in tools from the Signal Processing Toolbox help. Also, triggering the signal analyzer to record on the rising edge of a power signal can help locate the beginning of a frame transmission. Visual searching of the data through plots is limited by the number of points that can be displayed on a computer screen at one time (monitor resolution) as well as by the memory limitations of Matlab and the processor speed of the computer running it. If the observer zooms too far out, then too many pixels will be lost and the transmitted frame might be missed altogether. Also, if too much of the data is incorporated into a single plot, then the application response time is slow. Figure 1 shows the first results of encompassing entire frame transmissions into a single plot. Its domain is 1 million data points.

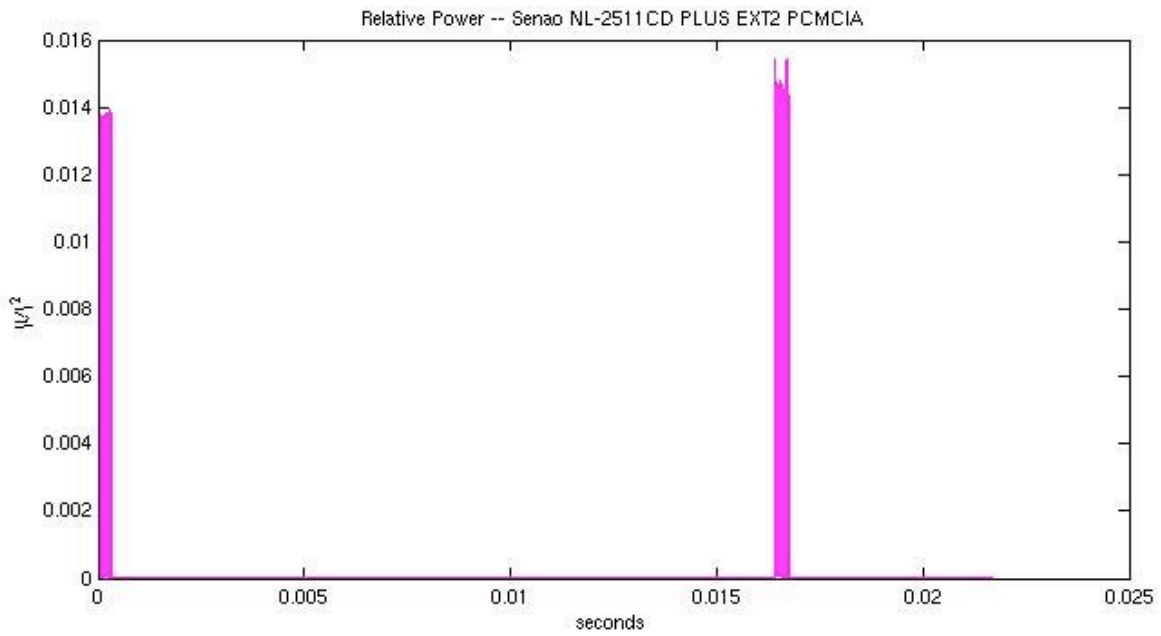


Figure 1 Frame transmissions are easy to locate in the relative power plot when a large time scale is used. The entire 22 millisecond span of the signal corresponds to 1 million sampled data points. The plot suggests about 17 milliseconds between these two frame transmissions.

The cards are tested without the presence of an access point in order to simplify the initial setup and data collection and to control variables. We can therefore deduce that these frames must be probe request frames from the unassociated card. (It would be interesting to see if different frame types—e.g. management, association, probe request/response—have distinguishable signal properties.) Monitoring stations in the presence of an access point or monitoring multiple stations configured in an ad hoc network would allow us to observe two-way communications and increase the quantity of frame transmissions. There would be much less empty space between frames.

Cards Tested

	Card 1	Card 2	Card 3	Card 4
Make:	Senao	Senao	Linksys	Netgear
Form:	PCMCIA	Mini PCI	PCI	PCI
Model:	NL-2511 CD PLUS EXT 2	NL-2511MP PLUS	WMP11	WG311T
Mac Address:	00:02:6F: 08:21:14	00:02:6F: 09:A4:7F	(not displayed on card)	00:0F:B5: 25:59:32
Serial Number:	03A174978	03C241609	BB7272946697	WG7514AAG020 404
FCC ID:	NI3-2511CD-	(not displayed) on	PKW-WMP11-	PY3WG311T2

	Card 1	Card 2	Card 3	Card 4
	PLUS3	card)	V27	
IEEE 802.11_	b	b	b	g
Can Set Channel ?	Yes	No	Yes	No

Graphs of RF Data

The IEEE 802.11b/g wireless network data travels on a sinusoidal carrier frequency at 2.4 GHz on the electromagnetic spectrum. The carrier frequency, itself, does not constitute the data being transmitted; instead, a second sinusoidal whose frequency varies across a range of 22 MHz of bandwidth is modulated and carries actual data for a single IEEE 802.11b/g channel. This second, lower frequency signal is the part that we examine in the graphs in the next section.

A general form of a sinusoidal voltage signal is $v(t) = A\cos(\omega t + \phi)$. The three parameters A, ϕ, and ω correspond to the amplitude, phase, and radial frequency. Any sinusoidal voltage can be represented as a function of time with knowledge of these three parameters. Sometimes there is a constant term added to the end, but it is usually neglected, and in the case of measuring the potential difference (voltage) between two signals, it cancels anyway.

Any of these three parameters (A, ϕ, ω) are fair game for fingerprinting wireless network devices because the behavior of any one of them (or any combination) may leak distinguishing information about the device. These 3 parameters constitute the signal. Each of the following 3 sections of this report corresponds to one of these parameters. Four plots (one for each tested card) comprise each section, followed by a brief discussion.

It is not currently known if one parameter or plot type is more useful for fingerprinting RF devices than the others. As previously shown in Figure 1, the power of the signal is the most intuitive way to identify frame transmissions because the frames transmit at high power. However, if we match this plot in time with, say, a phase plot, then we might determine that the phase plot has a unique, identifiable behavior during frame transmission as well. Finding the frames is important because the high-power part of the signal is the easiest to detect in a real-world setup, and not having to rely on the power plot to detect frames mean we can easily look for distinguishing frame features in the phase plot, too.

Relative Power

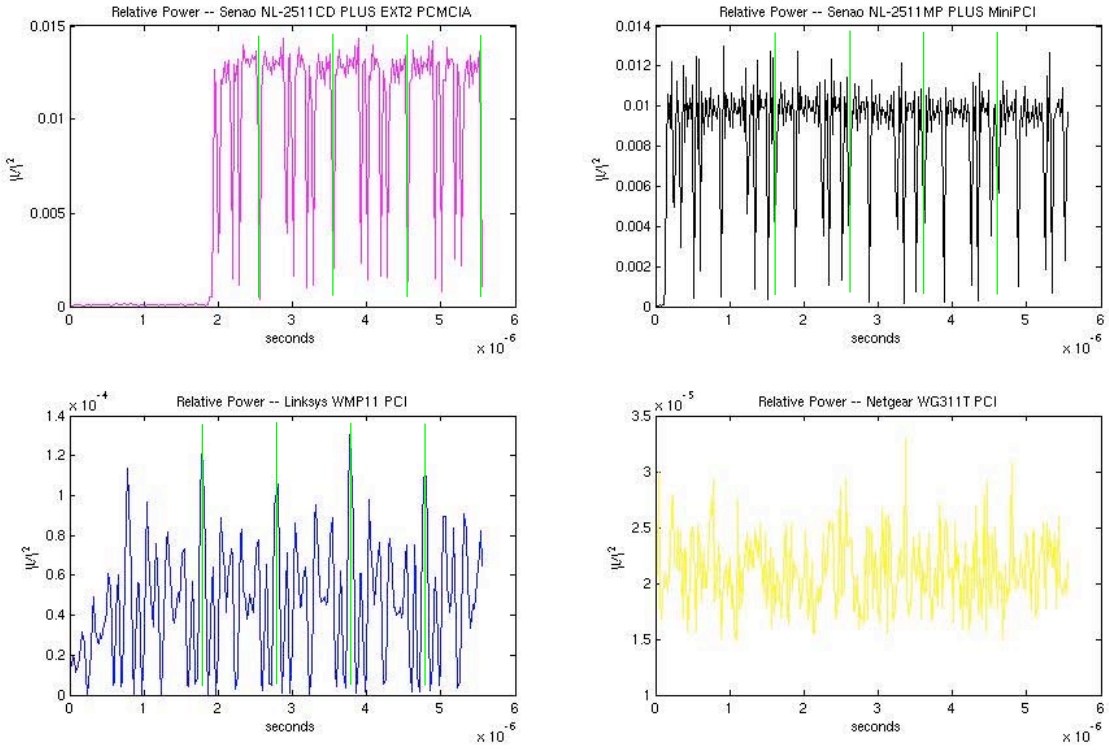


Figure 2: Relative power plots show periodic behavior during frame transmission for at least cards 1-3. The periods are approximately 1 microsecond. Vertical green lines are superimposed to help the viewer identify a single period. The plot of Card1 appears to catch the rising edge of the frame. Card1–Pink Card2–Black Card3–Blue Card4–Yellow

Power is an important parameter that stems from signal amplitude. Specifically, the power of a current or voltage signal is proportional to the square of its amplitude. Power is the more useful interpretation of the amplitude, which can be measured from the physical signal. The signal power level can also be gleaned from management frames where it recorded in dBm. This value is how applications such as NetStumbler relate signal power information to the user. One intra-frame power parameter is the *link margin*, which indicates the excess power used in transmitting the signal from one wireless device to another. Access points and stations transmit this parameter to manage power broadcast levels. The object is to send signals that are strong enough to be received by the intended device but not so strong that they cause unnecessary interference with other devices.

Studying the relationship between the measured signal power and the management frame recorded power may be useful for fingerprinting wireless network devices. A consistent difference between the two values may be unique to certain devices. Signal power is definitely an interesting link between the physical layer and the data link layer of the 802.11 protocol.

The term *relative power* is used for the vertical axis of the plots in Figure 2 (as opposed to simply

power) to indicate that the power is scaled. Standard transmission lines, including those used in this project, terminate with 50 Ohms of resistance. To convert relative power, $|V|^2$, to power measured in Watts, simply divide by 50. (Remember to also account for any attenuators in the transmission line setup.) The reason for neglecting the scaling factor in the plots is that power (as well as signal gain) are often measured in decibels over 1 Watt (dB), or decibels over 1 milliwatt (dBm). The power can also be measured relative to an arbitrary reference signal. In any of these cases, a ratio of the power is taken, and the resistances cancel.

It should be noted from the vertical scales in Figure 2 that cards 1 and 2 have power of the same order of magnitude. These are the Senao cards. Card 1 is PCMCIA and card 2 is Mini PCI, but they have nearly the same power levels—they likely use the same chipset. Cards 3 and 4 have power levels that are 2 and 3 orders of magnitude less[*]. Signal power levels may thus be useful for fingerprinting the devices based on manufacturer. Obviously, a 30 mW Orinoco card cannot produce a signal with a power of 90 mW, but a 100 mW Cisco card can.

A useful observation from Figure 2 is that the signals for the first 3 cards are periodic. This observation means that the variance in the signal is a function of the wireless device under test and not just random noise added to the signal. The periodic signal may correspond to the steady-state behavior of the circuitry inside the wireless card's chipset—or at least the circuitry immediately surrounding the card's antenna port. If this hypothesis is true, then the cyclic pattern may be a way to fingerprint the hardware of the device.

The signal from card 3 is not apparently periodic. It is the only IEEE 802.11g card of the bunch. (The others are 'b'.) More testing should be done on other 'g' cards to determine if this result is recurring. The phase plot that follows also shows a unique result for the plot of card 4, the 'g' card.

[*] Several orders of magnitude seems like a bit much—this result should be verified. Perhaps the signal analyzer did not trigger on a frame transmission. Figure 1 verifies that Card1 triggered on a frame; the same can be done for Cards 2-4.

Phase

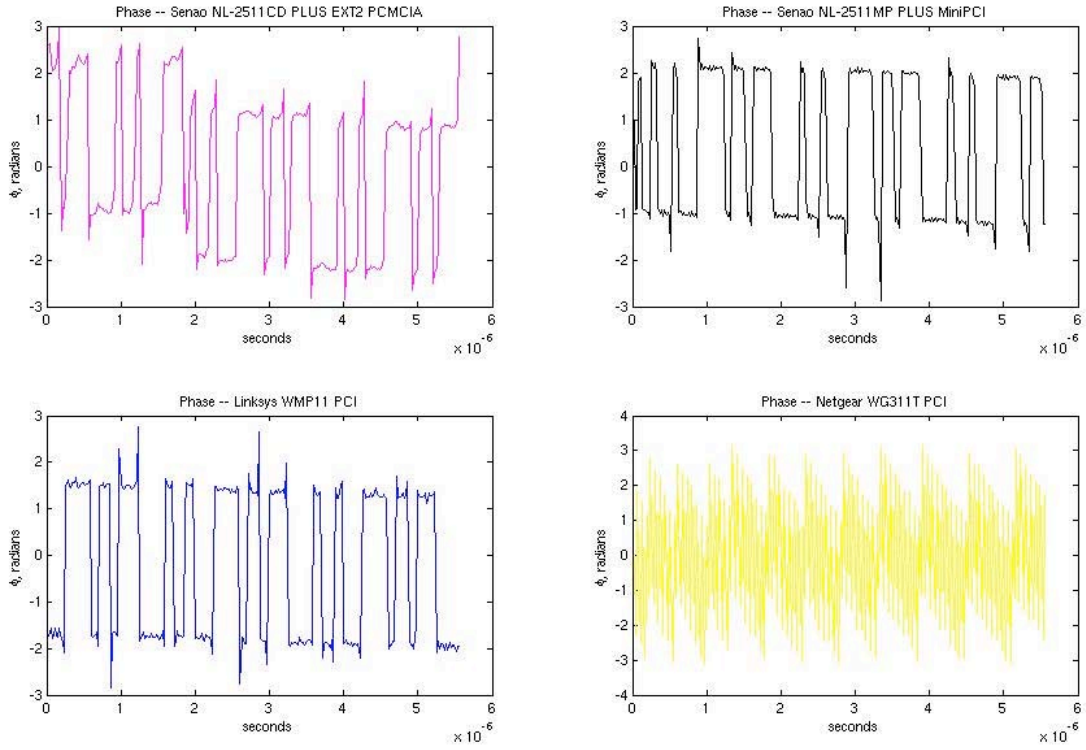


Figure 3 Plotting the phase of each signal versus time reveals tendency toward the discrete values that correspond to the $\pi/2$ radian phase shifts of the 802.11 standard's quadrature phase shift keying. The overshoots are reasonable features to investigate for fingerprints. Card1–Pink Card2–Black Card3–Blue Card4–Yellow

Because the data collected by the signal analyzer is complex, each sampled value can be mapped to both a magnitude and a phase. Figure 2 depicts the magnitude of the signal (actually the square of the magnitude). Figure 3 depicts the phase. Obtaining the phase is as simple as taking the arctangent of the imaginary part divided by the real part of the complex number—except that a 4-quadrant arctangent function is used. Instead of using Matlab's 'atan' function to return a principle angle between $-\pi/2$ and $\pi/2$, the 'atan2' function returns a value in the range $-\pi$ to π. This step is important for retaining information. Better still, the 'angle' function computes the ratio of imaginary to real parts, with the 4-quadrant arctangent, all in one step.

The IEEE 802.11 standard is phase modulated. Using quadrature phase shift keying (QPSK), the signal phase is shifted in increments of $\pi/2$ radians (out of a standard 2π phase period). By this method, two bits can be represented by each phase shift. Plotting the phase of the signal versus time is a step toward demodulation. Not surprisingly, this results in square-wave-type plots with plateaus at various levels corresponding to the discrete phase shifts. Several trials should be done and the results compared to see if there are uniquely identifiable features for a device given its phase plot. Not all of the plateaus are at exactly $\pi/2$ increments of phase shifts, and there is visible overshoot at many of the plateaus.

Frequency

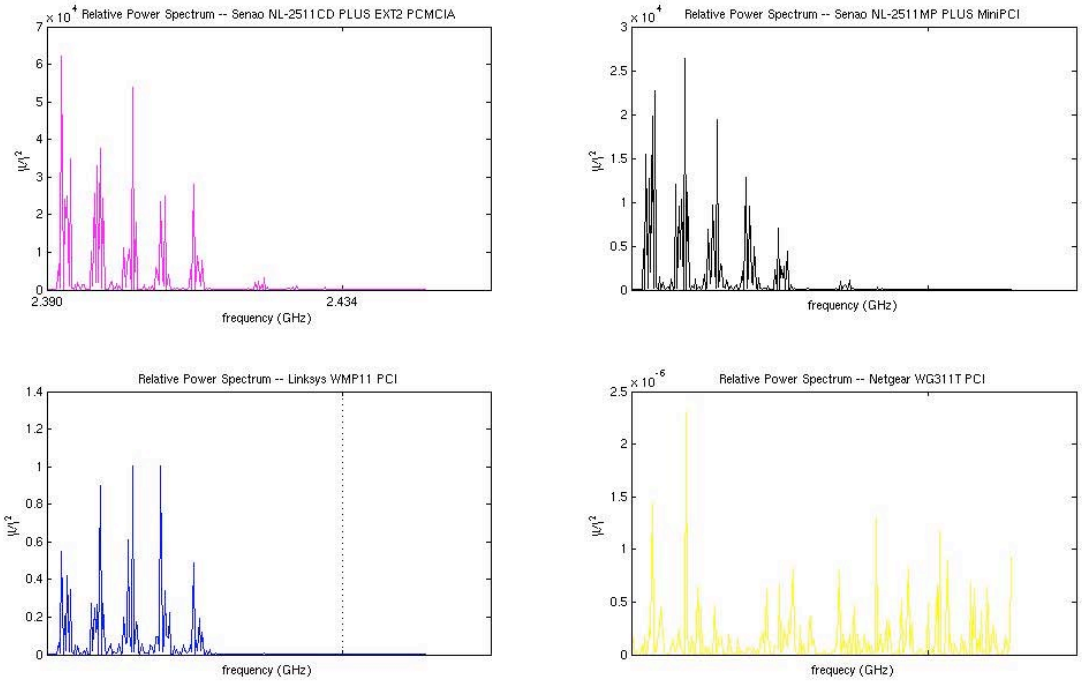


Figure 4 The Fourier transform of the relative power vs. time shows the relative power levels at the frequencies that comprise the wireless network transmissions. The 802.11b cards (Cards1-3) show power usage at 5 discrete frequencies. The 802.11g card (Card4) uses a more complicated modulation scheme that utilizes a broader range of frequencies to accomplish its higher throughput. Card1–Pink Card2–Black Card3–Blue Card4–Yellow

The magnitude and the phase plots in Figure 2 and Figure 3, taken together, are enough information to fully describe each signal. The frequency domain, however, is another useful interpretation of the signal. Figure 4 shows the power spectrum for each of the wireless cards as a function of the signals cyclic frequency, f, which is related to ω by a factor of 2π. To get these plots, the Fourier transform of the complex time-domain data is taken, and then the square of its magnitude is plotted.

IEEE 802.11b uses direct sequence spread spectrum modulation. The existence of the 5 discrete bands is consistent with such a modulation scheme where transmissions are spread across a width of the frequency spectrum. From these plots we can see if certain wireless card models use more or less power at certain frequencies. We can also check for the width and center frequencies of each band.

The power spectra of the different cards also verify that the cards transmit on frequencies specified by the IEEE 802.11 standard. The power spectra can check devices for transmission on unauthorized frequencies that could suggest a covert channel leaking potentially sensitive information from a user's computer. The reason for awareness of this issue is that many wireless devices are manufactured in foreign countries and then exported to the United States.

Future Work

Verifying reproducibility of measurements is very important and should be the next step in continuing research of RF-based fingerprinting. The initial results are promising because the plots for each card model are visibly different. These results should be verified to ensure that signal differences among cards are caused by actual card signatures and not by other variables. The same cards (the actual device with same serial number) should be retested for reproducibility, as well as different models of the same card (devices with the same model number but different serial numbers). Testing in such a manner will determine if a signature can be accurately traced to a particular brand or model of card.

With the exception of Figure 1, the plots focus on only the first 6 microseconds of the signal. This interval represents the first 256 data points. The beginning of the signal is chosen as an arbitrary (but logical) starting point to establish a method for collecting and analyzing data. Future work should investigate longer portions of the signal and the frames contained therein. During frame transmission, we expect the broadcast power to remain approximately constant, however, small nuances in the transmission could be unique to a certain model of card. Such nuances might be a gradual rate of power decrease/increase during frame transmission or power overshoot/undershoot.

A good way to obtain more frames in a shorter signal interval is to connect multiple devices to the mixer and have them communicate on their own infrastructure or ad hoc network. This method adds some extra variables, such as timing, to a card's behavior, but the RF features of the frames should be unaffected.

Card 4 is an 802.11g card, and its plots are observably different from those of the other cards. It is not known if the differences are caused by the difference in the modulation of the ‘g’ card’s signal or perhaps by the card trying to skip over the various channels to find an access point and thereby going out of range of the signal analyzer. (The 11 carrier frequency channels for the ‘b’ and ‘g’ standards are the same, so this is not a problem.) In short, the data from its plots might be valid or not. Additional testing would tell.

Conclusion

This project has proven at least one conclusive result: Not all models of wireless cards behave the same—even if they boast the same set of “standards” and are tested in a controlled environment. Speculation of this principle was grounds for this project, and this project has verified that result. We have not proven that different wireless cards of the same model do behave the same. (But they likely do.) We have also not proven that a single, exact same device behaves the same at different times. (But it probably does.) What needs to be done is to verify the “likely” and the “probably” from the preceding statements by reproducing the current test results for the same exact devices as well as for new devices of the same models. If the “likely” and the “probably” hold, then the theoretical basis for RF fingerprinting of wireless cards is established, and there is a good chance of being able to fingerprint wireless devices based upon their RF signatures.

The RF fingerprinting project provides results that merit deeper investigation. We have been able to

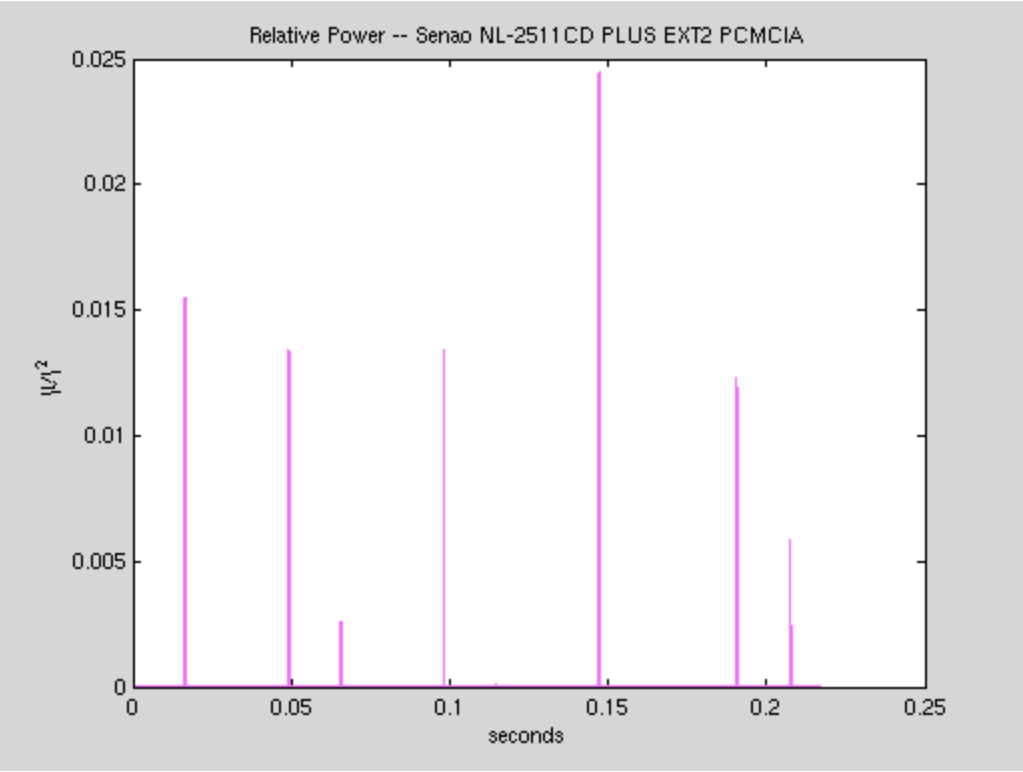
successfully access and interpret data transmissions at the physical layer. This project establishes a method for collecting data and establishes methods for interpreting the data in terms of power, phase, and frequency.

The observation that power and phase plots are visibly distinguishable among various cards is a reassuring result. If this result is consistent and reproducible among card models, then what remains for a wireless device fingerprint recognition system is a fingerprint storage, retrieval, and comparison mechanism. Of course, there must also be a way to extract the RF signal from the air in a real-world environment.

In addition to contributions toward development of a wireless network device fingerprinting system, the knowledge gathered through researching this project is, itself, valuable. We benefit from a better understanding of the signal upon which wireless network data is transmitted. Understanding of this project's fundamental tools and concepts is important for addressing current and ever-changing security threats that may emerge from—or be resolved at—the physical layer.

```
%This m-file plots NUMPOINTS points of the Relative power of an IEEE 802.11
%card.  With an order of 100,000s, NUMPOINTS plots enough data points
%to encompass several frames.

%The following variable must be loaded into the workspace:
%Test005_PC_1sec_Purple.mat

%For convenience, shorten name to var5, that is
var5 = load('Test005_PC_1sec_Purple.mat');

NUMPOINTS = 10000000;
X = var5.XDelta .* [1:NUMPOINTS];

%Plot relative power

figure(1)

plot(X,(abs(var5.Y(1:NUMPOINTS))).^2)
title('Relative Power -- Senao NL-2511CD PLUS EXT2 PCMCIA')
xlabel('seconds')
ylabel('|{\itV}|^2')
%Find the Blue (Matlab default) line [R G B], and make it purple.
set(findobj(gca,'Type','line','Color',[0 0 1]),'Color',[1 0 1],'LineWidth',1)

hold;
```

Current plot released



Published with MATLAB® 7.0

```
%This m-file plots the Relative power of 4 802.11 client cards

%The following variables must be loaded into the workspace:
%tst5.mat
%tst6.mat
%tst7.mat
%tst8.mat

% Copy structs as 1-D lists
% Remember, z is complex
z5 = tst5.Y;
z6 = tst6.Y;
z7 = tst7.Y;
z8 = tst8.Y;
t5 = (tst5.XDelta)*[1:1024];
t6 = (tst6.XDelta)*[1:1024];
t7 = (tst7.XDelta)*[1:1024];
t8 = (tst8.XDelta)*[1:1024];
% Now get magnitudes
mag5 = abs(z5);
mag6 = abs(z6);
mag7 = abs(z7);
mag8 = abs(z8);

%Plot relative power
figure(1)
subplot(2,2,1); plot(t5(1:256),mag5(1:256).^2)
title('Relative Power -- Senao NL-2511CD PLUS EXT2 PCMCIA')
xlabel('seconds')
ylabel('|{\itV}|^2')
set(findobj(gca,'Type','line','Color',[0 0 1]),'Color',[1 0 1],'LineWidth',1)

subplot(2,2,2); plot(t6(1:256),mag6(1:256).^2)
title('Relative Power -- Senao NL-2511MP PLUS MiniPCI')
xlabel('seconds')
ylabel('|{\itV}|^2')
set(findobj(gca,'Type','line','Color',[0 0 1]),'Color',[0 0 0],'LineWidth',1)

subplot(2,2,3); plot(t7(1:256),mag7(1:256).^2)
title('Relative Power -- Linksys WMP11 PCI')
xlabel('seconds')
ylabel('|{\itV}|^2')
set(findobj(gca,'Type','line','Color',[0 0 1]),'Color',[0 0 1],'LineWidth',1)

subplot(2,2,4); plot(t8(1:256),mag8(1:256).^2)
title('Relative Power -- Netgear WG311T PCI')
xlabel('seconds')
ylabel('|{\itV}|^2')
set(findobj(gca,'Type','line','Color',[0 0 1]),'Color',[1 1 0],'LineWidth',1)

hold
```

```
Current plot held
```

```
%This m-file plots the phase of 4 802.11 client cards
% b is the number of points (from the beginning) to be plotted
b = 256
%The following variables must be loaded into the workspace:
%tst5.mat
%tst6.mat
%tst7.mat
%tst8.mat

% Copy structs as 1-D lists
% Remember, z is complex
z5 = tst5.Y;
z6 = tst6.Y;
z7 = tst7.Y;
z8 = tst8.Y;
t5 = (tst5.XDelta)*[1:b];
t6 = (tst6.XDelta)*[1:b];
t7 = (tst7.XDelta)*[1:b];
t8 = (tst8.XDelta)*[1:b];

% Now get phase
% angle gets 4-quadrant-defined phase angle
% angle(z) = imag(log(z)) = atan2(imag(z),real(z))
phi5 = angle(z5(1:b));
phi6 = angle(z6(1:b));
phi7 = angle(z7(1:b));
phi8 = angle(z8(1:b));

%Plot phase
figure(2)

subplot(2,2,1); plot(t5(1:b),phi5)
title('Phase -- Senao NL-2511CD PLUS EXT2 PCMCIA')
xlabel('seconds')
ylabel('{\it\phi}, radians')
set(findobj(gca,'Type','line','Color',[0 0 1]),'Color',[1 0 1],'LineWidth',1)

subplot(2,2,2); plot(t6(1:b),phi6)
title('Phase -- Senao NL-2511MP PLUS MiniPCI')
xlabel('seconds')
ylabel('{\it\phi}, radians')
set(findobj(gca,'Type','line','Color',[0 0 1]),'Color',[0 0 0],'LineWidth',1)

subplot(2,2,3); plot(t7(1:b),phi7)
title('Phase -- Linksys WMP11 PCI')
xlabel('seconds')
ylabel('{\it\phi}, radians')
set(findobj(gca,'Type','line','Color',[0 0 1]),'Color',[0 0 1],'LineWidth',1)

subplot(2,2,4); plot(t8(1:b),phi8)
title('Phase -- Netgear WG311T PCI')
xlabel('seconds')
ylabel('{\it\phi}, radians')
set(findobj(gca,'Type','line','Color',[0 0 1]),'Color',[1 1 0],'LineWidth',1)

hold
```

```
b =

   256

Current plot held
```

```
%This m-file plots the frequency-domain relative power of 4 802.11 client cards
% b is the number of points (from the beginning) to be plotted
b = 256
%The following variables must be loaded into the workspace:
%tst5.mat
%tst6.mat
%tst7.mat
%tst8.mat

% Copy structs as 1-D lists
% Remember, z is complex
z5 = tst5.Y;
z6 = tst6.Y;
z7 = tst7.Y;
z8 = tst8.Y;

% Now transform to frequency domain using fft
Z5 = fft(z5);
Z6 = fft(z6);
Z7 = fft(z7);
Z8 = fft(z8);

% Get magnitudes
Mag5 = abs(Z5);
Mag6 = abs(Z6);
Mag7 = abs(Z7);
Mag8 = abs(Z8);

% Square the voltage magnitude
Pwr5 = Mag5.^2;
Pwr6 = Mag6.^2;
Pwr7 = Mag7.^2;
Pwr8 = Mag8.^2;

%f = 46065966*(1:b)

%Plot relative power spectra
figure(3)

subplot(2,2,1); plot(Pwr5(1:b).^2)
title('Relative Power Spectrum -- Senao NL-2511CD PLUS EXT2 PCMCIA')
xlabel('frequency')
ylabel('|{\itV}|^2')
set(findobj(gca,'Type','line','Color',[0 0 1]),'Color',[1 0 1],'LineWidth',1)

subplot(2,2,2); plot(Pwr6(1:b).^2)
title('Relative Power Spectrum -- Senao NL-2511MP PLUS MiniPCI')
xlabel('frequency')
ylabel('|{\itV}|^2')
set(findobj(gca,'Type','line','Color',[0 0 1]),'Color',[0 0 0],'LineWidth',1)

subplot(2,2,3); plot(Pwr7(1:b).^2)
title('Relative Power Spectrum -- Linksys WMP11 PCI')
xlabel('frequency')
ylabel('|{\itV}|^2')
set(findobj(gca,'Type','line','Color',[0 0 1]),'Color',[0 0 1],'LineWidth',1)

subplot(2,2,4); plot(Pwr8(1:b).^2)
title('Relative Power Spectrum -- Netgear WG311T PCI')
xlabel('frequecy')
ylabel('|{\itV}|^2')
set(findobj(gca,'Type','line','Color',[0 0 1]),'Color',[1 1 0],'LineWidth',1)

hold
```

```
b =

   256

Current plot held
```

VI. Wireless Fingerprinting

Abstract

As more devices include 802.11 wireless capabilities, the security of wireless devices drivers emerges as an important issue. If attackers can identify which wireless device driver is being used, they can better target their attacks. A tool that could identify potentially vulnerable 802.11 devices drivers and notify the device or prevent the device from connecting to a network could provide value by encouraging users to update wireless drivers. In this paper, we design and evaluate a new method for passively fingerprinting 802.11 wireless device drivers. This method is tolerant of packet loss and works in the presence of other wireless cards and traffic. We show our method for passive fingerprinting is extremely accurate in identifying the wireless driver running on a device.

Introduction

The number of devices that are capable of wireless communication is growing at a high rate. This makes them an emerging target for attackers using devices that employ wireless device driver exploits. Robert Lemos of SecurityFocus argues that drivers are particularly vulnerable [1]. In particular, he notes that while security is not a concern for most drivers because an attacker would require physical access to these devices, there are several exceptions, including Bluetooth and both wired and wireless network interface drivers. Further research by Swift et al. [2] reports that 85% of recent failures in Windows XP are caused by driver failures and [3] found that driver code has up to 3 to 7 times more errors then other Linux kernel code. A tool that could identify and quarantine computers with potentially vulnerable drivers would be useful.

In this paper, we design and evaluate a new method for passively fingerprinting 802.11 wireless device drivers. We designed our method to be purely passive, but our method could be used in a hybrid passive/active fingerprinting method.

Section two of this paper presents our method for wireless device fingerprinting and discusses how it works. In section three, we present our experimental data and evaluate how well the fingerprinting method performs. Section four discusses related work. In section five, possible future work is introduced, and section six presents our conclusions.

Passive Fingerprinting Method

While monitoring wireless traffic via Ethereal, we noticed that the wireless cards sent out periodic bursts of probe request frames. We graphed delta arrival times of the probe requests vs. frame sequence numbers as shown in Figure 18. This revealed a structured signature for each driver. We collected data over time intervals of more than one hour to make sure that the pattern seen was consistent.

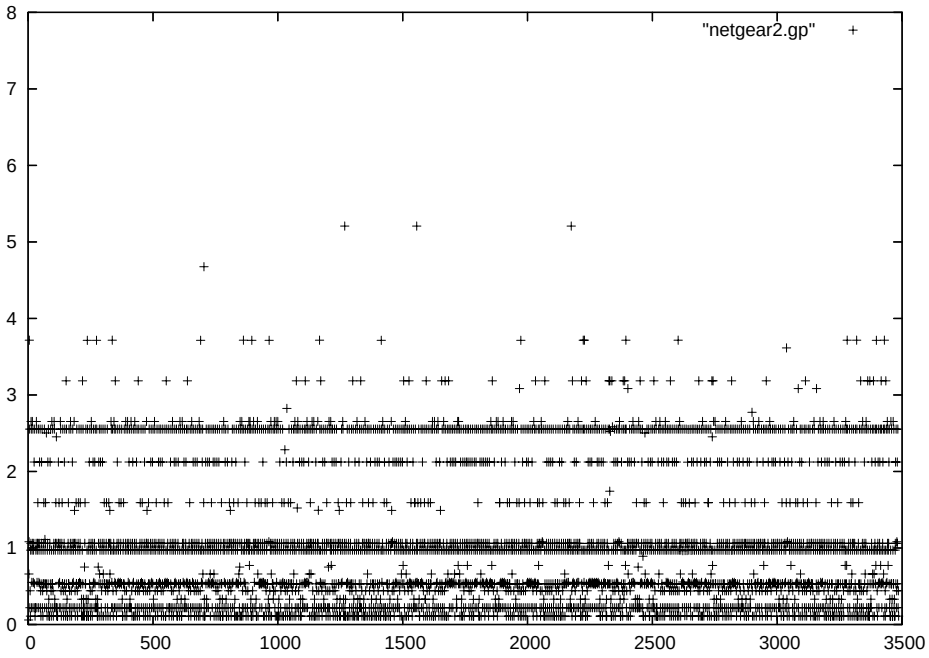


Figure 18. Plot of delta time vs. sequence number collected over a one-hour time period. This shows that probe requests for the Windows XP Netgear WG311t driver have a well-defined pattern.

Fingerprinting Algorithm Details

We then explored ways of automating the driver fingerprinting process using both statistical and machine learning tools. We considered clustering algorithms, but then we settled for a more generic statistical approach of placing the data points into bins and calculating proportions and means for each bin. The reasoning behind this is that we wanted a method that would run in "real time." The computation overhead of a clustering algorithm would have slowed down the method. We empirically tuned the bin width to 2-second-wide bins. Future work is needed to determine the optimal bin width for accuracy.

We built signatures for each driver of our 13 wireless network interface cards (NICs), generating the bin proportion and bin means. A simple individual driver signature is shown in Table 1. A master file was created to store all the driver signatures.

Table 1. Sample signature

Bin	Proportion	Mean
0	0.617	0.472
2	0.363	1.907
4	0.020	3.355

Test data from an unknown driver can be compared to all the signatures. We came up with a statistical equation to measure the "closeness" of the test data to each signature.

The equation we decided on after empirical testing is Equation 1:

$$C = \min(S \in s \forall \sum_{0}^{n} \left(\left| T_{pn} - s_{pn} \right| + s_{pn} \left| T_{mn} - s_{mn} \right| \right) \quad (1)$$

The lowest *C* value is the signature that most closely matches the test data. *S* is the set of signatures in the master signature file. *T* is the test data from the wireless card being fingerprinted. The summation occurs over all the bins in the test data with a non-zero proportion. This equation has the property that if the signature has a zero proportion for a bin, then the mean difference part of the summation is not factored into the equation for that bin.

Evaluation

To evaluate the suitability of our fingerprinting algorithm and isolate the factors that affect the signature of a driver, we ran a number of experiments. The first set of experiments was designed to pinpoint the variables that cause drivers to give off different signatures. After we discovered most of the features that change signatures, we built a master signature file containing all the different signatures produced by the cards used in this paper. We focused on Windows XP, Linux kernel 2.6, and Mac OS X 10.3 for our operating systems.

Isolating Variables Affecting Driver Signatures

The most obvious factor that affected the signature of a wireless driver was whether the card was associated to an access point (AP) or remained unassociated. Most cards have two distinct signatures, one when associated and another when unassociated as seen in Figures 19 and 20 respectively.

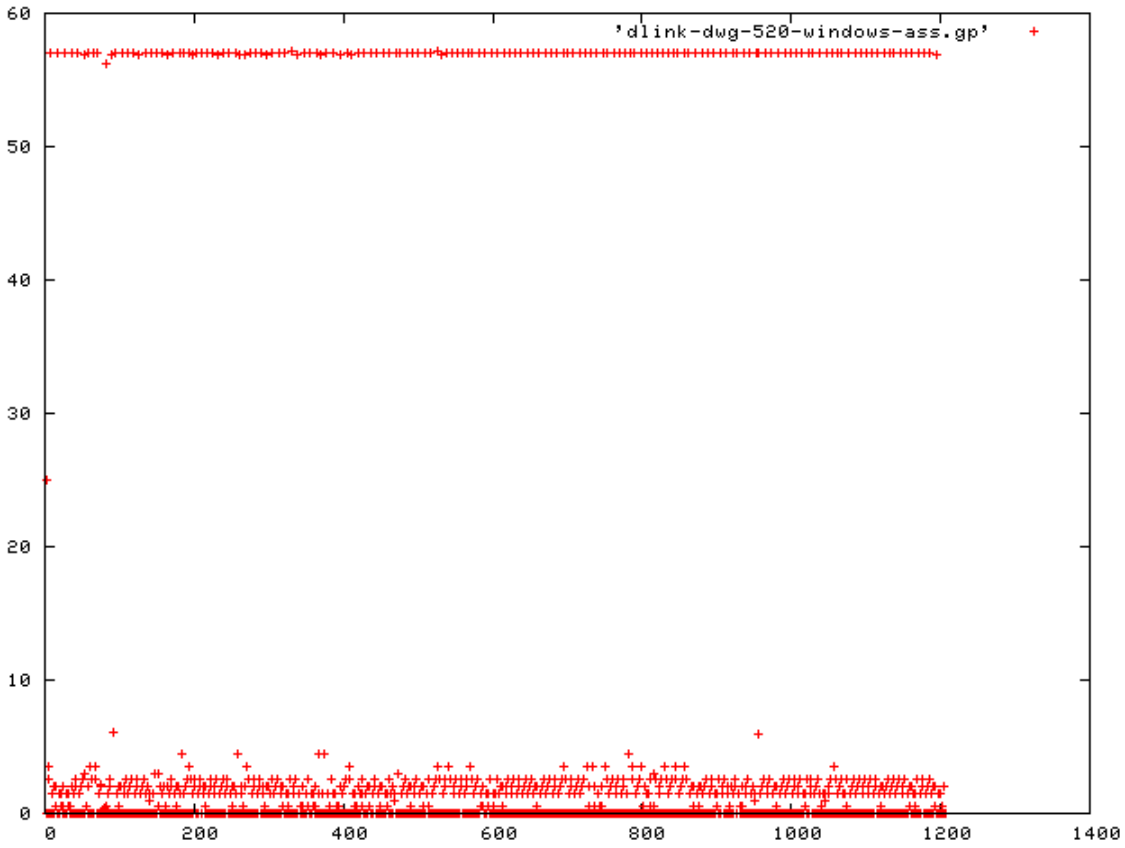


Figure 19. Delta time vs. sequence number for D-link dwg-520g associated.

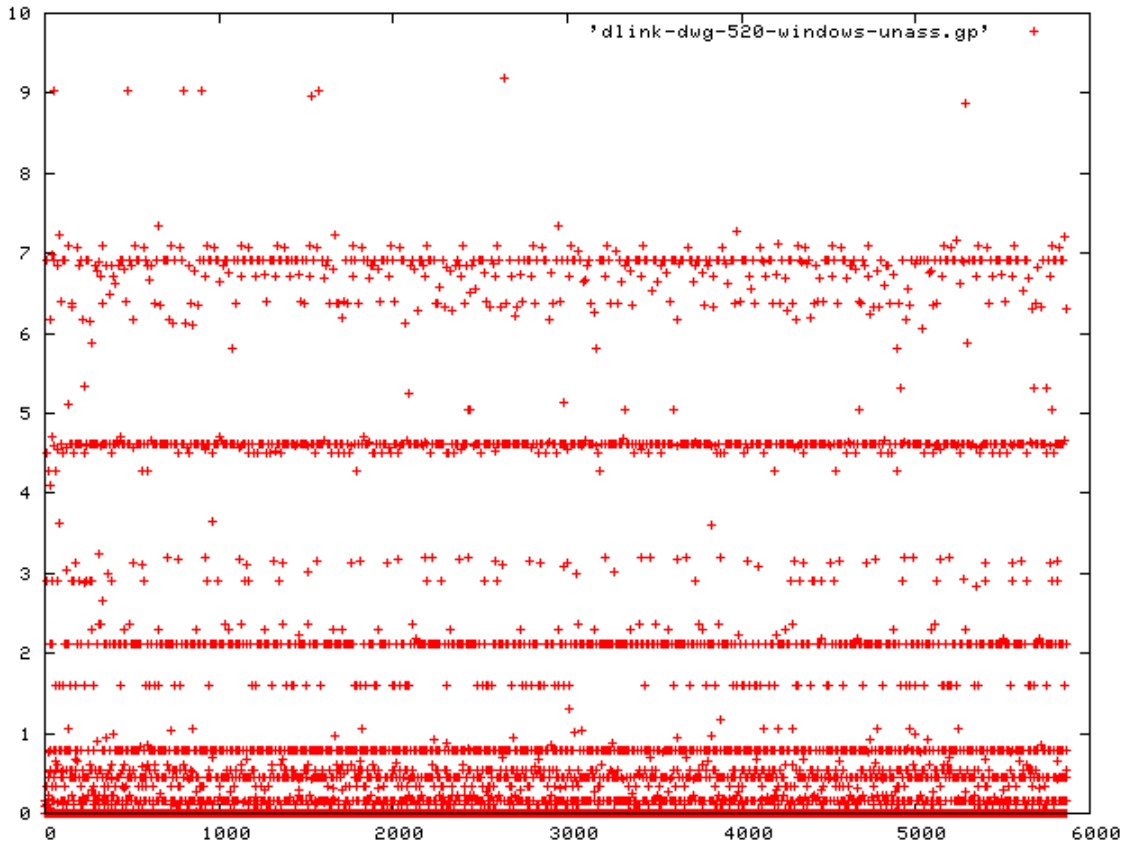


Figure 20. Delta time vs. sequence number for D-link dwg-520g unassociated.

Our second set of tests aimed to see if the actual card had any effect on the signature or if the driver was solely responsible for the signature. To perform this test, we used the Linux madwifi driver, which has a hardware abstraction layer (HAL) that allows the driver to work with any Atheros-based wireless card. We conducted experiments using different cards and the same madwifi driver. The signature for all cards using the madwifi drivers was identical, as shown in Figures 21 and 22. We also ran experiments using the same card and different drivers under Linux and Windows. When run under different drivers, the same card produced different signatures, leading us to believe the signature is tied to the driver. This behavior is displayed in Figures 22 and 23.

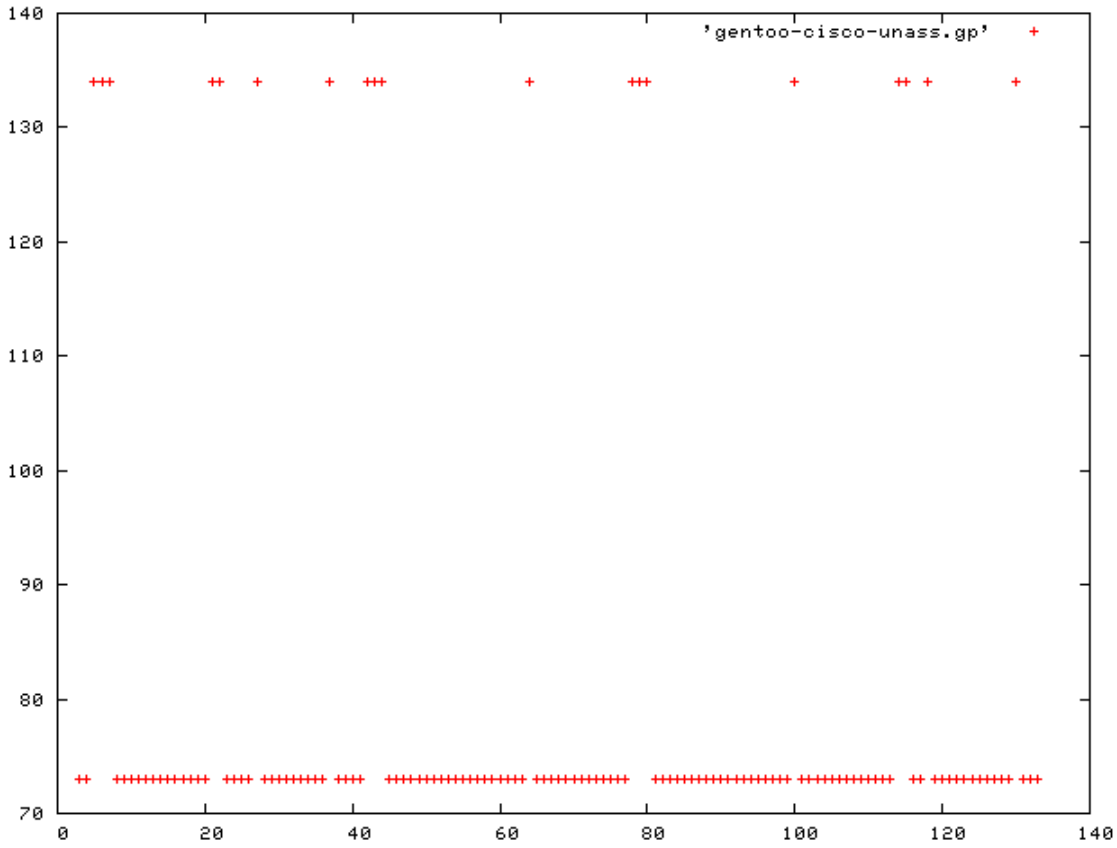


Figure 21. Delta time vs. sequence number for Cisco Aironet a/b/g pci running madwifi driver unassociated.

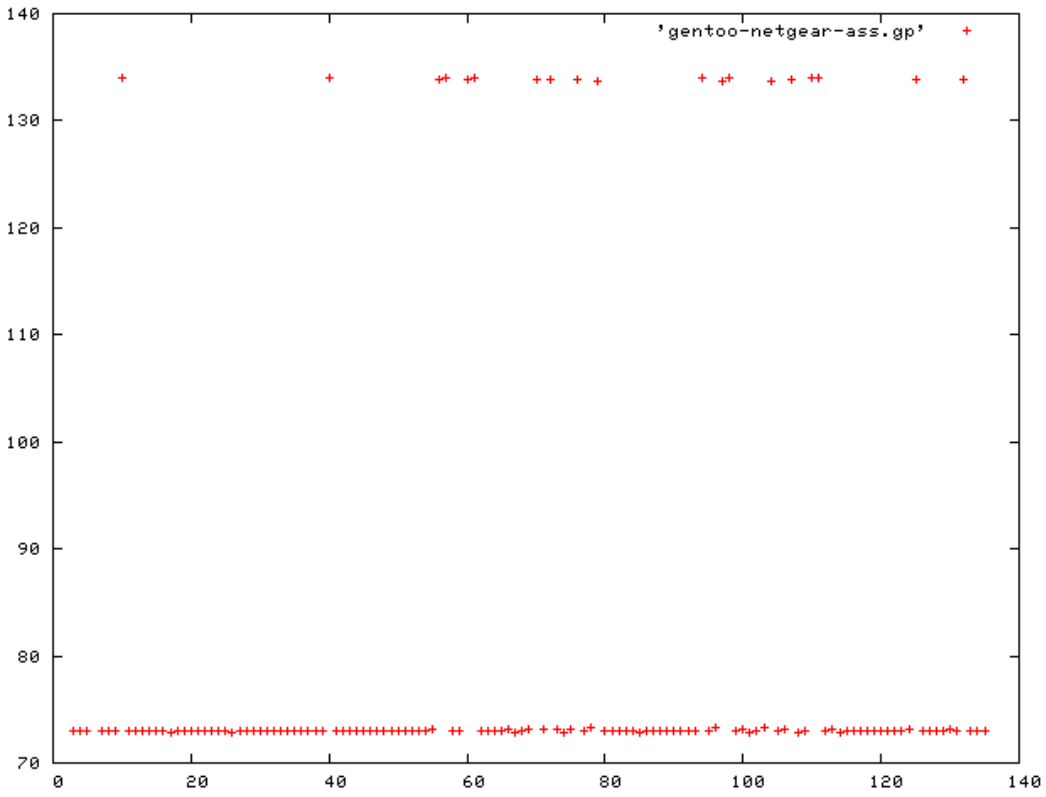


Figure 22. Delta time vs. sequence number for Netgear WG311t running madwifi driver unassociated.

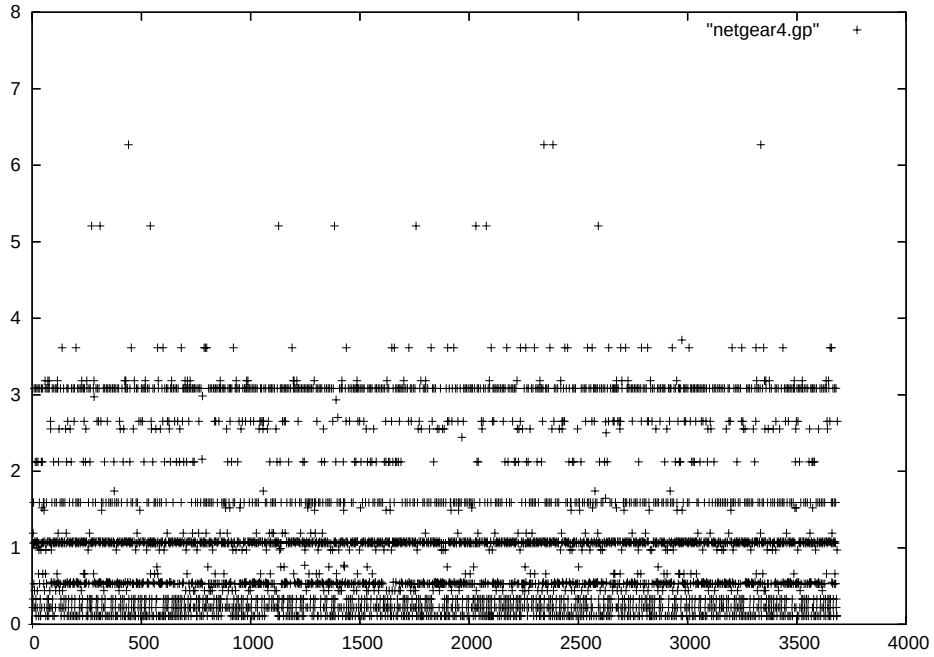


Figure 23. Delta time vs. sequence number for Netgear WG311t Windows XP driver unassociated.

We then explored the effect of several features of Windows XP on the signature. First, Windows XP has two different service pack (SP) levels: SP1 and SP2. After collecting data from a number of different cards using the same driver but a different SP level, we determined that SP level did not affect the signature. The next factor we investigated was the difference between letting Windows XP control the wireless card versus the vendor-provided proprietary tool. To switch between configuration modes, we used the check box labeled “Use Windows to configure the wireless network settings” on the “Wireless Networks” tab in the “Network Properties” window. Changing this option did indeed produce a different signature, as shown in Figures 24 and 25.

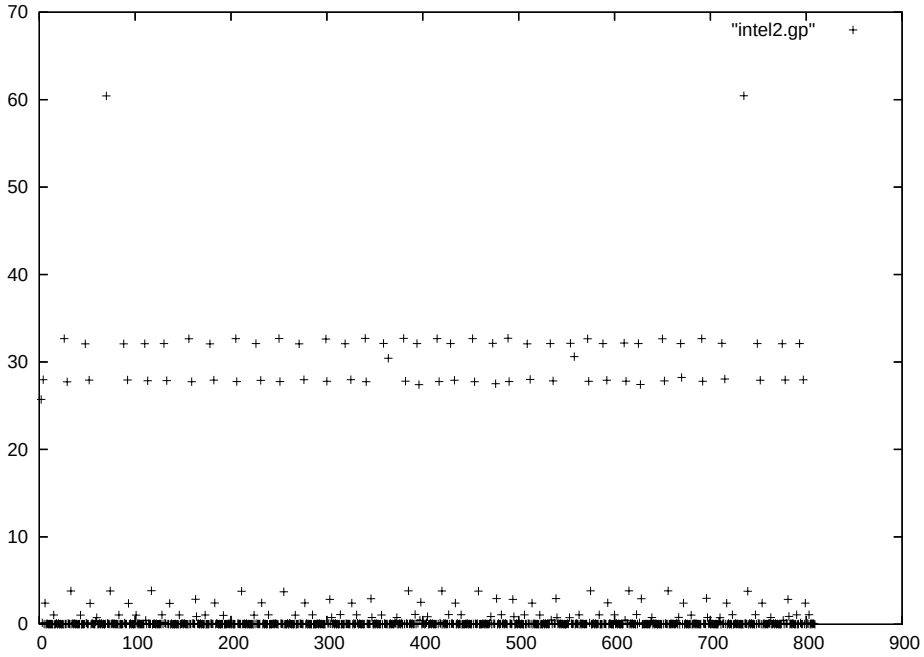


Figure 24. Delta time vs. sequence number for Intel2200 Wireless card using Windows XP to control the card.

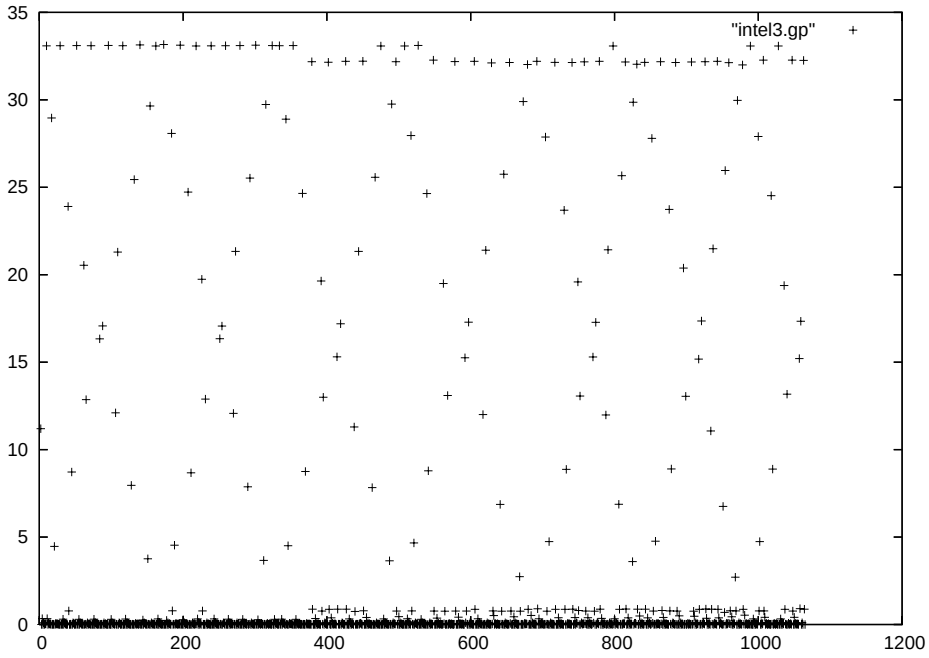


Figure 25. Delta time vs. Sequence number for Intel2200 wireless card using proprietary Intel tool to control the card.

Finally, the presence of other traffic could affect a signature. In our test, three extra unassociated cards that were sending out their own probe packets did not affect a driver's signature. We also ran tests with a light amount of traffic and did not notice any effect on the signatures produced.

Evaluation of Fingerprinting Algorithm

The algorithm preformed remarkably well, achieving a high degree of accuracy in identifying the correct signature from the 13 different wireless cards in the various configurations. Some cards did not generate probe packets in different configurations; those cases were not included in the results since a signature could not be produced.

Previous Work

Nmap has the capability to fingerprint a device's operating system based on active fingerprinting of the TCP/IP stack. P0f implements a passive method for fingerprinting operating systems, and Yoshe Kohn has also conducted research on actively fingerprinting physical devices based on TCP time stamps and clock skew.

Future Work

Active methods for layer 2 fingerprinting were attempted using the libwlan library. Our methods were unsuccessful because the cards did not have any distinguishing characteristics to the responses, so we decided to focus on passive methods. Future work could focus on active methods to gain better fingerprinting granularity to the driver version level. In addition, better exploration of the signature creation method and statistical fingerprinting algorithm could improve the accuracy of the overall method.

There are some potential flaws in our evaluation of the fingerprinting method that should be explored further. The first potential flaw is that the number of Service Set Identifiers (SSIDs) a NIC is configured to look for could affect the card's signature. The signature could also depend on whether the card has been associated in the past or not. Some probe packets are generic, while others have specific SSID strings for APs that have been seen before. It may be useful for the signature to incorporate the ratio between generic probes and probes with SSID strings in them. Finally, having more traffic may affect signatures because of the Network Allocation Vector (NAV), which is used for avoiding collisions.

Conclusions

Passive wireless fingerprinting appears to be doable, although we have not performed repeatability experiments for all variables. We may not even be aware of all significant variables.

References

[1] R. Lemos, "Device Drivers Filled with Flaws," *The Register*, May 27, 2005. [Online]. Available: http://www.theregister.co.uk/2005/05/27/device_driver_flaws/

[2] M. Swift, B. Bershand, and H. Levy, "Improving the Reliability of Commodity Operating Systems," in *19th ACM Symposium on Operating Systems Principles*, 2003. [Online]. Available: http://www.cs.rochester.edu/sosp2003/papers/p116-swift.pdf

[3] A. Chou, J. Yang, B. Chelf, S. Hallem, and D. Engler, "An Empirical Study of Operating System Errors," in *18th ACM Symposium on Operating Systems Principles*, 2001. [Online]. Available: http://sosp.org/2001/papers/chou.pdf

VII. Appendices

A. OpenSource Wireless Security Tools

B. Developer Notes for Sessionlogger

C. Sessionlogger manpage

D. Sniffer manpage

This page intentionally left blank.

Open Source Wireless Security Tools

Many open source tools have been written in recent years for testing, breaking, and defending wireless networks. We provide a short list below with descriptions of those tools. More information about these tools and others can be found at www.wardrive.net/wardriving/tools. Many good commercial tools have been developed as well, but we do not discuss or evaluate those here.

There are many tools that may be used for analyzing or attacking wireless networks. Network Sniffers are useful to people listening to network traffic. Examples include:

- NetStumbler (www.netstumbler.com) is a freeware wireless access point identifier that listens for SSIDs and sends beacons as probes searching for access points.
- Kismet (www.kismetwireless.net) is a freeware wireless sniffer and monitor that passively monitors wireless traffic and sorts data to identify SSIDs, MAC addresses, channels and connection speeds.
- Wellenreiter (www.wellenreiter.net/) is a freeware WLAN discovery tool that uses brute force to identify low traffic access points, hides your real MAC address, and integrates with GPS.
- THC-RUT (www.thehackerschoice.com) is a freeware WLAN discovery tool that uses brute force to identify low traffic access points.
- Ethereal (www.ethereal.com) is a freeware WLAN (and LAN) analyzer. The user can interactively browse and capture data, viewing summary and detail information for all observed wireless traffic.

There are also tools that can make it easy to deceive people trying to use wireless networks. Examples include:

- HostAP (hostap.epitest.fi) converts a WLAN station to function as an access point. It is available for WLAN cards that are based on Intersil's Prism2/2.5/3 chipset.
- AirSnarf (airsnarf.shmoo.com) is a soft AP setup utility that steals usernames and passwords from public wireless hotspots by confusing users with DNS and HTTP redirects from a competing access point (often using HostAP).
- Fake AP (www.blackalchemy.to/project/fakeap/) is an open source application that confuses attackers by generating beacon frames from thousands of bogus 802.11b access points.

A rogue client may be set up by modifying the MAC address to match an authorized MAC address on the access control list. An example of a tool to modify a MAC address is:

- SMAC (www.klcconsulting.net/smac) is a Windows MAC address

modifying utility which allows users to change the MAC address for network interface cards on the Windows 2000, XP, and 2003 Server systems, regardless of whether the manufacturers allow this option or not.

A Denial of Service (DoS) attack can be launched with a variety of devices, including commonly available devices such as microwaves and cellular telephones. In addition, there are also software tools to launch attacks. An example is:

- fata_jack (packetstormsecurity.nl/wireless/indexdate.shtml) is a Denial of Service tool that sends authenticate frames to an access point with inappropriate authentication algorithms and status codes. The access point then drops connections with wireless devices.

In addition, there are many tools designed to break WEP encryption. Examples include:

- WEPCrack (sourceforge.net/projects/wepcrack/) is a freeware encryption breaker that cracks 802.11 WEP encryption keys using the discovered weakness of RC4 key scheduling.
- AirSnort (airsnort.shmoo.com) is a freeware encryption breaker that passively monitors transmissions, and computes the encryption key when enough packets have been gathered.
- WEPWedgie (sourceforge.net/projects/wepwedgie/) is a toolkit for determining 802.11 WEP keystreams and injecting traffic with known keystreams in order to crack WEP within minutes.

Though the above tools can be used for defensive as well as offensive purposes on wireless networks, there are several tools that lend themselves mainly to protecting wireless networks. Examples include:

- WIDZ (Wireless Intrusion Detection System) (www.loud-fat-bloke.co.uk/tools.html) is an IDS for 802.11. It guards APs and monitors local frequencies for potentially malevolent activity. It can detect scans, association floods, and rogue APs. It can easily be integrated with the Snort or Realsecure Network 10/100 intrusion detection systems.
- AirDefense Guard (www.airdefense.net/products/intrusion_detection.shtm) performs signature analysis, detects policy deviation (including protocol assessment policy deviation), and checks for statistically anomalous behavior.
- Snort-Wireless (snort-wireless.org) is an open source solution similar to AirDefense Guard. It allows for 802.11 specific detection rules through the new "wifi" rule protocol. It also detects rogue APs, AdHoc networks, and NetStumbler use.

Developer Notes for Sessionlogger

Mark Lodato Jason Franklin
mlodato@sandia.gov jasfran@sandia.gov
Sandia National Laboratories, California

September 16, 2005

This document describes the design of the Sessionlogger program. For an end-user description of the program, see the man page by typing: `make man && less sessionlogger.1`.

1 Files

Sessionlogger is broken up into the following files:

common.c/h	Common output functions
constants.h	All constants used in the program
decode.c/h	Functions that decode raw frames
hash.c/h	Hash table implementation
lst.c/h	Linked-list implementation
main.c/h	Functions involving program operation
netowrk.c/h	All network-related code
packet_buffer.c/h	Queue implementations for buffering frames for raw writing
queue.c/h	Queue implementation
rawlogger.c/h	All raw log writing functions
sessionlogger.c/h	All session-related code
states.c/h	Stateful-flagging
struct.h	All structs used by the program
unused.c/h	Currently unused code that may be useful in the future

2 Walk through

When sessionlogger starts in *main.c*, it does a few initializations, reads the command-line arguments, then tries to connect to each sniffer listed in sniffer-file. For each successful connection, `main()` launches a new thread, starting at `serve()`. It then waits for an interrupt (SIGINT or SIGTERM). When it receives one, it closes all connections, flushes the output buffer, finishes all logs, and quits.

In `serve()`, each thread performs a handshake with its sniffer, getting the hostname of the sniffer, sending option flags and sending the user-specified filter.

It then, in a loop, waits for a packet to arrive from the sniffer. When it does, it calls `setup_s_packet()` to parse the frame, `handleFrm()` to perform session logging, and `pb_update()` to perform raw logging.

`handleFrm()` contains the primary stateful logging logic for sessions. Upon receiving a frame, `handleFrm()` calls `lookupSession()` to check if the current frame being processed is part of a session which has already been created or if the frame is the beginning of a new session. In order to lookup sessions associated with the current frame, the function `getKey()` is called to parse the MAC addresses of the frame and return a key struct. This keys struct is then passed to `lookup()` to return the corresponding value in the hash table of session.

Assuming the hash lookup returns a non-NULL session or what is referred to as an old session in the program, `updateSession()` is called to check if the current frame has already been seen (a duplicate frame), check if the frame has been seen by a different sniffer (a new sniffer), and update the session state accordingly.

If the hash lookup returns a NULL session or what is referred to as a new session in the program, `initSession()` is called and state is allocated for a new session. `initSession()` initializes the values of the session struct and returns the new struct to `handleFrm()` which inserts the struct into the hash table of sessions by calling `insert()`. After which, a new session log is written to the open session log file by calling `writeSession()`.

In both `initSession()` and `updateSession()`, specific type and subtype fields in the wireless frame being processed cause the program to set flags for logging and to begin and end sessions. In the current program, deauthentication and reassociation response frames cause `updateSession()` to end a session, and deauthentication frames cause `initSession()` to end a session. This behavior is a direct result of our definition of a session as what happens between authentication and deauthentication for a specific tuple (source MAC address, destination MAC address, BSSID).

After modifying an old session or creating a new session, `handleFrm()` iterates through a list containing all the sessions in order to time out sessions which have not been recently updated. Sessions that time out are written to disk and the memory they previously occupied is reclaimed by calling `endSession()`.

In order to add hostnames to the raw log file, a method of frame buffering was needed. If we simply write out the raw frame as we see it, we would have no way of going back to update the frame and add in a new hostname that saw the frame. Therefore, the last 5 management/data frames seen are saved in a queue. Any control frames seen after the last management/data frame is saved in a queue attached to that management/data frame. This is because control frames have no sequence numbers, so it would be difficult to check for duplicates. Once more than 5 frames are in the queue, the oldest management/data frame, along with its control frames, are output to the file. The buffering occurs in *packet_buffer.c* and the raw-log file operations occur in *rawlogger.c*.

3 Raw Log

The raw log is a *libpcap* format file with an IEEE 802.11 link type. If the **-H** option is specified, hostnames are added to the end of the file. This makes the frames look malformed to *ethereal*, but they seem to parse somewhat fine. You can manually see the hostnames by looking at the end of the raw dump. A parser is being developed that reads the special hostname-added files.

This page intentionally left blank.

NAME

sessionlogger - distributed, stateful wireless logging utility

SYNOPSIS

sessionlogger -s *sniffer-list* [**-c** *closed-file*] [**-d** *date-fmt*]
[**-f** *filter*] [**-H**] [**-o** *open-file*] [**-r** *raw-file*]
[**-v** *verbosity*]
sessionlogger -h

DESCRIPTION

Sessionlogger is a distributed, stateful wireless logging utility to record link-layer information about all 802.11 traffic within range. Throughout the physical site, one or more machines are set up in monitor mode and run **sniffer**(1). The user then runs **sessionlogger**, which connects to each **sniffer**, aggregates the data, and logs when wireless sessions are opened and closed.

OPTIONS

The following is a list of options available to **sessionlogger**.

NOTES: If using **-s** (that is, normal operation), at least one of {**-o**|**-c**|**-r**} must be specified. For each of these output filenames, the first '%' character is replaced with the current timestamp.

-c *closed-file*
Write a log of all session terminations to *closed-file*.

-d *date-fmt*
Set timestamp format (for **-o**, **-c**, or **-r**) to *date-fmt*. See **strftime**(3) for format of timestamp string. If unspecified, defaults to "%EY-%m-%d.%H:%M:%S-%Z".

-f *filter* Apply the specified **tcpdump** filter for each sniffer. See **tcpdump**(1) for format of filter string.

-h Show a help message showing program usage.

-H Append to the end of each frame in the raw log the hostname of each sniffer that saw the frame.
NOTE: This may make the file unreadable by other programs.

-o *open-file*
Write a log of all new sessions to *open-file*.

-r *raw-file*
Write a **libpcap** format log of all raw frames to *raw-file*.

-s *session-file*
Read from *sniffer-list* a list of the sniffers to connect to. Each line of the file is of the form:
ip address port

-s *verbosity*
Set the verbosity level (Range: 0-4; Default: 2).

LOG FORMAT

The open and closed logs are of the following format:

ft -> *lt*: *sa* -> *da* -> *ap* [(*hn*)] *to ap* *from ap* (<*s1*><*s2*><*s3*>)

ft Timestamp of first frame seen

lt Timestamp of last frame seen

sa	Source MAC address
da	Destination MAC address
bssid	BSSID of session
hn	List of hostnames that saw the session
to_ap	Bytes of data payload sent to the ap.
from_ap	Bytes of data payload sent from the ap.
s1	Frames seen in state 1 (unassociated and unauthenticated)
s2	Frames seen in state 2 (associated but unauthenticated)
s3	Frames seen in state 3 (associated and authenticated)

Type Flags

The following is a legend for the flags in s1, s2, and s3 above.

a	Association Request
A	Association Response
b	Beacon
c	Reserved Control Frame
C	Clear-to-Send
d	Data
D	Disassociation
f	Data + Contention Free {ACK | Poll | ACK+Poll}
F	Contention Free {ACK | Poll | ACK+Poll}
K	Acknowledgement
l	Power-Save Poll
m	Reserved Management Frame
n	Null data
p	Probe Request
P	Probe Response
r	Reassociation Request
R	Reassociation Response
S	Request-to-Send
t	ATIM
u	Authentication
U	Deauthentication
v	Reserved Data Frame
V	Reserved (Type 4) Frame

EXAMPLES

Emulate as a distributed **tcpdump** outputing to pcap.log:

```
sessionlogger -s s.lst -r pcap.log
```

Log all new sessions to a file, YYYY-MM-DD.HH:MM:SS-TMZ open.log:

```
sessionlogger -s s.lst -o %_open.log
```

Record a raw log (with hostnames), ignoring beacon frames:

```
sessionlogger -s s.lst -H -r pcap.log -f 'wlan[0] & 0xfc != 0x80'
```

CAVEATS

Some pcap filters may cause the session logger (**-o** and **-c**) to not function correctly.

AUTHORS

Sessionlogger and sniffer were written by Jason Franklin, Mark Lodato, and Dimitry Averin for Sandia National Laboratories.

SEE ALSO

sniffer(1), **tcpdump**(1), **pcap**(3), **strftime**(3)

This page intentionally left blank.

NAME

sniffer - wireless sniffer for use with sessionlogger

SYNOPSIS

sniffer -i *interface* [**-q**] [**-p** *port*]
sniffer -f *input-file* [**-q**] [**-p** *port*]
sniffer -h

DESCRIPTION

Sniffer is a tool that sniffs wireless frames using **libpcap** and sends them to **sessionlogger**. See **sessionlogger**(1) for more information.

OPTIONS

-f *input-file*
Test mode. Use the **libpcap** format *input-file* as input to the program rather than a real device. When using this mode, run the sniffer, connect with the session logger, then hit a key on the sniffer's console. You will probably have to CTRL-C to quit, since this mode is **unsupported** and is to be used only as a diagnostic.

-h Show a help message showing program usage.

-i *interface*
Listen on specified interface.

-p *port* Listen on specified port rather than default of 1946.

-q Quiet operation - turn off all console output.

KNOWN ISSUES

Currently, **sniffer** only works with drivers with no radio headers or Prism radio headers.

SEE ALSO

sessionlogger(1)

This page intentionally left blank.

Distribution

1	MS 0455	R. S. Tamashiro
1	MS 0801	D. S. Rarick
1	MS 0671	M. J. Skroch
1	MS 0671	J. F. Clem
1	MS 0671	B. J. Smith
1	MS 0671	R. E. Trellue
1	MS 0672	R. L. Hutchinson
1	MS 0795	P. C. Jones
1	MS 0795	K. S. Nauer
1	MS 0795	A. A. Quintana
1	MS 0795	R. A. Suppona
1	MS 0139	A. L. Hale
1	MS 0813	G. K. Rogers
1	MS 0813	R. M. Cahoon
1	MS 9011	N. A. Durgin
1	MS 0630	B. V. Hess
1	MS 9011	J. D. Howard
1	MS 9011	S. A. Hurd
1	MS 9011	J. A. Hutchins
1	MS 9011	R. McClelland-Bane
1	MS 9011	E. D. Thomas
1	MS 9011	T. J. Toole
3	MS 9011	J. A. Van Randwyk
1	MS 9012	B. A. Maxwell
1	MS 9158	M. W. Sukalski
1	MS 9159	H. R. Ammerlahn
1	MS 9152	J. A. Friesen
1	MS 9159	M. F. Hardwick
1	MS 9151	J. L. Handrock
1	MS 9151	C. T. Oien
1	MS 9151	L. M. Napolitano
1	MS 0630	K. E. Washington
1	MS 0795	D. Kilman
1	MS 0672	E. J. Lee
1	MS 0672	R. P. Custer
1	MS 0123	D. L. Chavez
1	MS 0672	J. T. Michalski
1	MS 0806	T. D. Tarman
1	MS 0672	L. G. Pierson
1	MS 0672	B. P. Van Leeuwen
1	MS 1206	M. H. Johnson
1	MS 1206	J. V. Vonderheide
2	MS 9018	Central Technical Files, 8945-1
2	MS 0899	Technical Library, 4536
1	MS 0188	D. Chavez, LDRD Office, 1011


Sandia National Laboratories