

401.4

Secure Communications



SANS

THE MOST TRUSTED SOURCE FOR INFORMATION SECURITY TRAINING, CERTIFICATION, AND RESEARCH | sans.org

Copyright © 2016, The SANS Institute. All rights reserved. The entire contents of this publication are the property of the SANS Institute.

PLEASE READ THE TERMS AND CONDITIONS OF THIS COURSEWARE LICENSE AGREEMENT ("CLA") CAREFULLY BEFORE USING ANY OF THE COURSEWARE ASSOCIATED WITH THE SANS COURSE. THIS IS A LEGAL AND ENFORCEABLE CONTRACT BETWEEN YOU (THE "USER") AND THE SANS INSTITUTE FOR THE COURSEWARE. YOU AGREE THAT THIS AGREEMENT IS ENFORCEABLE LIKE ANY WRITTEN NEGOTIATED AGREEMENT SIGNED BY YOU.

With the CLA, the SANS Institute hereby grants User a personal, non-exclusive license to use the Courseware subject to the terms of this agreement. Courseware includes all printed materials, including course books and lab workbooks, as well as any digital or other media, virtual machines, and/or data sets distributed by the SANS Institute to the User for use in the SANS class associated with the Courseware. User agrees that the CLA is the complete and exclusive statement of agreement between The SANS Institute and you and that this CLA supersedes any oral or written proposal, agreement or other communication relating to the subject matter of this CLA.

BY ACCEPTING THIS COURSEWARE YOU AGREE TO BE BOUND BY THE TERMS OF THIS CLA. BY ACCEPTING THIS SOFTWARE, YOU AGREE THAT ANY BREACH OF THE TERMS OF THIS CLA MAY CAUSE IRREPARABLE HARM AND SIGNIFICANT INJURY TO THE SANS INSTITUTE, AND THAT THE SANS INSTITUTE MAY ENFORCE THESE PROVISIONS BY INJUNCTION (WITHOUT THE NECESSITY OF POSTING BOND), SPECIFIC PERFORMANCE, OR OTHER EQUITABLE RELIEF.

If you do not agree, you may return the Courseware to the SANS Institute for a full refund, if applicable.

User may not copy, reproduce, re-publish, distribute, display, modify or create derivative works based upon all or any portion of the Courseware, in any medium whether printed, electronic or otherwise, for any purpose, without the express prior written consent of the SANS Institute. Additionally, User may not sell, rent, lease, trade, or otherwise transfer the Courseware in any way, shape, or form without the express written consent of the SANS Institute.

If any provision of this CLA is declared unenforceable in any jurisdiction, then such provision shall be deemed to be severable from this CLA and shall not affect the remainder thereof. An amendment or addendum to this CLA may accompany this courseware.

SANS acknowledges that any and all software and/or tools, graphics, images, tables, charts or graphs presented in this courseware are the sole property of their respective trademark/registered/copyright owners, including:

AirDrop, AirPort, AirPort Time Capsule, Apple, Apple Remote Desktop, Apple TV, App Nap, Back to My Mac, Boot Camp, Cocoa, FaceTime, FileVault, Finder, FireWire, FireWire logo, iCal, iChat, iLife, iMac, iMessage, iPad, iPad Air, iPad Mini, iPhone, iPhoto, iPod, iPod classic, iPod shuffle, iPod nano, iPod touch, iTunes, iTunes logo, iWork, Keychain, Keynote, Mac, Mac Logo, MacBook, MacBook Air, MacBook Pro, Macintosh, Mac OS, Mac Pro, Numbers, OS X, Pages, Passbook, Retina, Safari, Siri, Spaces, Spotlight, There's an app for that, Time Capsule, Time Machine, Touch ID, Xcode, Xserve, App Store, and iCloud are registered trademarks of Apple Inc.

Governing Law: This Agreement shall be governed by the laws of the State of Maryland, USA.

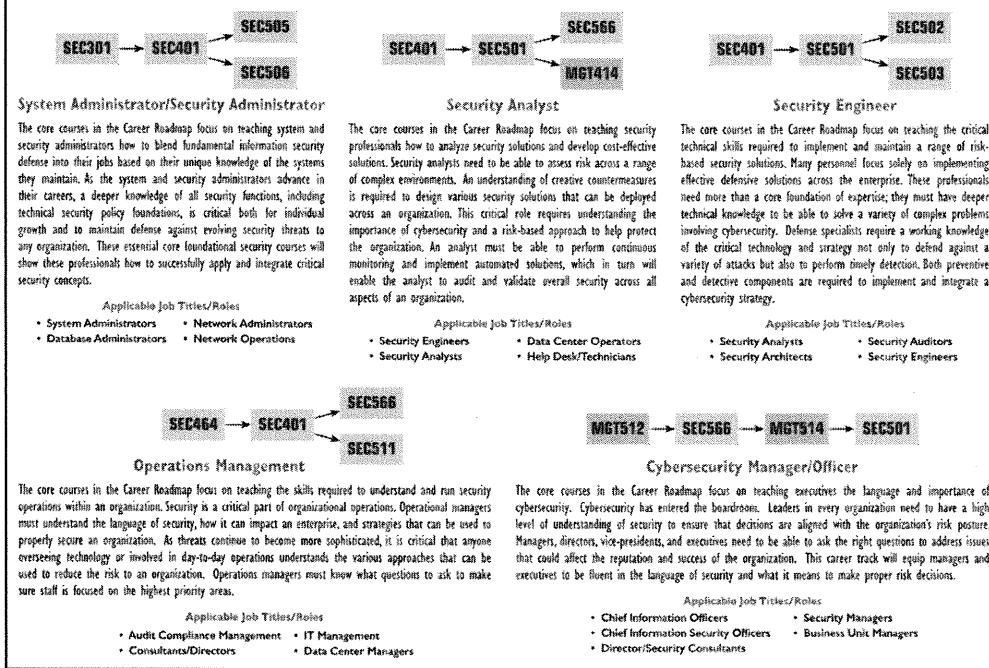
SECURITY 401

SANS Security Essentials

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

This page intentionally left blank.

SANS CYBER DEFENSE CORE SECURITY ROADMAPS



This page intentionally left blank.

SANS CYBER DEFENSE SPECIALIZED ROADMAPS

SEC440 → SEC480 → SEC566

Security Architect

The core courses in the Career Roadmap focus on planning, designing, and implementing an effective security solution. In order for security to be effective it must be customized to the unique business, mission, and risks an organization faces. The security strategist must be able to identify core metrics and use them to design and oversee the implementation of a security system and network architecture. Having a secure robust network architecture is critical for an organization to have effective security.

Applicable Job Titles/Roles

- Security Managers • System Architects
- Data Center Analysts • Design Engineers

SEC901 → SEC911 → SEC903 → FOR572

Security Operations Center (SOC) Analyst

The core courses in the Career Roadmap focus on building, running, and deploying a SOC. Security has become critical for the success of an organization constantly under attack. Defense against the vast array of attacks requires continuous monitoring of the network and assessment of the security infrastructure. Properly trained people who can detect and respond to attacks are critical to the overall success and security of an organization.

Applicable Job Titles/Roles

- Security Consultants • Security Operations Supervisors
- SOC Managers • Security Operations Directors

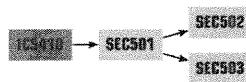
MGT415 → SEC566 → AUD507 → SEC911

Security Risk Officer

The core courses in the Career Roadmap focus on assessing and analyzing risk and using that information to guide the priorities for security. In order for organizations to be successful in security, they must take a risk-based approach. Risk allows an organization to identify the vulnerabilities that have the biggest impact, based on the threats that have the highest likelihood of success, and which are most linked to the organization's critical assets. Proper metrics that map back to risk are used to assess and verify that an organization's security program is focused on the correct areas.

Applicable Job Titles/Roles

- Risk Engineers • System Managers
- Risk Officers • Auditors

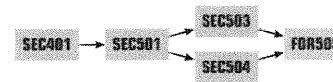


Industrial Control Systems (ICS) Analyst

The core courses in the Career Roadmap focus on teaching how to assess, implement, and secure ICS. Anyone who works in critical infrastructure needs to understand the associated threats and methods for security and the proper ways to protect systems that support a variety of ICS environments. ICS represent unique challenges not only in terms of threats, but also in terms of the unique methods that must be used to reduce risk to these systems. The focus is on providing an appropriate level of security based on the security challenges that these organizations face.

Applicable Job Titles/Roles

- Control System Engineers • Control System Managers
- Operational Analysts • System Administrators



Intrusion Analyst

The core courses in the Career Roadmap focus on teaching the foundations of security, as well as on the prevention and detection of threats. The most masterful prevention measures may be circumvented by skilled attackers. Successful attacks must be quickly identified to minimize the damage. The focus is on implementing appropriate prevention methods, rapid detection and assessment of malicious activity, and containment of harm in the aftermath of a successful attack.

Applicable Job Titles/Roles

- System Administrators • IDS Specialists
- Security Analysts/Specialists • SOC Engineers
- Intrusion Detection Analysts

This page intentionally left blank.

Module 18: Encryption 101

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Module 18: Encryption 101

This section intentionally left blank.

Encryption 101

SANS Security Essentials IV: Secure Communications

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Introduction: Encryption 101

Cryptography, the science of secret writing, helps us communicate without revealing the meaning of information to adversaries and also potentially validates to whom we are communicating. It can protect any kind of data, from very sensitive information, such as Internet-based commerce and banking transactions, to harmless messages you would just rather no one else knew about, such as a letter to a friend. Cryptography, also abbreviated as “crypto,” can provide a great deal of confidentiality and integrity checks for information. However, it is not a silver bullet, and it can lead to a tremendous false sense of security unless used properly. Cryptography should always be a part of a larger defense-in-depth strategy, providing just one layer of the security onion.

Objectives

- Cryptosystem fundamentals
- General types of cryptosystems:
 - Symmetric
 - Asymmetric
 - Hash

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Objectives

We begin our study with some examples that illustrate the importance of sound cryptographic practices. We then take a closer look at the basic reasons cryptography, despite its potential power, is difficult to implement correctly. Then, we dive into the technical material with a discussion of how it all works, building a foundation for the cryptosystems covered in the next section.

Crypto Fundamentals

The student will have a basic understanding of the fundamental concepts of cryptography.

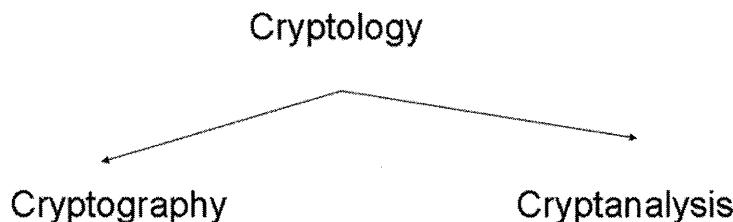
SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Crypto Fundamentals

This section intentionally left blank.

Cryptography and Cryptology

- **Cryptography** : Art and science of hiding the meaning of a communication from unintended recipients. The word “cryptography” comes from the Greek words, kryptos (hidden) and graphein (to write).
- **Cryptology**: Encompasses cryptography and cryptanalysis.



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Cryptography and Cryptology

Who creates these encryption algorithms? Computer scientists called “cryptographers,” who are well trained in several different fields of mathematics and who usually work in groups, take many years to invent and refine ciphers. But with so much depending on cryptography, individuals called “cryptanalysts” dedicate their lives to breaking ciphers. Some cryptanalysts work for the military and for governments; others are simply interested in the study of ciphers and want to find weaknesses in ciphers to ensure that they cannot be broken by others. The generic term for the study of both cryptography and cryptanalysis is called “cryptology.”

The following are key terms relating to cryptography and cryptology:

Block cipher: Obtained by segregating plaintext into blocks of n characters or bits and applying the identical encryption algorithm and key to each block.

Cipher: A cryptographic transformation that operates on characters or bits.

Ciphertext or cryptogram: An unintelligible message.

Clustering: Situation in which a plaintext message generates identical ciphertext messages using the same transformation algorithm, but with different cryptovariables or keys.

Codes: A cryptographic transformation that operates at the level of words or phrases.

Cryptanalysis: Act of obtaining the plaintext or key from ciphertext that is used to obtain valuable information and to pass on altered or fake messages to deceive the original intended recipient.

Cryptographic algorithm: A step-by-step procedure used to encipher plaintext and decipher ciphertext.

Plaintext: A message in cleartext readable form.

What is Cryptography?

- Cryptography: "Hidden writing"
- Encryption: Coding a message in such a way that its meaning is concealed
- Decryption: The process of transforming an encrypted message into its original form
- Plaintext: A message in its original form
- Ciphertext: A message in its encrypted form

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

What is Cryptography?

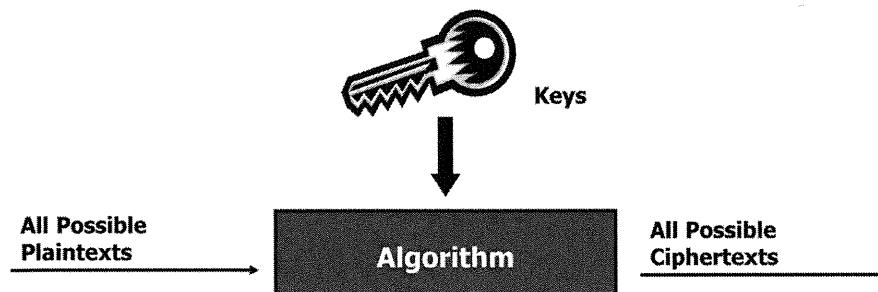
Cryptography is vitally important to information security. One of the main goals of cryptography is to help fend off eavesdroppers. The idea is that communicating over any kind of medium has the inherent risk that an unauthorized third-party could be listening in, and we want to minimize or eliminate that risk. So, in its most basic form, cryptography garbles text in such a way that anyone who intercepts the message cannot understand it.

Nearly every cryptographic algorithm performs two distinct operations: encryption and decryption. *Encryption* is the practice of coding a message in such a way that its meaning is concealed. How the message is transformed depends on a mathematical formula called an *encryption algorithm* or a *cipher*. Once a message has been transformed with a cipher, the resulting message is called *ciphertext*. Because ciphertext contains the message in its encrypted form and not its native form, it is unintelligible or has no meaning. For the recipient of the ciphertext to read the message, the recipient must *decrypt* it. *Decryption* is the process of transforming an encrypted message back into its original *plaintext* or *cleartext* form. It is important to note that a plaintext message refers to any type of message in its unencrypted form. A plaintext message is not just an ASCII text message; an executable is also considered a plaintext message if it is not encrypted.

Who creates these encryption algorithms? Computer scientists called *cryptographers*, who are well trained in several different fields of mathematics and usually work in groups, take many years to invent and refine ciphers.

But with so much depending on cryptography, there are also individuals called *cryptanalysts*, who dedicate their lives to *breaking* ciphers. Some cryptanalysts work for the military and for governments; others are just interested in the study of ciphers and want to find weaknesses in ciphers to ensure they cannot be broken by others. The generic term for the study of both cryptography and cryptanalysis is called *cryptology*.

Cryptosystems



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Cryptosystems

A *cryptosystem* is the collection of all possible inputs and all possible outputs, in addition to the algorithm and keys. But never forget about the humans. Good cryptography is very, very strong and in some instances, as with the Rijndael algorithm, cannot be "broken" for 14 trillion years or more. That is pretty strong. If you were an attacker, which would you attack? A very strong algorithm or the users? Which is weaker? Never forget that humans are a critical aspect of a cryptosystem. They must be trained properly in using the system as well as protecting their keys. Lose the keys (by tricking a user into giving them up) and the whole security of the cryptosystem collapses.

We know what you are thinking, what are "all possible plaintexts"? Imagine any form of data, or any kind of message you can think up. "I went to the store" is a valid inclusion into all possible plaintexts just as "Bob was here," or an mp3 file. The resultant cryptographic transformations of all possible plaintexts are "all possible ciphertexts."

Keys

- Keys permit the existence of unrestricted algorithms
- Keys might be any one of a large number of values
- The strength of a cryptosystem rests with the strength of its keys
- Keyspace matters!

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Keys

Cryptographic keys are simply values used to initialize a particular algorithm. The important aspect of keys in regards to cryptosystems is that only the key, not the algorithm, needs to be protected. This means that algorithms might be widely distributed and their internal workings publicly documented. It is only the key that must be protected from thievery by communicating entities.

Keys are critical components of a cryptosystem. These keys are similar to a key to your house or safety deposit box insofar as cryptographic keys provide access to protected resources, namely information intended to be kept secret.

The uniqueness of cryptographic keys is just as important as the keys themselves. Most airline travelers are probably familiar with the keys that come with luggage locks. These keys are far from unique. Purchase one set of lock and keys, and you have a fairly good chance of opening just about any randomly selected luggage lock you come across. Luggage-lock keys are a great example of a lack of uniqueness or of an "insufficient keyspace."

Keyspace is a critical concept concerning cryptographic keys. The larger the keyspace, the less likely an attacker is to discover a given key through brute force. A brute force attack on a key involves trying every possible key until finding one that works. For instance, the Caesar Cipher had only a keyspace of 25 possible keys, which is trivial to exhaust or brute force.

Contrast luggage keys against car keys, for example. Car keys need to be more unique than luggage keys because of the importance and expense of automobiles.

Car keys can be said to have a larger keyspace and, therefore, offer more protection because the possible combinations of ridges and valleys on a car key is quite large. Because it is practically impossible for an attacker to guess the correct sequence of ridges and valleys that match your original car key (a brute-force attack), the attacker needs to either steal your keys or make an illicit copy to gain unauthorized access to your car.

Conceptually, cryptographic keys should, at a minimum, provide the same level of uniqueness required for car keys, an abundant combination of ridges and valleys. In reality, of course, cryptographic keys need a much larger keyspace than that used for car keys. It should be impossible for an attacker to guess a cryptographic key that matches the one used to encrypt correspondence. Let's say the total number of possible unique car keys is approximately 200,000. Although that might not be accurate, even if the number is 10 million, the total number of possible unique cryptographic keys for a given cryptosystem needs to a billion times larger simply to afford the keyspace protection against guessing an encryption key through brute force. Why? It's far easier to use a computer to iterate through a billion cryptographic keys than it is to physically recreate a million car keys. In short, a cryptographic keyspace must be absolutely enormous to afford sufficient protection.

So what should this mean to you? Keyspace matters. The bigger, the better.

Key Protection

40 bits of protection

```
0011101010010101010110101010101000110
```

128 bits of protection

```
0011101010010101010110101000110 0011101010010101011010101000110 0011101010010101011010101000110
```

128-bit keys offer approximately a trillion times more protection than 40-bit keys.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Key Protection

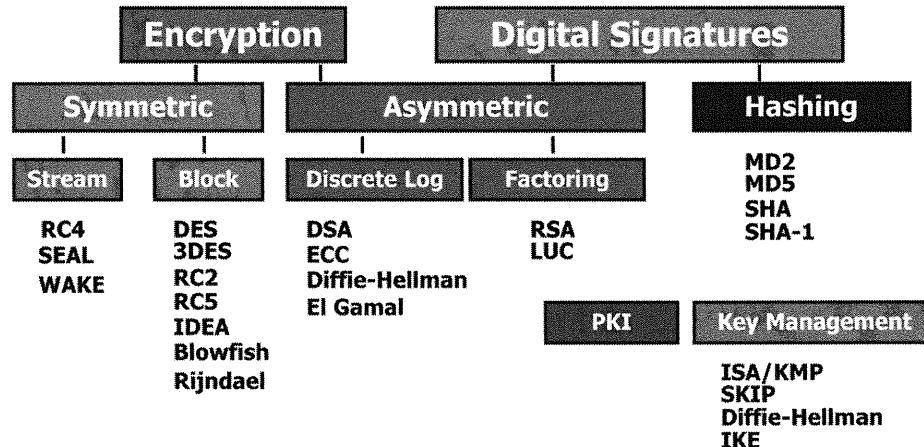
The uniqueness of a cryptographic key directly correlates to the amount of protection the key provides. In our previous discussion of car keys, the total number of combinations of ridges and valleys constitutes the size of the keyspace; the larger the number of combinations, the more likely the key is unique (making it more difficult for an attacker to guess a matching key).

The limitations of car keys are their length and their engraving depth; most are only 1.5 to 3 inches long depending on the manufacturer. Keys can be engraved only so deeply before weakening the substance of which they are made (fragile keys are little use). These physical traits limit the number of keys that can be made, therefore, limiting the total keyspace. If, however, car keys were 12 or even 20 inches long, the total keyspace would dramatically increase because they could support larger combinations of ridges and valleys. (Not too many consumers would want to carry around a 20-inch car key, though.)

Conceptually, the length of cryptographic keys is similar to the length of car keys; the longer the key, the more opportunity for uniqueness, and the more difficult it is for an attacker to guess a corresponding key.

In the case of cryptographic keys, the length of the key correlates to the amount of protection the key provides. A 128-bit key offers about one trillion more times protection than a 40-bit key. Although it is obvious that a 128-bit key is longer than a 40-bit key, the difference in the amount of protection is exponential, not linear. Go ahead, prove it to yourself: $2^{40} (1.1 \times 10^{12})$ compared to $2^{128} (3.4 \times 10^{38})$.

Enterprise Crypto: The Big Picture



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Enterprise Crypto: The Big Picture

When we talk about cryptography, it is no wonder people get dizzy. Not only is the mathematics of cryptography fairly esoteric and convoluted, but there are also many different kinds of cryptographic systems. We discuss each of these types of crypto in more detail in this course, but this slide provides an overview.

Symmetric stream ciphers are fast, and asymmetric factoring algorithms are slow. Diffie-Hellman is great for secure key exchanges, but not necessarily optimal for encryption. What does this all mean? Do not fret, we'll explain. For now, you need to know that the use of cryptography in the enterprise is a multifaceted endeavor. Different types of cryptography are used for different types of situations and often, cryptographic systems will be employed in concert.

For instance, to encrypt a message, that is, transform it in such a way that prying eyes cannot read it; we might choose either *symmetric* algorithms such as RC4 or Blowfish, or *asymmetric* algorithms like RSA or ECC, but not any of the hashing algorithms such as MD5 or SHA-1.

However, to digitally sign a message, that is, give some type of "digital proof" as to the signer's identity, we might choose RSA or ECC, but not any of the symmetric algorithms.

Finally, if we need high-speed encryption with the advantage of digital signatures, we might choose Diffie-Hellman to exchange a symmetric key, hash our message using SHA-1, digitally sign the hash using RSA, and encrypt the message and hash for transmission using 3DES.

As we can see, cryptography in the enterprise promises to be a challenging topic. After completing this module, the reader will have a rudimentary understanding of all the previous topics.

Security by Obscurity is No Security!

- DVD "encryption" is a case-in-point
- Proprietary algorithms are high risk
- "Tamper-proof" hardware can be defeated with sufficient effort
- Technical solutions usually do not satisfactorily address legal issues

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Security by Obscurity is No Security!

Everyone loves DVDs. Never before have we been able to see our favorite movies in such breathtaking detail on our home televisions. But not everyone is aware of the lessons in cryptography best practices that lurk behind the scenes of DVD mania:

- Never believe in the strength of a secret or proprietary cryptographic algorithm. The algorithm will be eventually discovered, and if knowing the algorithm makes it trivial to decrypt a message without the appropriate key, all communications encrypted with that algorithm are compromised.
- Never rely on a single technology (or any other measure) as your only line of defense. Defense-in-depth is layering countermeasures for completeness and redundancy. Just encrypting everything is not enough.
- Above all, never attempt to write your own encryption system. There are plenty of superb algorithms with free implementations available. Unless you are a seasoned cryptographer, and you think you can improve on AES, Blowfish, RSA, and so on, do not bother trying.

So, what happened with DVDs? The motion picture industry spent years secretly developing its own standard for encryption—the Contents Scrambling System (CSS). CSS attempted to prevent unauthorized playing of DVDs by encrypting the data on the DVDs. Each DVD included a key that could be used to decrypt the data and a hash (fixed-length value computed from the plaintext) to verify that the data was correctly decrypted.

That key was encrypted and could be decrypted only with one of the player keys, which were built into every DVD player. Instead of submitting the CSS standard for review, which would have taken advantage of the collective brainpower of cryptologists worldwide, they implemented the standard themselves, and released a product (DVDs) that relied on the cipher.

According to Frank Stevenson, who published a cryptanalysis of CSS, the cipher was designed with a 40-bit key length (inadequate in itself) to meet U.S. export regulations. However, only 225 keys are necessary in a brute-force attack. He estimates it would take less than 18 seconds on a 450MHz PC to recover a disk key from the hash. According to Stevenson, "If the cipher was intended to get security by remaining secret, this is yet another testament to the fact that security through obscurity is an unworkable principle."

Soon after, a couple of technologists, Canman and SoupaFr0g, decoded that magic algorithm and released a program that became very popular. DeCSS 1.2b pulls the decrypted data off the DVD disk and stores it so it can be played like any other multimedia file. Don't want to pay \$20 for a movie DVD? No problem! Just "borrow" it from a friend.

Professional cryptanalysts spend their time looking for tiny flaws and even tinier clues in encrypted messages, so as to break the cipher. Canman was a very good amateur, and he broke an under-scrutinized crypto algorithm. For an algorithm to be good, it has to be objectively examined by people whose job it is to find flaws. The motion picture industry thought they would be clever, but with crypto, clever is not sufficient. There is no substitute for public scrutiny of a cipher.

Beware of Overconfidence

- Large key lengths is a case-in-point
- Simply using popular cryptographic algorithms, with large key lengths, does not make your system secure
- You must protect your key!
- Cryptanalytic compromises usually originate from totally unexpected places

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Beware of Overconfidence

Our second case study explores the risks of being overly confident with cryptographic solutions. All aspects of cryptosystems are subject to attack, especially the keys. Despite their importance, keys are seldom adequately protected. There are many situations that threaten key integrity. When a workstation is compromised (or under surveillance by the FBI or other law enforcement), capturing keystrokes is trivial. A faulty cipher implementation might temporarily expose keys. But perhaps the most likely cause of key compromise is the tendency of humans to fail to protect their keys, storing them on sticky pieces of paper under the keyboard or blurting them out to anyone who calls and claims to be with "Security" or "Technical Support."

Crypto History

- The history of cryptography is long and interesting
- The next couple of slides discuss some of the highlights

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Crypto History

Even though the popularity of modern computers has brought crypto to a new level of automation, the concepts behind cryptography are not new. In 3000 B.C., the Egyptians referred to crypto as secret writing because they were able to use various techniques to keep their communication secret among a small group of people. This was the beginning of secret communications. The technique of using special symbols seems trivial today, but it was difficult to crack at the time.

Crypto History: Secret Writing

Egyptian hieroglyphics: 3000 B.C.

- Derived from the Greek word “hieroglyphica,” which means sacred carvings
- Hieroglyphics evolved into hieratic, which was a stylized script that was easier to use

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Crypto History: Secret Writing

Cryptography dates back to the ancient Egyptians, who began using secret writing called *hieroglyphics* as early as 3000 B.C. The term comes from the ancient Greek word *hieroglyphica*, meaning *sacred carvings*. The Egyptians used this writing to hide messages from unintended recipients.

Crypto History: Spartan Scytale

- 400 B.C.—Military cryptography
- Strip of papyrus or parchment wrapped around a wooden rod
- Message to be encoded was written lengthwise down (or up) on the wrapped material of the rod
- Material was unwrapped and carried to the recipient
- Material was rewound on a rod of the same diameter, to read

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Crypto History: Spartan Scytale

In 400 B.C., the Spartans developed a fairly sophisticated method of encryption based on the current form of communication. With this technique, the two parties that wanted to communicate were required to use wooden rods of the same diameter. The sender wrapped a long piece of cloth around the wooden post and wrote his message vertically across the pole. He then removed the cloth from the rod and carried it with him. If anyone looked at the cloth, it contained random type characters. It was only when the cloth was wrapped around a wooden rod of the same diameter that the message was revealed.

Crypto History: Caesar Cipher

- 50 B.C—Julius Caesar
- Substitution cipher
- Substituted letters of the alphabet for other letters of the same alphabet
- Mono-alphabetic substitution
- Involved shifting the alphabet three letters and substituting those letters
- Sometimes known as c3 substitution cipher:
 - CAB becomes FDE.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Crypto History: Caesar Cipher

Based on what we have learned already, it should be of no surprise that Julius Caesar used encryption. For his time, his technique was extremely advanced. The Caesar cipher used a simple rotation to encrypt a message. The Caesar cipher used a rot-3, or rotation 3, in which each letter of the alphabet was rotated three letters forward. In our language, the letter A became D. To decrypt the message, you would rotate the letter back three places in the alphabet.

Unix ROT 13

Unix systems use a substitution cipher called ROT 13:

- It shifts the alphabet by 13 places
- Another shift of 13 places brings the alphabet back to its original position, thus decoding the message

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Unix ROT 13

Unix systems use a substitution cipher called ROT 13:

- It shifts the alphabet by 13 places.
- Another shift of 13 places brings the alphabet back to its original position, thus decoding the message.

Polyalphabetic Cipher

- Accomplished through the use of multiple substitution ciphers
- Blaise De Vigenere, a French diplomat born in 1523, consolidated the cryptographic works of Alberti, Trithemius, and Porta to develop the polyalphabetic cipher
- Because multiple alphabets are used, this approach counters frequency analysis

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Polyalphabetic Cipher

This cipher can be attacked by discovery of the periods when the substitution repeats. The difficulty posed by this cipher is that the same letter in the plaintext does not transform to the same ciphertext letter (different alphabets). For example, if the plaintext was "hello," the first letter l might transform into the letter g and the second letter l might transform into the letter r.

This polyalphabetic substitution was performed in the German Enigma machine.

Crypto History: Battista Cipher Disk

- In Italy, around the year 1460, Leon Battista Alberti developed cipher disks for encryption
- There are two concentric disks
- Each disk had an alphabet around its periphery
- By rotating one disk with respect to the other, a letter in one alphabet could be transformed to a letter in another alphabet

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Crypto History: Battista Cipher Disk

As we go through history, the closer we come to modern times, the more advanced the techniques become. Remember that although these techniques seem simple with the use of computers, they would be considered fairly difficult to develop and crack. Leon Battista took the concept of the Caesar cipher to the next level. Rather than developing a scheme in which each letter was rotated three spaces in the alphabet, why not develop a scheme in which you change the key and rotate a letter any number of spaces? Battista did this by creating two disks, one slightly smaller than the other. He then attached these in the center. Each disk included the letters of the alphabet; by rotating the disks, a certain number of places gave a different value for x in the rotation cipher.

Crypto History: Cryptanalysis

- Because of their expertise in mathematics, statistics, and linguistics (around the 7th century), the Arabs invented cryptanalysis

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Crypto History: Cryptanalysis

A cryptologist is someone who works in the area of encryption. These people specialize in encryption and the mathematics surrounding it. As with any technique, the only way to determine the robustness is to crack it. Determining weaknesses in an algorithm allowed you to build more robust techniques. As obvious as this methodology is, the Arabs—with their extensive knowledge in mathematics—originally came up with it. Although this concept was developed a long time ago, it still serves as a basis for strong crypto today.

Crypto History: Jefferson Disks (1)

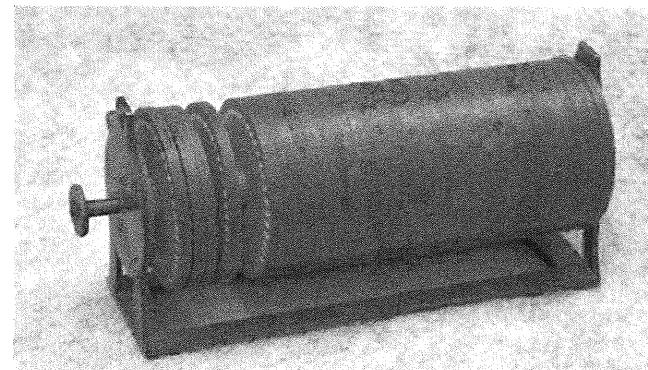
- In 1790, Thomas Jefferson developed an encryption device
- The device contained a stack of 36 disks that could be rotated individually
- A message was assembled by rotating each disk to the proper letter under an alignment bar that ran the length of the disk stack
- The alignment bar was rotated through a specific angle, and the letters under the bar were the encrypted message

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Crypto History: Jefferson Disks (1)

Thomas Jefferson also did extensive work in the area of encryption. He took the ideas Battista and others developed and expanded them into more complex structures. Battista used a two-disk mechanism to encrypt and decrypt information. Thomas Jefferson developed a 36-disk mechanism that was based on the unique letters in the English alphabet.

Crypto History: Jefferson Disks (2)



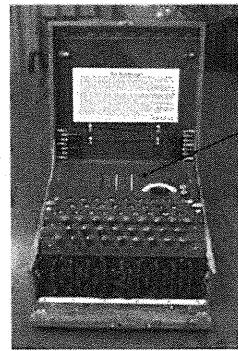
COURTESY OF THE NATIONAL CRYPTOLOGIC MUSEUM)

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

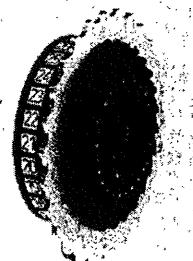
Crypto History: Jefferson Disks (2)

This diagram shows a picture of the machine that Thomas Jefferson created. Essentially it contains 36 disks, each with a unique marking. Aligning the disks in a certain way allowed someone to encrypt a message and decrypt the message. The alignment of the 36 disks served as the key. Even if two people had the same device but not the same key, they would not be able to read an encrypted message.

Crypto History: Enigma



Enigma rotor



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Crypto History: Enigma

The German enigma rotor machine has the following properties:

- Poly-alphabetic substitution cipher machine.
- Used by the Germans in WW II.
- Developed by Dutchman Hugo Koch in 1919.
- Produced for the commercial market in 1923 by Arthur Scherbius.
- Working with the French from 1928 to 1938, Pole Marian Rejewski solved the wiring of the German 3 rotor enigma.
- In 1938, the Germans changed the number of rotors to six.
- The Poles and the French constructed a prototype machine called “the bombe” for use in breaking the enigma cipher. Bletchley park in England took over the work of breaking the enigma cipher.

Crypto History: Hebern Machines

- Rotor systems are also referred to as “Hebern machines”
- Other rotor machines include:
 - The Japanese red and purple machines
 - American sigaba (big machine)
 - Sigaba ciphers were never broken

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Crypto History: Hebern Machines

Any system based on a rotor mechanism, such as Battista and Jefferson, are often referred to as “Hebern machines.”

Other rotor machines include the following:

- The Japanese red and purple machines
- American sigaba (big machine)
 - Sigaba ciphers were never broken.

As with any encryption scheme, the method's robustness is determined by how well it is implemented. Some rotor systems are easily breakable, and others claim that they have never been broken.

Crypto History: One-time Pad

One-time pad:

- The key has the same length as the message
- The key is used only once and never used again
- Ideally, the key's components are truly random and have no periodicity or predictability, thus making the ciphertext unbreakable

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Crypto History: One-time Pad

The one-time pad was invented in 1917 by Major Joseph Mauborgne of the United States army signal corps and Gilbert Vernam of AT&T.

If used properly, that is, the key is truly random, it must satisfy the following: The key is the same length as the data to be enciphered, the key is used only once, and the one-time pad is unbreakable. For large amounts of data, it is sometimes difficult or impossible to satisfy the first two constraints.

Crypto History: Vernam Cipher

- It is a one-time pad
- It is implemented through a key that consists of a random set of non-repeating characters
- Each key letter is added as modulo 26 to a letter of the plaintext
- The key is used only once, and then it is never used again
- The length of the key character stream is equal to the length of the message

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Crypto History: Vernam Cipher

Some consider the Vernam cipher to be unbreakable. This is not entirely true because all encryption is breakable from a brute-force perspective. It might take 800 years to crack, but it is still breakable. With the Vernam cipher, each message is encrypted with a different key, which is referred to as a “one-time pad.” When people say that the Vernam cipher is unbreakable, they are actually saying that the usefulness of the message has expired by the time you crack the key.

Book or Running Key Cipher

- This cipher uses text from a source, such as a book, to encrypt the plaintext
- The key, known to the sender and the intended receiver, might be the page and line number of text in the book
- Text is matched character for character with the plaintext, and modulo 26 addition is performed to affect the encryption

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Book or Running Key Cipher

This type of cipher eliminates periodicity, but it is attacked by exploiting the redundancy in the key.

For example, the sender might tell the receiver to go to the book, *All Quiet on the Western Front*, and go to page 101. Then, the receiver would be instructed to go to the second paragraph and, starting with the first word in that paragraph, use the letters as the key.

Import and Export Issues

- COCOM (Coordinating Committee for Multilateral Export Controls):
 - 17 members
 - 1991 allowed export of encryption
 - Prevent crypto from being exported to dangerous countries
- Wassenaar Arrangement:
 - 1995, 28 countries followed up to COCOM
 - Symmetric crypto free for export
 - Export of other crypto still requires a license
- European Union Controls:
 - Regulated by the Council Regulation (EC) No. 1334/2000
 - Focused on export of encryption
- United States Controls:
 - No import restrictions
 - Signed the Wassenaar Arrangement but has stricter export controls
 - Looser export controls occurred on July 2000:
 - Retail crypto
 - Crypto source code

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Import and Export Issues

COCOM (Coordinating Committee for Multilateral Export Controls):

- 17 members
- 1991 allowed export of encryption
- Prevent crypto from being exported to dangerous countries

Wassenaar Arrangement:

- 1995, 28 countries followed up to COCOM
- Symmetric crypto free for export
- Export of other crypto still requires a license

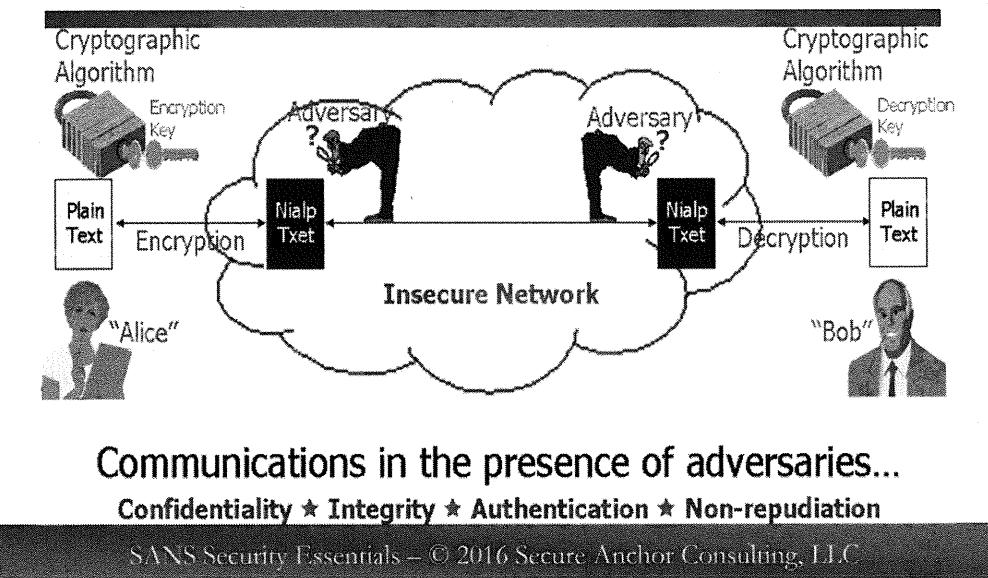
European Union Controls:

- Regulated by the Council Regulation (EC) No. 1334/2000
- Focused on export of encryption

United States Controls:

- No import restrictions
- Signed the Wassenaar Arrangement but has stricter export controls
- Looser export controls occurred on July 2000:
 - Retail crypto
 - Crypto source code

The Challenge



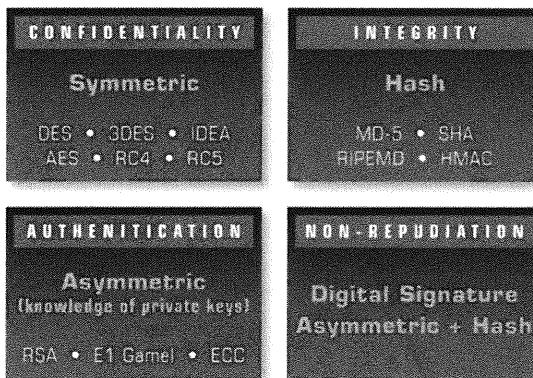
The Challenge

So far, we have discussed the need for cryptography and introduced practical applications in our case studies. We now look at what are the real user requirements.

The diagram portrays the challenge of communicating over an insecure network. Alice and Bob want to exchange information securely. Their cipher is built on basic transformations, permutations, and substitutions. The result of the cipher is that the message is transformed so that, without knowledge of the key used in the system, the message is unreadable. Remember that even if someone knows how the algorithm works, without the key he should still be unable to decipher the message.

Goals of Cryptography

- The goals of a cryptosystem are:



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Goals of Cryptography

Like many of us, Alice does not care *how* the cryptography works, as long as it works. She needs to send a message to Bob with the same level of integrity she would have if she walked up and handed it to him. In addition to being unreadable by adversaries (*confidentiality*), we might have the following requirements:

- **Authentication:** If Alice walks up to Bob and hands him a message, he positively knows the message is from Alice. Alice might require the cryptosystem to provide an equivalent service for her, validating the authenticity of the person with which she is communicating.
- **Integrity:** It should be possible to prove the message has not been tampered with—that this message is exactly the same as the one Alice sent to Bob.
- **Non-repudiation:** The system should provide validation so someone is able to prove in a court of law that Alice, and only Alice, sent the message.

The technology to do this is available, but for this system to work in practice, the non-technical issues are also important. Alice and all users of the system must be trained in its use and its limitations and have access to the keys, yet keep them protected and current. Processes must be as foolproof as practical. Think about *social engineering*, human error, and operator efficiency, accuracy, and understanding.

The Players

In this chapter, we have followed the convention of assigning human names to the participants in secure communications. We give the names "Alice" and "Bob" to two communicating parties.

It is also common practice to use the name "Eve" as the person who is trying to break the encryption or read Alice and Bob's message. Although these names personalize our situations involving crypto, we need to remember that they are just metaphors. Although we might say, "Alice decides to use crypto algorithm X," keep in mind that users of crypto rarely make these kinds of deliberate, conscious choices. Alice probably bought some crypto product that selects a cipher from a set of available ones. The point is that users are generally not encumbered with the details of the cryptography.

Essential Mathematics

Now we turn our attention from Alice and Bob to bits and bytes. Cryptography is a mathematical specialty that includes aspects of probability theory, information theory, complexity theory, number theory, abstract algebra, and more. Our discussion of crypto, however, does not require delving into these fields. Nevertheless, there are a few mathematical operations that are necessary for understanding our subsequent discussion, namely the OR, exclusive OR (XOR), and modulo functions.

General Symmetric Encryption Techniques

Confidentiality

- The goal is to garble the original message so its meaning is concealed
- Basic techniques:
 - Substitution:
 - XOR
 - Arbitrary substitution
 - Rotation
 - Permutation
 - Hybrid
- These techniques are used by symmetric key systems

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

General Symmetric Encryption Techniques

Essential Operations

The main goal of encryption is to garble text so someone cannot understand it. Two basic methods of encrypting or garbling text are *substitution* and *permutation*. A third approach is actually a hybrid, a mixture of both.

Digital Substitution (Encryption)



21 Bit Key
1010011 1010010 1001110



Plaintext In ASCII
C = 1000011
A = 1000001
T = 1010100

XOR Operation:

0 if the compared bits
are the same

1 if they are different

1010011 1010010 1001110
1000011 1000001 1010100

E_K(M) = C

0010000 0010011 0011010

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Digital Substitution (Encryption)

George Boole, a mathematician in the late 1800s, invented a form of logic algebra that provides the basis for electronic computers and microprocessor chips. His logical operations were a set of *truth tables*, in which each of the inputs and outputs were either TRUE or FALSE.

The Boolean Exclusive OR (XOR) function is one of the fundamental operations used in cryptography. The output of an XOR is TRUE if exactly one of the inputs is TRUE; otherwise, the output is FALSE.

Computations require numbers, so we use 0 and 1 instead of TRUE and FALSE. The output of an XOR operation is 0 if both inputs are the same, and the output is a 1 if the two inputs differ.

These properties of XOR make it useful to cryptographers for two reasons. First, any value XORed with itself is 0 ($0 + 0 = 0$, $1 + 1 = 0$). Second, any value XORed with 0 is just itself ($0 + 0 = 0$, $1 + 0 = 1$).

Suppose Alice has a secret message to send to Bob, composed of the three-character message CAT. This translates to a standard 7-bit ASCII bit stream:

1000011 1000001 1010100

Now, suppose that Alice and Bob have already shared the following 21-bit secret key:

1010011 1010010 1001110

Alice converts the plaintext into ciphertext by XORing the message with the key:

$$\begin{array}{r} 1000011 \quad 1000001 \quad 1010100 \\ + \quad 1010011 \quad 1010010 \quad 1001110 \\ \hline 0010000 \quad 0010011 \quad 0011010 \end{array}$$

The output of this algorithm, 0010000 0010011 0011010, now becomes the ciphertext.

Digital Substitution (Decryption)



21 Bit Key
1010011 1010010 1001110

+

Ciphertext
**0010000
0010011
0011010**

Plaintext In ASCII
**C = 1000011
A = 1000001
T = 1010100**

**1010011 1010010 1001110
0010000 0010011 0011010**

D_K(C) = M

1000011 1000001 1010100

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Digital Substitution (Decryption)

Bob receives the ciphertext from Alice and, in turn, XORs it with the secret key:

$$\begin{array}{r} 1010011 \quad 1010010 \quad 1001110 \\ + \quad 0010000 \quad 0010011 \quad 0011010 \\ \hline 1000011 \quad 1000001 \quad 1010100 \end{array}$$

The recovered plaintext is Alice's original message. So XOR naturally acts as a cipher: The original message XORed with a key yields a jumble of bits; XORing that jumble with the key again yields the original message.

Another Boolean function sometimes seen in cryptography is OR. The output of an OR is TRUE if either of the inputs is TRUE; otherwise, the output is FALSE. Using binary digits, the output is a 1 if either or both inputs are a 1; the output is a 0 only if both inputs are 0.

Arbitrary Substitution

- It uses a one-to-one substitution of arbitrary characters
- Given one character mapping, you cannot determine the key, as with rotation substitution
- For example:
 - Plaintext: A B C D E
 - Ciphertext: W K M P D
 - So "CAB" becomes "MWK"
- It is easy to break using character frequency analysis

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Arbitrary Substitution

Substitution involves exchanging one character (or byte) for another. Simple substitution schemes use mapping; therefore, one character is substituted with another character to encrypt a message, with decryption being the inverse action. The mapping function is the key; that is, anyone who knows how the characters were mapped to encrypt the message can decrypt the message.

Consider a simple example. Suppose we define the following map (only a portion of the alphabet is shown):

Plaintext: A B C D E ...

Ciphertext: W K M P D ...

To encrypt the word "CAB," Alice would substitute characters and send the string "MWK." In turn, Bob would reverse the substitution to recover the plaintext.

For substitution to work, there must be a unique one-to-one mapping from plaintext character to ciphertext character. A many-to-one or one-to-many mapping would make decryption difficult or impossible. For example, if W replaced both A and C, you would still be able to encrypt the message; therefore, CAB would become WWK. But if we tried to decrypt it now, we would not know whether the W should be an A or a C because they are both mapped to the same letter.

Rotation Substitution

- It uses a one-to-one substitution of characters
- It "rotates" the alphabet by X characters
- Easy to remember, for example:
 - Plaintext: A B C D E
 - Ciphertext: D E F G H
 - So "CAB" becomes "FDE"
- Caesar Cipher was ROT-3
- Usenet uses ROT-13 (symmetric)

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Rotation Substitution

Introduction

Symmetric encryption is based on simple mathematics that utilize a single key. Whatever can be encrypted with one key can be decrypted only with the same key. Arbitrary substitution requires a mapping for every character in the alphabet. An alternate substitution method that does not require mapping is rotation. In this type of substitution, we shift every character a set number of spaces. For example, if we shift A three spaces, it becomes D, B becomes E, and so on.

Caesar Cipher

The Caesar Cipher, invented by Julius Caesar to encode messages to his generals, is a famous rotation cipher. If Alice were using this "ROT-3" scheme, she would encrypt her message as "FDE." In its day (roughly 50-60 B.C.), the Caesar Cipher was considered good enough to fool almost anyone because very few people could read, even fewer could write, and couriers would rather kill a snooper than let him capture a message. Caesar was no fool, though—he did not use just one encryption tool. He also transliterated Latin into Greek and used other forms of subterfuge.

Though many people believe the Caesar cipher is the earliest cipher, cryptography actually goes back nearly 2000 years earlier to ancient Egypt and China.

ROT-13

Although character rotation is a trivial scheme, rotation ciphers came back into vogue in the early 1980s, primarily in the form of ROT-13. Shortly after USENET newsgroups and electronic mailing lists became popular, subscribers realized they did not always want to see the contents of a message.

Some messages contained jokes that might offend some subscribers. Other messages might contain riddles or puzzles complete with answers that the recipients might not have wanted to see before reading the riddle or puzzle.

The answer was to encrypt (or obscure) jokes and answers using ROT-13. ROT-13 was never meant to be a strong cipher—it is trivial to break. The point was for the reader to make a deliberate effort to decipher the message. No one could later claim accidental discovery, nor could anyone ruin a puzzle by accidentally glimpsing at the solution. ROT-13 eventually became part of newsreader software and a common function of the Unix operating system. ROT-13 had another nice feature. Because there are 26 letters in the English alphabet, ROT-13 is a symmetric operation; the same implementation will both encode plaintext and decode ciphertext. This is because performing ROT-13 followed by ROT-13 is actually ROT-26, which would take you back to the original letter you started with.

It is also important to note that, with rotation, if you figure out the mapping for one character, then you've discovered the entire key. Another flaw with substitution encryption is its predictability. If you use only one set of substitution rules, the encrypted message is easy to crack. Cryptographers responded by inventing more complicated substitution schemes.

Summary

One-to-one character substitution is very weak because they can be defeated with frequency analysis. Cryptanalysts long ago made tables showing the relative frequency with which letters, letter pairs (*bigraphs*), and letter triples (*trigraphs*) appear in a variety of languages. In all character-based languages, some letters occur with a greater frequency than others. In the English language, the letter E occurs approximately 13% of the time, and the letter T occurs approximately 9.3% of the time. So by looking at the enciphered message, we can see which letter appears more often than most, and assume that the enciphered letter is an E. The next most frequently occurring letter would probably be a T, and so on. By looking at letter pairs (instead of just single letters), we can achieve an even more accurate guess.

Permutation

- Keeps the same letters, but changes the position within the text
- Changes the order from xyz to zxy
- For example:
 - Change 1 2 3 4 5 to 3 5 2 1 4
 - So order becomes drroe
- Very easy to break
- Substitution and permutation can be combined together

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Permutation

Permutation, also called *transposition*, shuffles the order in which characters (or bytes) appear rather than substituting one for another. Consider this simple example. Suppose that Alice and Bob chose the key word “SCUBA” to determine the character permutation order. If we alphabetize the letters in the key word, we obtain the string ABCSU. Because A is the first letter, it is assigned the number 1 and U is assigned the number 5; the string 43521 then determines the way in which we will move around letters. Alice takes her message, breaks it into blocks of five characters (because that is the length of the key word), and then moves the characters within each block accordingly.

Unfortunately, permutation is also relatively easy to break. Remember, that a few thousand or million combinations are nothing for a computer, it can defeat an adversary using pencil and paper. Today's computer-based methods still use substitution and permutation, but in combination, applied many times. Let's look at the mechanics of current encryption methods.

General Types of Cryptosystems

The student will have a high-level understanding of the major types of cryptosystems.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

General Types of Cryptosystems

This section intentionally left blank.

Types of Cryptosystems

3 general types of crypto algorithms:

- Symmetric:
 - Secret key
 - Single or 1-key encryption
- Asymmetric:
 - Public key
 - Dual or 2-key encryption
- Hash:
 - One-way transformation

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Types of Cryptosystems

In today's cryptosystems, there are three general types of crypto algorithms: *secret key* or *symmetric*, *public key* or *asymmetric*, and *hash*. Each is used because it provides a different function from other crypto systems. These schemes are usually distinguished from one another by the number of keys employed. The remainder of this section discusses these different types of algorithms.

Secret Key

Symmetric Key Cryptosystems

- aka "secret key" encryption:
 - Fast! Single key for encryption and decryption
 - Requires secure key distribution channel (scalability)
 - No technical non-repudiation
- Requires a secure channel:
 - Pre-shared key
 - Asymmetric encryption
 - Diffie-Hellman key exchange

The diagram illustrates the process of symmetric key cryptography. It shows two parties, "Alice" and "Bob", each with a profile picture. Alice's side shows a "Plain Text" box connected to a "Cryptographic Algorithm" circle, which then outputs "Encrypted Text". This encrypted text is sent via a "SECURE CHANNEL" to Bob's side. On Bob's side, the "Encrypted Text" enters another "Cryptographic Algorithm" circle, which then outputs "Plain Text". Both sides also have a "Key" input arrow pointing to their respective algorithm circles. Below the channels is a dashed line labeled "INSECURE NETWORK". To the right, a box titled "Examples:" lists DES, Triple-DES, RC4, and IDEA.

Examples:

- DES
- Triple-DES
- RC4
- IDEA

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Symmetric Key Cryptosystems

Symmetric key cryptography uses a single key for both encryption and decryption; this key is the shared secret between sender and receiver. Because symmetric key encryption uses only one key for both encryption and decryption, the key must be kept secret and is also referred to as *secret key encryption*. The primary application of symmetric encryption is privacy, where only the parties with the key can encrypt and decrypt messages for each other.

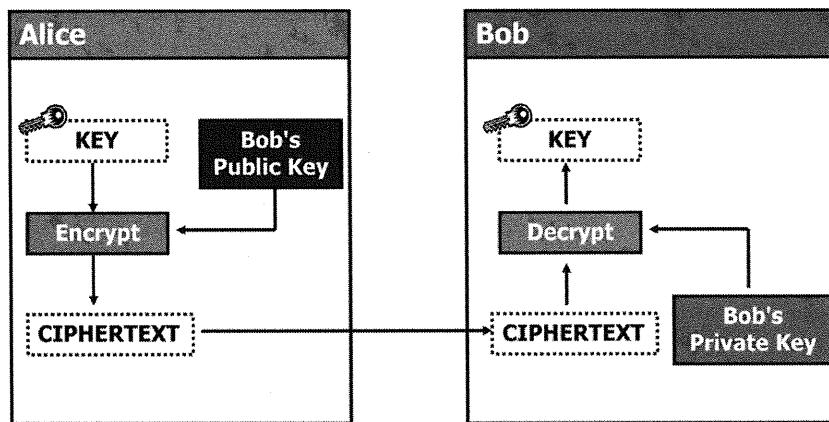
The big issue with secret keys is managing the key creation and exchange to avoid key compromise. Also, the greater the number of parties that share the secret key, the greater the exposure of the key.

The bottom line is this: Because symmetric key cryptosystems are so much faster than asymmetric-key systems but lack the latter's key management and digital signatures, the two are often combined to achieve the best of both worlds.

There are a number of symmetric encryption schemes in common use today, all believed to be mathematically strong. If a cryptanalyst cannot defeat the ciphers by finding a weakness in the mathematical algorithms, then the remaining approach is a brute-force attack to guess all possible keys. Key size does matter, as explained in a paper by Matt Blaze, Whitfield Diffie, Ron Rivest, Bruce Schneier, and others in the cryptographic community. The paper, *Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security* (<http://www.counterpane.com/keylength.html>), describes brute-force attacks that are within the cost and computing means of a variety of attackers, and the key lengths necessary to keep such attackers at bay.

Examples of symmetric encryption schemes in common use today are the Advanced Encryption Standard (AES), Blowfish, the Data Encryption Standard (DES), Triple DES, and the International Data Encryption Algorithm (IDEA).

Asymmetric Key Exchange



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Asymmetric Key Exchange

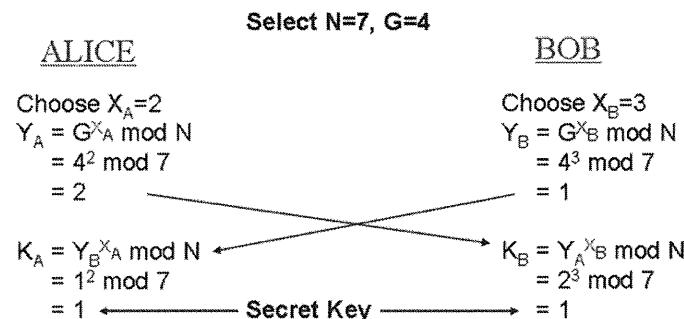
Using public key technology to encrypt messages is rather expensive in terms of computational resources. In layman's terms, asymmetric is really slow and can take a lot of time to encrypt a message depending on the message's size.

If you are going to encrypt a single message that is relatively small, slow performance might not be a concern; however, if you are going to encrypt many large messages frequently, it might be best to use public-key technology to exchange a symmetric key and use a symmetric algorithm to encrypt all those messages. Why? We discussed previously that symmetric algorithms are much faster than asymmetric algorithms, so when encrypting many large messages frequently, symmetric algorithms make more sense performance-wise.

For example, to encrypt a 128-bit symmetric key using an asymmetric algorithm takes considerable less time than to encrypt a message that is perhaps 167 kilobytes, or 1,336,000 bits. To put these sizes in context, the symmetric key is over 10,000 times smaller than the message. Decidedly, it is far better performance-wise to use public key technology to encrypt a relatively small thing such as a symmetric key as opposed to encrypting a document.

Diffie-Hellman Key Exchange

Alice and Bob agree on the value of a large prime number, N and a generator, G. Each calculates a private key (X) and public key (Y). The secret key (K) is derived from X and the other person's Y.



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Diffie-Hellman Key Exchange

Diffie and Hellman first published the concept of two-key crypto in 1976, but it was some time later that they developed the Diffie-Hellman asymmetric algorithm, which is referred to today as the "Diffie-Hellman" and is used only for key exchange. This method provides a mechanism so Alice and Bob can determine the same secret key, even on a network with someone observing all of their communications. Essentially, it allows two parties to exchange a secret key in the presence of an adversary over a nonsecure network.

Alice and Bob start by agreeing on a large prime number, N. They then choose a generator number, G, where $G < N$, and G also meets some other conditions. Alice and Bob then each follow these same steps:

1. Each chooses a large, random number, $X < N$. X is the private key.
2. Each calculates the value $Y = G^X \bmod N$. Y is the public key and is sent to the other party.
3. Each computes the secret key $K = Y^X \bmod N$, where Y' is the other party's public key.

Note that each party's Y is openly shared, but X is kept secret; these are the public and private keys, respectively. For that matter, N and G might also be well known. This scheme works because the secret key values (K) that Alice and Bob compute independently are the same; namely, $K = G^{XX'} \bmod N$, where X is their own private key and X' is the other party's private key (derived from the value of Y'). Because both X values are private, an eavesdropper cannot discover K except by brute-force methods. And if N is large enough, this cannot be accomplished in a *reasonable amount of time*.

The figure shows a Diffie-Hellman example where $N=7$ and $G=4$. As shown, Alice and Bob choose private key (X) values of 2 and 3, respectively, from which they calculate public key (Y) values of 2 and 1, respectively. After swapping their Y values, both independently compute a secret key (K) value of 1.

This scheme works because Alice and Bob are using the same computation to calculate the secret key, namely:

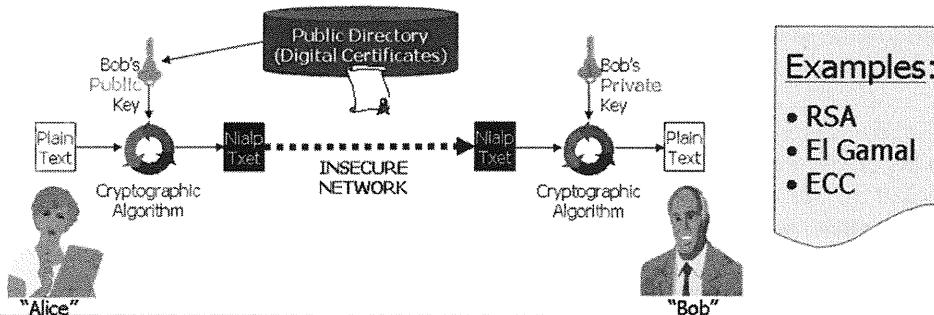
$$\begin{aligned} K &= Y^X \bmod N \\ &= (G^{X^t} \bmod N)X \bmod N \\ &= G^{XX^t} \bmod N \end{aligned}$$

Here, X and Y are their own private and public keys, and Y^t is the other party's public key. In this example, Alice and Bob used $N=7$, $G=4$, and two private keys with the values 2 and 3; therefore, their shared secret key should be $4^{2*3} \bmod 7 = 4096 \bmod 7 = 1$, which is exactly what they both got.

Public key

Asymmetric Key Cryptosystems

- "Public key" encryption:
 - Slow! Public/private key pair
 - Public keys widely distributed within digital certificates
 - Used as a secure channel for symmetric key exchange
 - Technical non-repudiation via digital signatures



Examples:

- RSA
- El Gamal
- ECC

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Asymmetric Key Cryptosystems

The management problems associated with symmetric keys are so overwhelming that they virtually preclude their use by themselves in e-commerce. But we can use public key computation to develop a shared message key. Also, algorithms like Diffie-Hellman can be used to exchange a secret key. Again, the general idea is to exchange keys securely, perhaps only once, to secure a given session, such as a visit to a Web page to execute a credit card transaction.

Public key cryptography or *asymmetric encryption* methods have two keys: one used for encryption and the other for decryption. From a mathematical standpoint, anything that is encrypted with one of the keys can be decrypted only with the other key. Asymmetric encryption has many applications, but the primary ones today are key exchange (for symmetric encryption), authentication, and non-repudiation.

Stanford University professor Martin Hellman and graduate student Whitfield Diffie first described modern asymmetric encryption publicly in 1976. Their paper described a two-key cryptosystem in which two parties could engage in a secure communication over a non-secure communications channel without sharing a secret key. The mathematical trick of asymmetric encryption depends on the existence of so-called *trapdoor functions*, or mathematical functions that are easy to calculate, whereas their inverse is difficult to calculate. Here are two very simple examples:

- **Multiplication versus factorization:** Multiplication is easy; given the two numbers 9 and 16, it takes almost no time to calculate the product of 144. But factoring is harder; it takes longer to find all of the pairs of integer factors of 144, and then to determine the *correct* pair that was actually used.
- **Exponentiation versus logarithms:** It is easy to calculate, for example, the number 3 to the 6th power to find the value 729. But given the number 729, it is much harder to find the set of integer pairs, x and y, so that $\log_x y = 729$ and then, again, to determine that pair was actually used.

The previous examples are trivial, but they are examples of the concept; namely, the ease of multiplication and exponentiation versus the relative difficulty of factoring and calculating logarithms, respectively. Actual asymmetric encryption algorithms use integers that are prime and can be several hundred digits in length. Multiplying two 300-digit primes, for example, yields a 600-digit product; finding the two prime factors of a 600-digit number is beyond the capabilities of today's known methods. In this case, then, factoring is said to be *intractable* because of the difficulty of solving the problem in a timely fashion.

Keys are derived in pairs and are mathematically related, although knowledge of one key by a third-party does not yield knowledge of the other key. One key is used to encrypt the plaintext, and the other key is used to decrypt the ciphertext; it does not matter which key is applied first, but both keys are required for the process to work.

One of the keys is designated as the *public key* and may be advertised as widely as the owner wants. The other key is designated as the *private key* and is never revealed. If Alice wants to send Bob a message, she merely encrypts the plaintext using Bob's public key; Bob decrypts the ciphertext using his private key.

This two-key scheme can also be used to prove who sent a message. If Alice, for example, encrypts some plaintext with her private key, Bob (or anyone else) can decrypt the ciphertext using Alice's public key. The benefit here is that Bob (or whoever successfully decrypts the ciphertext) knows for sure that Alice encrypted the message (authentication), and Alice cannot subsequently deny having sent the message (non-repudiation).

In the real world, how are these asymmetric key systems used? They are typically used to perform key exchange for symmetric key algorithms.

Bottom line: Despite being much slower than symmetric-key cryptosystems, asymmetric-key systems are widely used because of their powerful key management and digital signatures—often in concert with symmetric key systems to attain the best of both worlds.

Who Invented Public Key Crypto?

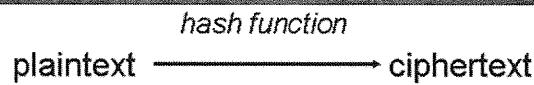
The true history of asymmetric encryption—and answering the question of its invention—is somewhat murky. There is no question that Diffie and Hellman were the first to publicly publish on the topic. Their classic paper, “New Directions in Cryptography,” appeared in the November 1976 issue of *IEEE Transactions on Information Theory*. Diffie and Hellman were not trying to solve the key exchange problem, *per se*, but were trying to make the problem obsolete by inventing a scheme that used a split key; that is, one key for encryption and a second key for decryption. They published their *concept* of split key crypto, but did not identify a function that would work. Rivest, Shamir, and Adleman described their implementation in the paper, “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems,” which was published in the February 1978 issue of the *Communications of the ACM (CACM)*.

Some sources, however, credit Ralph Merkle as the first to describe a system that allows two parties to share a secret using what is now called a “Merkle Puzzle.” His early work was largely misunderstood, and although he submitted a paper to *CACM* some years earlier, his description did not appear until April 1978. He certainly was not the first to publish, but did he have a workable idea before Diffie and Hellman?

The true invention of public key cryptography probably does not belong to anyone in the United States, however. The article, “The Open Secret,” in the April 1999 issue of *WIRED Magazine* reports that asymmetric encryption was probably first invented by James Ellis of the UK's Government Communications Headquarters (GCHQ) in 1969. Ellis' work was classified until the late 1990s, so there was no public mention of it, and it is possible that Ellis influenced the work of Diffie and Hellman. The U.S. National Security Agency (NSA) claimed to have knowledge of this type of split key crypto as early as 1966, but there is no known documentation.

Hash Functions

Message digests or One-Way encryption



- No key used during encryption:
 - Irreversible one-way transformation
 - The key length is the hash length
 - Plaintext (and length of plaintext) is not recoverable from the ciphertext
 - Examples: MD2, MD4, MD5, RIPEMD-160, SHA-1, and SHA-2:
 - Some algorithms have issues with predictable collisions
 - Also called “message digests” or “one-way encryption”
- Primary use: Message integrity

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Hash Functions

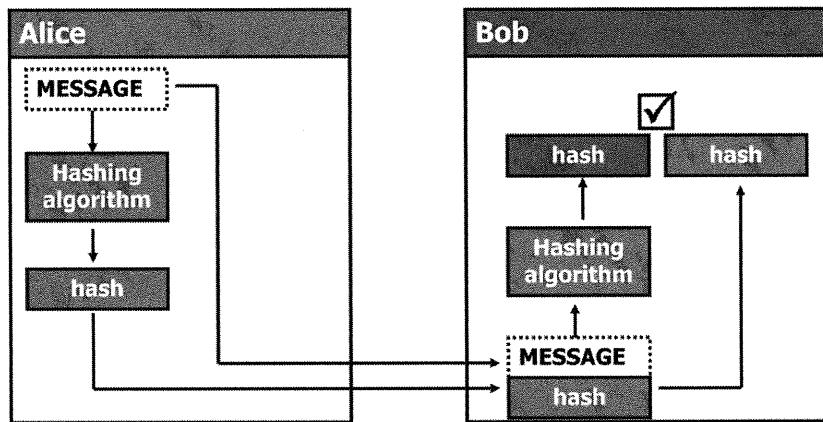
There are three types of cryptography algorithms: secret key, public key, and hash functions. Unlike secret key and public key algorithms, *hash functions*, also called *message digests* or *one-way encryption*, have no key. Instead, a fixed-length hash value is computed based on the plaintext that makes it impossible for either the contents or *length* of the plaintext to be recovered.

The primary application of hash functions in cryptography is message integrity. The hash value provides a digital fingerprint of a message's contents, which ensures that the message has not been altered by an intruder, virus, or by other means. Hash algorithms are effective because of the extremely low probability that two different plaintext messages will yield the same hash value.

There are several well-known hash functions in use today:

- **Message Digest 2 (MD2):** Byte-oriented, produces a 128-bit hash value from an arbitrary-length message, designed for smart cards.
- **MD4:** Similar to MD2, designed specifically for fast processing in software.
- **MD5:** Similar to MD4 but slower because the data is manipulated more. Developed after potential weaknesses were reported in MD4.
- **Secure Hash Algorithm (SHA):** Modeled after MD4 and proposed by NIST for the Secure Hash Standard (SHS), produces a 160-bit hash value. This was published by NIST as FIPS PUB 180-1. NIST superseded PUB 180-1 with FIPS PUB 180-2, released Aug 1, 2002. The -2 release provides three more SHA algorithms: SHA-256, SHA-384, and SHA-512. For more on this standard, see: <http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf>

Integrity



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Integrity

In the diagram, Alice intends to send a message to Bob. Alice's computer is represented by an orange box bearing Alice's name, whereas Bob's computer is represented by a green box bearing his name. In this scenario, Alice is concerned only in the message's integrity. Alice is not concerned with third-party eavesdropping.

Alice inputs the message into a hashing algorithm of her choice. The hashing algorithm outputs a hash of the message. Alice transmits both the original message and the fingerprint to Bob. Again, message integrity is Alice's primary concern, not message confidentiality. Finally, Alice informs Bob of the hashing algorithm she used. Informing Bob of the hashing algorithm is important so Bob can confirm message integrity.

Upon receiving the message and hash, Bob employs the same hashing algorithm as Alice. Bob inputs the message to the hashing algorithm, which generates a fingerprint. Bob then compares this hash against the received fingerprint. If both hashes are identical, Bob confirms Alice's message was unaltered in transit. If the hashes do not match, Bob knows something occurred during transmission.

Digital Signatures

- Digital signatures use public key cryptography to "sign" documents
- The signatures are probably authentic
- The signatures are nonrepudiable
- They "sign" a document by encrypting a one-way hash with a private key

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Digital Signatures

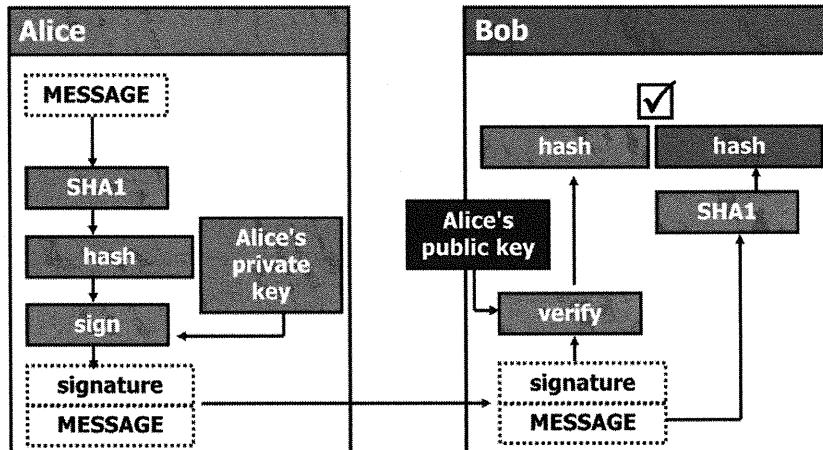
Semantically, digital signatures are equivalent to signatures affixed to documents with pen and ink: The signature is meant to identify the signer uniquely.

Because pen and ink are useless in an electronic environment, cryptography, specifically asymmetric algorithms, are used to provide the required uniqueness. Handwritten signatures have long held protected legal status as an official recognition of approval on a paper document (despite the fact that handwritten signatures are notoriously easy to forge). In this digital age, it seems only fair that we have a method of signing electronic documents that is unique as well as difficult to forge.

Public key cryptography allows users to employ their private key to encrypt data, in effect *signing* the data. Because a given private key is intended for one and only one owner, use of the private key in encryption unmistakably associates the user's identity with the encrypted data. This is semantically equivalent to the user hand-signing the data.

Recipients of the signed data employ the user's public key to decrypt and, therefore, verify the sender's signature. In short, by leveraging asymmetric algorithms, users can establish a person's digital signature is authentic. In addition, users can also establish authenticity even if the sender denies having signed the data. This is called *non-repudiation*.

Digital Signature Example



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Digital Signature Example

In this figure, Alice intends to send a message to Bob. An orange box bearing her name represents Alice's computer and a green box also bearing his name represents Bob. As in the hashing algorithm example, Alice creates a unique fingerprint of the message she intends to send to Bob. In this case, Alice chooses SHA-1 as the hashing algorithm. Alice also wants to digitally sign the document for two reasons:

- To protect the hash from modification during transmission
- To prove to Bob that she sent the message

To digitally sign the hash, Alice employs her private key to encrypt the hash. Because Alice is supposed to be the only owner of her private key, use of the private key binds her identity to the hash. Alice then sends both the plaintext message and the signature to Bob. As in the hashing algorithm example, Alice's concern is for the integrity of the message, not necessarily the message's confidentiality.

To mitigate an attacker who is intercepting, altering, and creating a new hash of her message, Alice signs the hash—that is, cryptographically safeguards it from modification.

Upon receiving the message and Alice's digital signature, Bob employs Alice's public key to decrypt the signature. By decrypting the signature, Bob retrieves the hash of the message generated by Alice.

Bob then generates a fingerprint of the received message and compares it to the received hash. If both hashes are identical, then Bob knows two things:

- The message did not change in transmission.
- The message was sent by Alice.

In addition, Alice cannot deny she sent the message because only her public key could decrypt the signature—assuming, of course, that no one stole her key.

Here is the problem: How does Alice know whether someone stole her private key? Digital signatures in and of themselves aren't very secure. What? It's not that the cryptography used in digital signatures is weak or somehow substandard; its how the cryptography is employed that matters. The issue relates to the implementation concept we discussed at the beginning of Module 13.

How do we know that Alice's public key really belongs to Alice? What if somebody is impersonating Alice? Can we be sure? How did we get Alice's public key—from a server, from e-mail? How do we know it hasn't been corrupted?

As you can see from these few simple questions, cryptography isn't the issue here; it is both the infrastructure and semantic assumptions people make in the usage of the cryptography. These quandaries lead us directly into the brick wall known as the public key infrastructure (PKI).

Summary

- Encryption plays a critical part in the protection of information
- Most systems that employ encryption use all three types of encryption:
 - Symmetric
 - Asymmetric
 - Hash

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Summary

Cryptography, the science of secret writing, is an essential component of computer and network security at all levels. Information security professionals must be comfortable with at least the basic terms and concepts associated with this field so that they can understand products, services, and vendor claims.

Although the crypto methods used today are vastly stronger and more complex than algorithms used even 30 years ago, the same two fundamental operations still form the basis of symmetric encryption schemes used to encrypt messages for privacy, namely *substitution* and *permutation*. Substitution is the method of replacing, or substituting, characters in a message with other characters, whereas permutation (transposition) moves characters around within the message. Today's algorithms tend to employ many rounds of both.

Crypto schemes that operate on a single bit or byte at one time are usually called *stream ciphers*, whereas those that work on larger collections of bits and bytes are called *block ciphers*. Block ciphers are most common, although there are a number of stream ciphers used in the field.

A crypto key governs the transformation of the plaintext into ciphertext. Modern crypto algorithms can be broadly classified into three categories based on the number of keys employed and the goals they accomplish. Each of these methods is used for specific applications. The categories are:

- Symmetric encryption algorithms use a single key for both encryption and decryption. Key lengths between 128 and 256 bits are generally thought to be adequate; shorter keys are deemed weak. Common algorithms such as AES (Rijndael), DES, 3DES, IDEA, RC4, and RC5 are used for privacy.
- Asymmetric encryption algorithms use a pair of very large, mathematically related keys. Asymmetric encryption uses a two-key system, whereby one of the keys is used to encrypt data, and the other is used for decryption. This depends on the existence of so-called *trapdoor functions* that are easy to calculate, whereas the inverse function is very difficult (intractable).

With trapdoor functions, one key does not yield knowledge of the other key. One of the keys, therefore, can be widely distributed and is called the *public key*; the other key is kept secret and is called the *private key*. Common schemes such as Diffie-Hellman, RSA, and ECC might be used for such functions as key exchange, user authentication, and digital signatures. RSA is a public key algorithm invented in 1977 by Rivest, Shamir, and Adleman. RSA and Elliptic Curve Cryptography (ECC) are discussed further in the next section.

- Hash functions are one-way encryption; they employ no key, and the hash operation cannot be reversed to recover the original plaintext from the hash value. Hash functions such as SHA are used for message integrity.

In today's environment, it is rare to find only one of these algorithms in use; it is far more common to find a set of these protocols used together to form a cryptosystem. PGP is such a cryptosystem, and can provide privacy, message integrity, and authentication for e-mail applications. In the same manner, SSL/TLS is used as a cryptosystem for secure e-commerce transactions.

Although cryptography is necessary for security, it is not sufficient by itself. There are bad crypto schemes, bad implementations of good crypto schemes, and misuse of good implementations. Just as security is a process, so is the management and use of crypto; thus, security administrators—and users—need to be trained in the art of cryptography.

Module 19: Encryption 102

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Module 19: Encryption 102

This section intentionally left blank.

Encryption 102

SANS Security Essentials IV: Secure Communications

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Introduction: Encryption 102

This section intentionally left blank.

Objectives

- Crypto concepts
- Symmetric and asymmetric cryptosystems
- Crypto attacks
- Steganography (stego)

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Objectives

The previous module gave you a tour of some of the important issues and concepts in the field of cryptography. It showed that encryption is real, it is crucial, it is a foundation of much that happens in the world around us today, and most of all, it is completely transparent to us.

One of this section's goals is to show you how some of the world's most popular ciphers, which are in constant use in many different sectors, operate under the covers. Along the way, we share some pragmatic lessons we learned the hard way, hoping our experience will be of help to you in the future.

We begin by examining the conceptual underpinnings of today's major ciphers. In particular, we discuss Triple DES, a good alternative for the now obsolete and relatively short key length DES algorithm. Then, we discuss the new standard for encryption, the Advanced Encryption Standard (AES).

Our next stop is the widely deployed RSA public-key algorithm.

We wrap up the section with an overview of emerging Elliptic Curve Cryptosystems (ECC). ECCs, whose processing power and storage requirements are relatively low, are rapidly growing in popularity due to the proliferation of small electronic devices with limited power and storage.

Crypto Concepts

The student will have a high-level understanding of the mathematical concepts that contribute to modern cryptography.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Crypto Concepts

This section intentionally left blank.

Concepts in Cryptography (1)



- What if...
 - We can find a mathematical "problem" that exhibits characteristics of one-way functions (with trapdoors)? or, as mathematicians would prefer to say, a problem that is "impossible" to solve in polynomial time?
- Hmm...
 - We could use it to build a new cryptosystem!

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Concepts in Cryptography (1)

The previous module defined the four main goals of a cryptosystem: *confidentiality*, *integrity of data*, *authentication*, and *non-repudiation*. However, how do we construct a cipher that enforces these characteristics? Mathematics has fields such as probability theory, information theory, complexity theory, number theory, abstract algebra, and finite fields that are all rich in ideas that can contribute to our cipher.

The previous section also introduced one-way mathematical functions. Such functions can have trapdoor properties that make them well suited for public key cryptography, in which the trapdoor allows a message to be decrypted using a different key than the one used to encrypt the message. If the public key were used to encrypt the message, the trapdoor in this case is the corresponding private key.

One-way functions that are computationally hard—that is, impossible to solve in polynomial time—can make things very difficult for an adversary eavesdropping on our communications, say over an insecure public network like the global Internet. At the same time, the existence of a trapdoor can be used to provide an easy solution to the intractable problem for use by the sender or the recipient.

Concepts in Cryptography (2)

Computational Complexity deals with time and space requirements for the execution of algorithms.

Problems can be **classified** as tractable or intractable.



Tractable Problems

"Easy" problems. Can be solved in polynomial time (i.e., "quickly") for certain inputs

Examples:

- constant problems
- linear problems
- quadratic problems
- cubic problems

Intractable Problems

"Hard" problems. Cannot be solved in polynomial time (i.e., "quickly")

Examples:

- exponential or super-polynomial problems
- factoring large integers into primes (RSA)
- solving the discrete logarithm problem (El Gamal)
- computing elliptic curves in a finite field (ECC)

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Concepts in Cryptography (2)

Mathematics is filled with intractable problems. So a cipher designer can start by just picking one and trying it out. Evaluating an algorithm's *computational complexity* reveals the time and space required to execute it and helps us classify the problem as either tractable (easy) or intractable (hard).

When computers are used to solve problems, we don't care about the exact number of operations—we are more interested in how the amount of input to the problem (or program) affects the number of operations it takes to solve (or execute). *Big-O notation* is used to give a general idea of how many operations a problem takes relative to the input size n . The big-O function isn't usually specifically defined; it is mostly used as a notational shorthand to indicate a problem's complexity.

Relatively easy problems (symmetric encryption) can be solved in *polynomial time* —that is, the relationship between the input size and the number of operations required to solve the problem is constant, linear, quadratic, cubic, and so on. *Constant time*, $O(1)$, means they take the same number of operations to solve regardless of the input size. *Linear time*, $O(n)$ means the number of operations increases linearly with the input size—when the input size is doubled, the problem takes twice as long to solve. *Quadratic time* is $O(n^2)$, *cubic time* is $O(n^3)$, and so on.

Problems are considered intractable (or hard) when they cannot be solved in polynomial time (asymmetric encryption). Examples are *exponential*, $O(2^n)$ and *superpolynomial* (somewhere between polynomial and exponential), which are considered so complex as to be hard or intractable. A cubic-time algorithm might take thousands of years to solve, whereas an exponential-time algorithm might take longer than the universe is expected to last.

It can be hard to prove whether a problem is intractable or not. Someone might prove a particular problem can be solved in superpolynomial time, only to have someone later discover it can be solved a different way in polynomial time. So it is more accurate to state that the problems we use in cipher algorithms are *believed* to be intractable by most researchers in complexity theory.

There's always the highly unlikely chance that easier solutions have been overlooked or just haven't been discovered yet.

Three well-known examples of intractable problems include factoring large integers into their two prime factors (the basis for RSA), solving the discrete logarithm problem over finite fields (the basis for El Gamal), and computing elliptic curves over finite fields (the basis for Elliptic Curve Cryptosystems). Now, let's examine each of these three important classes of intractable problems in greater detail, because each one of these forms the basis of important cryptosystems, which are widely used all over the world today.

Concepts in Cryptography (3)

An Example of an Intractable Problem...

Difficulty of factoring a large integer into its two prime factors

- A “hard” problem
- Years of intense public scrutiny suggest intractability
- No mathematical proof so far

Example: RSA

- based on difficulty of factoring a large integer into its prime factors
- ~1000 times slower than DES
- considered “secure”
- *de facto* standard
- patent expired in 2000

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Concepts in Cryptography (3)

Factoring integers doesn't seem that hard. It doesn't take much thought to figure out that 15 can be factored into 1×15 and 3×5 . So why is it on our list of intractable problems?

The operative word here is “large.” The larger the integer, the harder it is to factor. In fact, there is no known recipe for factoring other than trial and error: keep multiplying primes together until you arrive at the number. Remember that even though most researchers in complexity theory believe factoring large integers is a hard problem, there is no unequivocal proof to that effect. It is only the years of public scrutiny of the problem that lead us to conclude the problem cannot be solved in polynomial time.

Perhaps the most popular public key algorithm today, RSA, takes advantage of the intractability of the integer factorization problem. We discuss RSA in depth later in the section.

Concepts in Cryptography (4)

Another Intractable Problem...

Difficulty of solving the discrete logarithm problem - for finite fields

- A "hard" problem.
- Years of intense public scrutiny suggest intractability.
- No mathematical proof so far.
- The discrete logarithm problem is as difficult as the problem of factoring a large integer into its prime factors.

Examples

- El Gamal encryption and signature schemes
- Diffie-Hellman key agreement scheme
- Schnorr signature scheme
- NIST's Digital Signature Algorithm (DSA)

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Concepts in Cryptography (4)

Another intractable problem is the *discrete logarithm problem for finite fields*. The discrete logarithm is based on a statement of the form $a^x \bmod n = b$, where a , b , n , and x are integers and a and n are known. The *mod* operator just means we take the remainder of the first number (a^x) when divided by the second number (n). Finding b when we know x is easy, but not the other way around.

For example, it is easy to calculate $8^3 \bmod 7$ - because $8^3 = 512$ and the next lowest multiple of 7 is 511, the remainder must be $512 - 511 = 1$. But it takes trial and error to discover that $8^x \bmod 7 = 1$ is satisfied only by $x = 3$. This problem is the discrete logarithm. Just as with prime factorization, the problem *really* gets hard when x is a hundred- or thousand-bit number.

Again, the notion that discrete logarithms are intractable is the consensus of computational complexity researchers, and there is no unequivocal proof that this problem cannot be solved easily. It is the years of public scrutiny of the problem that leads us to conclude that it is a hard problem that cannot be solved in polynomial time. You can prove an algorithm is not secure by breaking it; you just cannot prove an algorithm is secure. But how does it compare with the previous intractable problem we looked at—the factorization of large integers into two primes? Evidence shows the discrete logarithm problem is just as difficult.

So we should be able to use the discrete logarithm problem in building a cipher. In fact, several ciphers in use today are built upon the intractability of the discrete logarithm problem over finite fields: the El Gamal encryption and signature schemes, the Diffie-Hellman key agreement scheme, the Schnorr signature scheme, and the Digital Signature Algorithm (DSA) by the U.S. Department of Commerce's National Institute of Standards and Technology (NIST).

Concepts in Cryptography (5)

Yet Another Intractable Problem...

Difficulty of solving the discrete logarithm problem
--as applied to elliptic curves

- A “hard” problem
- Years of intense public scrutiny suggest intractability
- No mathematical proof so far
- In general, elliptic curve cryptosystems (ECC) offer higher speed, lower power consumption, and tighter code.

Examples

- Elliptic curve El Gamal encryption and signature schemes
- Elliptic curve Diffie-Hellman key agreement scheme
- Elliptic curve Schnorr signature scheme
- Elliptic Curve Digital Signature Algorithm (ECDSA)

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Concepts in Cryptography (5)

The ciphers we named in the previous section use the discrete logarithm problem, but only for certain sets of numbers that belong to what are known as *finite fields*. This problem also makes for a good cipher algorithm when applied to *elliptic curves*.

This class of problem is considered every bit as intractable as the previous two. Plus, it lends some additional useful features to our algorithm: high security levels even at low key lengths, high speed processing, and low power and storage requirements. These characteristics are useful in crypto-enabling the many new devices that are rapidly appearing in the marketplace—for example, mobile telephones, information appliances, smart cards, and even the venerable ATM.

Symmetric, Asymmetric and Hashing Cryptosystems

The student will have a basic understanding of commonly used symmetric, asymmetric, and hashing cryptosystems.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Symmetric, Asymmetric, and Hashing Cryptosystems

This section intentionally left blank.

DES

- Data Encryption Standard
- Released March 17, 1975
- Rather fast encryption algorithm
- Widely used; a de facto standard
- Symmetric key, 64-bit block cipher
- 56-bit key size: Small 2^{56} keyspace
- Today, DES is not considered secure

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

DES

DES was the most commonly used encryption algorithm in the world. The U.S. government proposed its adoption as a national standard on March 17, 1975, for use with unclassified computer data. Based on IBM's Lucifer cipher, DES is specified in Federal Information Processing Standard (FIPS) 42. The American National Standards Institute (ANSI) adopted DES as a standard (ANSI X3.92) in 1981, calling it the "Data Encryption Algorithm (DEA)."

Due to the internal bit-oriented operations in the design of DES, software implementations are slow, whereas hardware implementations are faster. The National Institute of Standards and Technology (NIST) standardized four different DES operation modes for use in the United States: electronic codebook (ECB) mode, cipher block chaining (CBC) mode, output feedback (OFB) mode, and cipher feedback (CFB) mode.

DES Weaknesses

- DES is considered non-secure for very sensitive encryption. It is crackable in a short period of time
- See the book, *Cracking DES*, by O'Reilly
- Multiple encryptions and key size increase the security
- Double DES is vulnerable to the meet-in-the-middle attack and only has an effective key length of 57 bits
- Triple DES is preferred

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

DES Weaknesses

From the beginning, concerns were raised about the strength of DES, due to the rather small key length of 56 bits (a 64-bit ciphertext block minus 8 bits for parity), resulting in a keyspace containing only 2^{56} possible different keys. The effectiveness of attacks based on brute-force searches depends upon keyspace size. Because of DES's relatively small keyspace, brute-force attacks are feasible. DES was first (publicly) cracked in the RSA Challenge, a program that offers monetary rewards for breaking ciphers and solving computationally intensive mathematical problems. The DES challenge took only five months for the public to solve, and subsequent attempts are taking less and less time.

Consequently, DES is no longer considered secure because of its key size, and not because the algorithm has been broken. In fact, anyone can build a DES-cracking engine these days. All the information you need, including sample code, is available in a book called *Cracking DES*. However, with the global e-commerce infrastructure build-out proceeding at a furious pace, due to all the new e-business initiatives that are sprouting up all over the world, the need for a fast, symmetric block cipher is extremely urgent. If DES can no longer be considered to be secure, what can we do in the interim?

Again, DES was already widely deployed in both hardware and software products, and it had withstood unbridled cryptanalysis for decades. It didn't take long to realize what a great advantage it would be to somehow increase DES's key size and use the existing implementations until a new standard was built.

One way to effectively increase the key length is to perform the encryption more than once. That is, encrypt the cleartext, and then encrypt the resulting ciphertext, and so on. But this only works if the cipher algorithm is not a *group*.

DES Advantage

- In 1992, it was proven that DES is not a group. This means that multiple DES encryptions are not equivalent to a single encryption. THIS IS A GOOD THING!
- If something is a group, then:
 - $E(K2, E(K, M)) = E(K3, M)$
- Because DES is not a group, multiple encryptions will increase the security

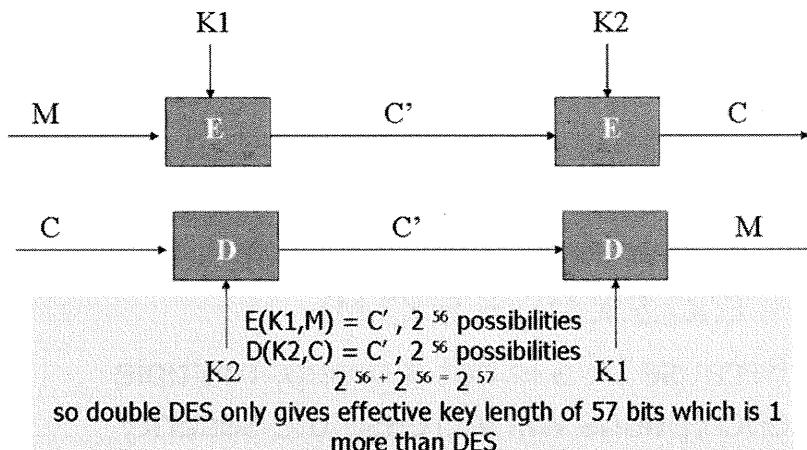
SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

DES Advantage

The function E is a group if $E(K2, E(K, M)) = E(K3, M)$. In other words, encrypting once with key K and then again with key $K2$ is equivalent to encrypting once with $K3$. Thus, if a cipher algorithm is a group, encrypting multiple times is no stronger than encrypting once.

Whether an algorithm is a group is an important statistical consideration. If it is a group, then applying the algorithm multiple times is a waste of time. In 1992, it was proven that DES is not a group, in fact, so encrypting multiple times with DES is not equivalent to encrypting once. That's good news; it means that encrypting more than once with DES can increase the security of the ciphertext.

Meet-in-the-Middle Attack



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Meet-in-the-Middle Attack

But encrypting twice with DES (Double DES) does not increase the effective key size significantly. If a cryptanalyst is able to obtain both a cleartext message (M) and its corresponding ciphertext (C), she can perform a *meet-in-the-middle attack*.

We already mentioned that brute-force attacks on DES are feasible, which means we can attempt to decrypt a message with every possible key until we find the one that gives us sensible cleartext. For a meet-in-the-middle attack, we first encrypt the cleartext M with every possible key K_1 :

$$C' = E(K_1, M)$$

giving us 2^{56} values of intermediate ciphertext C' . Then, we decrypt the ciphertext C with every possible key K_2 :

$$C' = D(K_2, C)$$

again giving us 2^{56} values of C' . The values of K_1 and K_2 that yield the same C' in the previous equations are the two keys used for the double DES encryption. The number of operations and, therefore, the resulting key length, is only $2^{56} + 2^{56} = 2^{57}$. This gives us an effective key length of only 57 bits, which is only 1 more bit than DES.

Triple DES

USAGE

Supported in latest releases of web clients, such as Microsoft Internet Explorer.

Prefer Triple DES over DES (which is—officially—no longer considered to be secure).

VULNERABILITIES

Cracking Triple DES means examining all possible pairs of crypto-variables.

So far, there have been no public reports claiming to have cracked Triple DES....

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Triple DES

To thwart meet-in-the-middle attacks, Triple DES adds a third round of encryption. Thus, when performing the two steps of the meet-in-the-middle attack, a cryptanalyst ends up with two sets of ciphertext that won't be comparable—they are separated by another encryption step. Triple DES is well-known and widely implemented, and it has been intensely scrutinized by the global community of cryptologists. Furthermore, it uses the same tried and true DES algorithm, and all existing DES implementations can be used to perform Triple DES.

Support for Triple DES is built right into popular web clients, such as Microsoft Internet Explorer, Firefox, Chrome, and Safari. Using Triple DES from the end-user perspective is often as simple as reconfiguring the web browser's SSL version 3 support to prefer Triple DES over DES. SSL version 2 should be disabled altogether, or else the user could unwittingly communicate with a site using only DES, and those communications could be intercepted.

Triple DES can be configured to use either 2 or 3 unique keys, yielding a key strength of either 112 bit (2 Keys) or 168 bit (3 Keys).

Triple DES conducts three passes of the DES algorithm. There is a concept of a round that represents the number of iterations within the algorithm. Each encryption algorithm has its own specification regarding number of rounds—DES uses 16 rounds. To appreciate the extra effort required to use Triple DES, the standard DES algorithm is executed 16 times (rounds), whereas the Triple DES is executed 48 times (rounds). Thus, Triple DES will require three times the amount of resources to perform the encryption and decryption.

AES (1)

- Advanced Encryption Standard
- A new encryption algorithm(s) that is being designed to be effective well into the 21st century

THE FIVE "AES" FINALISTS !

• MARS	IBM
• RC6™	RSA Laboratories
• Rijndael	Joan Daemen, Vincent Rijmen
• Serpent	Ross Anderson, Eli Biham, Lars Knudsen
• Twofish	Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niels Ferguson

Significance

Developing "good" cryptographic algorithms that can be trusted is hard. The only practical way to develop such algorithms is to perform the development process in an open manner, and under intense public scrutiny of the global cryptographic community. Can you think of a recent example in which this was not followed?

Countdown to AES !

- 1/2/1997, the quest for AES begins...
- 8/9/1999, five finalist algorithms announced
- Announced winner – Rijndael
- 12/26/2001 – AES approved!

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

AES (1)

On January 2, 1997, NIST announced the initiation of an effort to develop the Advanced Encryption Standard (AES). A formal call for algorithms was made on September 12, 1997. The call stipulated that AES must specify unclassified, publicly disclosed encryption algorithm(s), available royalty free, worldwide. In addition, the algorithm(s) would implement symmetric key cryptography as a block cipher and (at a minimum) support a block size of 128 bits and key sizes of 128, 192, and 256 bits. The evaluation criteria were divided into three major categories: *security, cost, and algorithm and implementation characteristics*.

NIST selected the five AES finalists on August 9, 1999. In October 2000, Rijndael (pronounced "Rain Doll") was announced as the winner and was approved as the official AES cipher. The two Belgian researchers who developed Rijndael are Dr. Joan Daemen (YO-ahn DAH-mun) of Proton World International and Dr. Vincent Rijmen (RYE-mun) of Katholieke Universiteit Leuven.

Cipher	Developers:
MARS	IBM
RC6™	RSA Laboratories
Rijndael	Joan Daemen, Vincent Rijmen
Serpent	Ross Anderson, Eli Biham, Lars Knudsen
Twofish	Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niels Ferguson

On December 26, 2001, NIST announced the approval of FIPS 197, which describes AES as an official government standard, by the U.S. Secretary of Commerce. FIPS 197 became effective on May 26, 2002.

The AES has supplanted the inadequate 56-bit DES, which is to be used only in legacy systems. AES has three key sizes: 128-bit, 192-bit, and 256-bit. Testing of the algorithm was performed by NIST and the Canadian Communications Security Establishment (CSE).

AES Algorithm

```
Cipher (byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
begin
    byte state[4,Nb]

    state = in

    AddRoundKey(state, w)

    for round = 1 step 1 to Nr-1
        SubBytes(state)
        ShiftRows(state)
        MixColumns(state)
        AddRoundKey(state, w+round*Nb)
    end for

    SubBytes(state)
    ShiftRows(state)
    AddRoundKey(state, w+Nr*Nb)

    out = state
end
```

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

AES Algorithm

The pseudo-code for the AES algorithm displays in the slide. Let's discuss it briefly:

- *in[]* is an array containing the block of plaintext. In general, *in[]* is $4 * Nb$ bytes in size—16 bytes in the current AES specification.
- *out[]* is an array containing the ciphertext and is also $4 * Nb$ bytes in size.
- *w[]* is the array containing the expanded key and is $Nb * (Nr + 1)$ words in size.

The code also shows one additional data structure, *state*, a two-dimensional array containing the current value of the transformed ciphertext. The transformations themselves are defined by four function calls: *AddRoundKey()*, *SubBytes()*, *ShiftRows()*, and *MixColumns()*. The specifics of these transformations are described in the next section.

The rest of the code is rather straightforward:

- The plaintext is moved into the *state[]* array.
- The first round key is applied.
- There are then $Nr - 1$ rounds that apply all four transformations.
- The final round applies to all but the *MixColumns()* transformation.
- The *state[]* array is moved into the ciphertext data structure.

AES Basic Functions

- AES algorithm employs four basic transformations:
 - **AddRoundKey:** XOR Round Key with State
 - **SubBytes:** Substitute bytes in State s to form State s' on a byte-for-byte basis using S-box
 - **ShiftRows:** Left circular shift of rows 1-3 in State s by 1, 2, and 3 bytes, respectively
 - **MixColumns:** Apply mathematical transformation to each column in State s to form State s'

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

AES Basic Functions

The AES algorithm employs four basic transformations:

- *AddRoundKey()*: Takes the appropriate round key and performs a bit-by-bit XOR with the current state.
- *SubBytes()*: Using a substitution box (S-box) defined in the specification, substitutes each 8-bit quantity in the State array to a different 8-bit value.
- *ShiftRows()*: Circularly shifts left the contents of state array rows 1, 2, and 3 by 1, 2, and 3 bytes, respectively. A left circular shift of one byte, for example, means that the bytes in columns 1, 2, and 3 move to positions 0, 1, and 2, respectively, and the value in byte position 0 moves to position 3.
- *MixColumns()*: Another byte value substitution, but in this case performed on a column (32-bit) basis; for example, rather than perform an S-box substitution on a per byte basis, this transformation applies a polynomial transformation on four bytes at a time.

AES (2)

USAGE

The AES algorithm has been developed to replace DES/3-DES, which is no longer officially considered to be secure.

DES/Triple DES is very widely used throughout the world today, and AES is expected to be just as popular...

VULNERABILITIES

No major vulnerabilities reported and viewed as a solid replacement for DES/3DES.

Only feasible attack is brute-forcing the keys.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

AES (2)

The AES development process has given us a splendid opportunity to see first-hand what it takes to develop a cryptographic algorithm. The process is inherently complex, and the only realistic way to reduce the risk of producing a weak algorithm is to open up the development activity to all interested parties and to the intense scrutiny by the global community of cryptologists.

Contrast the AES development process with that of DVD encryption. The DVD algorithm was developed in relative secrecy, and soon after DVDs began to use it, it was cracked. This embarrassing episode should be remembered by anyone developing a new cipher.

RSA

USAGE

Wide-spread support in major web clients, such as Microsoft Internet Explorer.

Has become even more popular since the patent expired 2000....

VULNERABILITIES

Cracking RSA generally means compromising poor implementations, or those using small key lengths.

So far, there have been no public reports claiming to have compromised the RSA algorithm itself....

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

RSA

The RSA algorithm has been widely implemented all over the world in all kinds of cryptography-enabled applications. It can be used to support both encryption and digital signature schemes. As a central part of the Secure Sockets Layer (SSL), it is also included in major web clients, such as Microsoft Internet Explorer and Netscape Communicator.

Although there have been a large number of claims to having cracked the RSA algorithm, they have all turned out to be false. Vulnerabilities have been found in certain RSA implementations, however. Poor implementations of the RSA algorithm can be compromised, but as in the case of other cryptographic algorithms, it does not mean the algorithm itself has been cracked.

The working mechanism of most public key (asymmetric) cryptographic algorithms are generally openly published and widely known. The security of the cryptosystem comes from the secrecy and size of the private key and not from the secrecy of the algorithm itself. As for other cryptographic algorithms, it is important to ensure that the key size is not so small that brute-force attacks become feasible due to the small size of the resulting key space.

Elliptic Curve Cryptosystem (1)

USAGE

Where high speed, low power consumption, low storage requirements, and high security at small key lengths is critical, e.g., in wireless communications, electronic cash, and ATMs

Growing in popularity...

VULNERABILITIES

Cracking ECC generally means compromising poor implementations, or those using small key lengths.

So far, there have been no public reports claiming to have cracked the ECC algorithm itself...

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

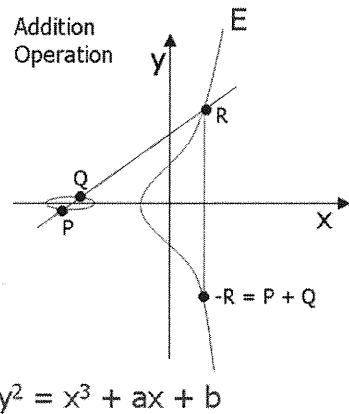
Elliptic Curve Cryptosystem (1)

Elliptic Curve Cryptosystems (ECCs) are capable of supporting both an encryption/decryption scheme as well as a digital signature scheme. In addition, the ECC has some very interesting characteristics: high security even at relatively small key lengths (that is, a higher strength per bit), high-speed implementations, low processing power requirements, and low storage requirements.

The previous properties make ECC a particularly attractive cryptographic option for use in resource-constrained computing environments such as mobile telephones, PDAs, information appliances, smart cards, and even the venerable ATMs.

Through the efforts of Certicom and others, ECC has enjoyed strong acceptance over the last 5 years and has been included in SSL/TLS standards and NIST standards. We expect to see increasing deployments of ECC-enabled applications in our e-commerce-enabled environments.

Elliptic Curve Cryptosystem (2)



Elliptic Curve Discrete Logarithm Problem (ECDLP)

- So far, the only publicly-known solutions for the ECDLP are fully-exponential. Per bit, this makes ECDLP stronger than RSA.
- The best-known algorithm for solving the ECDLP is the Pollard rho-method which involves performing a number of addition operations (see figure on left)
- The Elliptic Curve Diffie-Hellman Problem (ECDHP) has been mathematically proven to be equivalent to the ECDLP.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Elliptic Curve Cryptosystem (2)

In 1985, Neil Koblitz and Victor Miller independently proposed the Elliptic Curve Cryptosystem (ECC). Its security depends on the intractability of solving the discrete logarithm problem over points on an elliptic curve.

This slide depicts a general form of an elliptic curve $y^2 = x^3 + ax + b$, which is useful for understanding the implementation of elliptic curves over finite fields. The mathematics behind elliptic curves is actually very complex, and we do not go into the details in this course. However, we illustrate an interesting property of elliptic curves in this slide. It is that of adding two points, P and Q, on the elliptic curve. It is a property of elliptic curves that the addition of any two points on an elliptic curve always is another point (shown as -R in the figure) on the elliptic curve.

What exactly do we mean when we say that solving the "discrete logarithm problem over points on an elliptic curve" is an intractable problem? Briefly, for an elliptic curve C over a field F, let xP represent the point P added to itself x times (save one) and let $Q = xP$. Then, the elliptic curve discrete logarithm problem (ECDLP) is to determine x, given the known values of P and Q. Finding x from P and Q becomes intractable after the size of the underlying finite field reaches a threshold for which the total expected processing power necessary to solve for x becomes impractical.

So far, the only publicly known algorithms for solving the ECDLP are of exponential-time complexity. The best-known is the Pollard rho-method algorithm, which involves performing a number of addition operations (such as the $P + Q = -R$ operation).

RSA Versus DES (Asymmetric Versus Symmetric)

This 1000-fold difference in speed is likely to remain independent of technology advances:

- In software, DES is about 100 times faster than RSA
- Fastest implementation of RSA can encrypt kilobits/second
- Fastest implementation of DES can encrypt megabits/second
- It is often proposed that RSA can be used for secure exchange of DES keys

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

RSA Versus DES (Asymmetric Versus Symmetric)

Symmetric cryptography is generally much faster than asymmetric. Whereas the fastest hardware RSA implementation can encrypt on the order of kilobits per second, hardware DES is on the order of megabits per second. (DES was designed to run slowly in software, so in software, it is only about 100 times faster). The major drawback to symmetric cryptography is that, because both the sender and receiver use the same key, the key has to be exchanged via a secure mechanism before the two parties can communicate. Therefore, RSA is often used for the initial exchange of a symmetric session key. After the session key has been securely transmitted, Triple DES or some other symmetric cipher is used for the remainder of the session. So, we take advantage of the speed of a symmetric cipher without the worry of a shared key getting misplaced or stolen.

Comparing Key Length

Security (Bits)	Symmetric	DSA/DH	RSA	ECC
80		1024	1024	160
112	3DES	2048	2048	224
128	AES-128	3072	3072	256
192	AES-192	7680	7680	384
256	AES-256	15360	15360	512

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Comparing Key Length

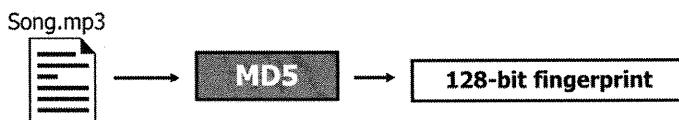
Bigger does not imply better. Typically within a given crypto algorithm, stronger ciphers result as the key size increases. When comparing key sizes across different algorithms, then the rule changes.

When comparing symmetric versus asymmetric cryptosystems not only are asymmetric algorithms more resource intensive (eat up your CPU time) than symmetric routines, but they are also less secure at the same key length. To compensate for this disparity and to increase the work factor for cracking a symmetric key pair, asymmetric key sizes are typically much longer. Although symmetric keys range from 40 bit to 256 bit, asymmetric keys are typically 1,000 bits or longer.

The table in the slide, taken from NIST, provides a comparison of key lengths for various algorithms. We can see that using Triple DES with two keys (112 bit) is equivalent in strength and work factor to RSA using 1024-bit keys. AES using 192-bit keys is as strong as RSA with key sizes approaching 8,000 bits. ECC also requires longer key sizes than symmetric crypto algorithms, but only needs to be double the size.

MD5

- MD5 takes variable-length input
- Output is 128-bit unique fingerprint



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

MD5

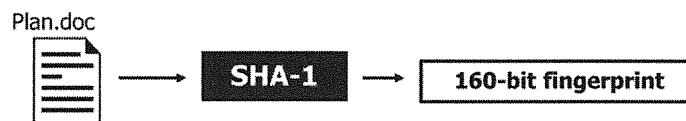
MD5 was created by Ron Rivest in 1991 and is more secure than its precursors, MD2 and MD4. MD5 is actually an extension of the MD4 algorithm and is more conservative in design. MD4 executed incredibly fast, but the cryptographic community felt MD4's collision protection was less than optimal. MD5's conservativeness increases its security but diminishes its execution speed. Most users don't mind sacrificing a little speed for the sake of enhanced security.

MD5 accepts arbitrary lengths of input and produces a fixed length output that is 128 bits, which is referred to as the key length. A hashing algorithm's output might be referred to as a hash, digest, or fingerprint. In the illustration, the song.mp3 file is input into MD5, and a 128-bit unique fingerprint of the file is created. MD5 does not modify the original file in any manner whatsoever.

As an extra precaution, security-conscious users may choose to cryptographically sign the fingerprint to guard the fingerprint against inadvertent modification. Digitally signing a fingerprint cryptographically safeguards the integrity of the fingerprint.

SHA-1/SHA-2

- SHA-1/SHA-2 takes variable-length input
- Output is 160-bit unique fingerprint



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

SHA-1/SHA-2

NIST developed the Secure Hash Algorithm (SHA) in 1993. SHA-1 was released in 1994 to correct an unpublished flaw in the original release of SHA. SHA-1 is a Federal Information Processing Standard, specifically FIPS 180-1, and is permissible for U.S. Government systems. SHA and SHA-1 are pronounced "shaw" and "shaw-one" respectively.

SHA-1 produces a 160-bit fingerprint compared to MD5's 128-bit fingerprint. Although SHA-1 is slower than MD5, SHA-1's larger fingerprint makes it more secure against collision attacks.

In the illustration, the plan.doc file is input into SHA-1; subsequently, SHA-1 produces a 160-bit fingerprint. As with MD5, security-conscious users would cryptographically sign the fingerprint to safeguard the fingerprint against inadvertent modification.

With both MD5 and SHA-1, security-conscious users must explicitly conduct a second, independent step to protect the fingerprint against modification. Hashing algorithms do not provide any intrinsic means of protecting the fingerprint once generated.

Cryptanalysis

The student will be able to identify common attacks used to subvert cryptographic defenses.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Cryptanalysis

This section intentionally left blank.

Cryptanalysis (1)

- **Analytic:**
 - Uses algorithms and mathematics to deduce key or reduce key space to be searched
- **Statistical:**
 - Uses statistical characteristics of language or weaknesses in keys
- **Differential:**
 - Analyzes resultant differences as related plaintexts are encrypted using a cryptographic key

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Cryptanalysis (1)

We now look at some general types of cryptanalysis:

- *Analytic*: Uses algorithms and mathematics to deduce key or reduce key space to be searched.
- *Statistical*: Uses statistical characteristics of language or weaknesses in keys.
- *Differential*: Analyzes resultant differences as related plaintexts are encrypted using a cryptographic key.

Cryptanalysis (2)

- Linear:
 - Linear analysis of pairs of plaintext and ciphertext
- Differential linear:
 - Applies differential analysis with linear analysis

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Cryptanalysis (2)

- *Linear*: Linear analysis of pairs of plaintext and ciphertext.
- *Differential linear*: Applies differential analysis with linear analysis.

Birthday Attack

- When 23 people are put together, the odds are greater than 1/2 that 2 or more people will share a birthday
- Hash collisions is related to that probability

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Birthday Attack

Cryptanalysts can sometimes use a phenomenon known as the *birthday paradox* to attack hash signatures. People in large groups often find that at least two of them share the same birthday. They're usually astonished at the coincidence, thinking that the odds must be very slim that two people could be born on the same day of the year. It is true that it would be rather unusual to find a person with your exact birthday unless the group was very large. The odds of finding someone born on a particular day are 1 in 365 (assuming all days of the year are equally likely birthdays and nobody was born on February 29).

However, just specifying that any two people have the same birthday, without specifying whose birthday it is, improves the odds considerably. For a group as small as 23 people, the odds are greater than 50% that two or more of them will share a birthday. If each of the 23 people compares birthdays with another, you'd have 253 comparisons. The odds, then, that *none* of the 23 have the same birthday are $(364/365)^{253} = 0.4995$. Thus, the odds that two of them share a birthday are $1 - 0.4995 = 0.5005$.

Just as pairs of people in a group might have the same birthday, pairs of messages might have the same hash signature. Of course, there are many more possibilities for hash signatures than birthdays, but the same logic applies. If an attacker can find any two messages that generate the same hash value, that is, a *collision*, she could substitute one message for the other at will. For example, maybe she has a list of password hashes but not the cleartext. If she can hash enough of her own generated cleartext to cause a collision, she has a password that works just as well as the real thing.

The entire attack is a statistical probability problem.

Steganography Overview

The student will understand what steganography is and how it differs from cryptography.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Steganography Overview

This section intentionally left blank.

Steganography (Stego)

- Data hiding (Steganography means "covered writing")
- Involves concealing the fact that you are sending "sensitive" information
- Dates to Ancient Greece, modern awareness relatively new
- Can hide in a variety of formats:
 - Images (bmp, gif, jpg)
 - Word documents
 - Text documents
 - Machine-generated images (fractals)



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Steganography (Stego)

Steganography (stego) is a means of hiding data in a carrier medium. Steganography means, "covered writing." In concept, it dates back to ancient Greece. However, as a means of hiding data electronically, it is a new concept.

The modern form of stego can take many forms, although all involve hiding data in something else called a *carrier file*. This can be hiding a document in an image, hiding a short message in a document, and even hiding an image in a sound file! The applications are only limited by the tool being used, the carrier file, and the imagination of the sender.

Stego can be used for a variety of reasons but most often it is used to conceal the fact that sensitive information is being sent or stored. It can also be used to disguise encrypted data. This helps prevent attacks on encrypted data, or in scenarios where encrypted data is inappropriate for transmissions—for example, in countries where encryption is against the law.

Crypto Versus Stego

- Cryptography (crypto) provides confidentiality but not secrecy
- It is fairly easy to detect that someone is sending an encrypted message; it is just very hard for someone to read it
- With stego, you might not even know someone is sending a message —the true intent is hidden

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Crypto Versus Stego

Cryptography (crypto) is a tool to protect confidentiality and integrity and provide non-repudiation for the senders of data. However, despite all of these benefits, crypto does not guarantee the secrecy of your data. Scrambling the data into unintelligible ciphertext can prevent others from reading the file, but it does not keep them from realizing that the data is there. It is easy to detect an encrypted message; it is difficult to read one.

One unwanted side effect of using encryption is that it can mark a user's most important and confidential files. It is similar to keeping valuable items in a bank vault, or an armored car. Encryption keeps the content very safe, but when attackers are in hot pursuit, they know what to target for the valuables.

An encrypted conversation can also raise suspicions. If two parties suspected of a crime had suddenly started trading extensive encrypted messages the week before the crime occurred, even though we might not know what they were saying, it would definitely raise some flags and concerns.

When handling extremely confidential data, it would be ideal to obfuscate the information and keep it as undetectable as possible. Secrecy keeps an attacker from even trying to subvert the encryption on these files. They see image or sound files, yet they have no idea that they are also carriers of encrypted data.

Steganography Doesn't Guarantee Safety

One important thing to keep in mind when using stego is, that even though the secrecy provided by stego is great, the data's protection still relies on the encryption algorithm that is being used. Some stego programs use weak or untested encryption algorithms, or in some cases no encryption at all! Some stego tools have a choice of encryption methods of varying effectiveness that require you to choose between. Users are often duped into a false sense of security while using a stego tool. They think that if the data is hidden, it is safe. However, if stego is detected, the safety of your hidden message is only as good as the encryption that is used to protect it. If the confidentiality of your data is important to you, always verify the stego tool that you are using has a proven encryption algorithm. If it doesn't, or you are unsure, encrypt the data with a tool using a proven algorithm (such as PGP) before running it through the steganographic process.

Detecting Cryptography (1)

- It is very easy for both humans and computers to detect that a message is encrypted. For example, "test" becomes:

```
eJrMIedoDcgYmK7/XwY6Q+7RAeuPDSe0FziMLDU1GyUhc0WPcat  
AaIpw+Urc0MUXI257b1q11gFZN4S0rXwAKg2Tzqn9ois7+1pJHO  
dxI2fH9LCQmxRBpZ79oFh+wFwcuPV3wW4Mgoh1HL2JQ7Sarr  
JuZixgRoV+iW/HtoWx2Mvop+4CACHtTxbv8SjchhNFLaQNVQA  
1o00UgR+m7bJh42bWfR5cdGBYkVTzglbu5QXzFodk3PmtG+ghq  
NCz2CZ5Vzv3H581bSeydcM5zjK7DUd4OZEDSa9kF+9xKdyDMC  
fvFW5DyhIjkOBUVo8jvQMn/3n08vGcx/5CcDVV6MF4xh5hPbV6  
NfP2OaOyNVXcHwn9n6/swH4OnrBciX8MCgFJCyXrwnlYI1GK7R  
BO67zw0imUkBABfAqc+Jwnbv2HJAAU0NDC+Vd+d9I4UZN6QJd  
7RN821ID10ScXeLDNiCq8hxXHJM8qaP5gQp5iC2ExoPffPPI8KRsb  
OKcK5XPP57T
```

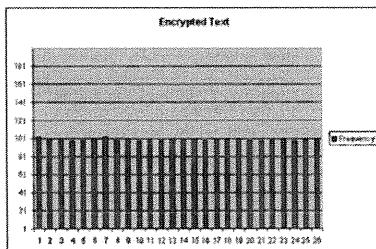
- A human can infer that because this is unreadable, it may be encrypted.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Detecting Cryptography (1)

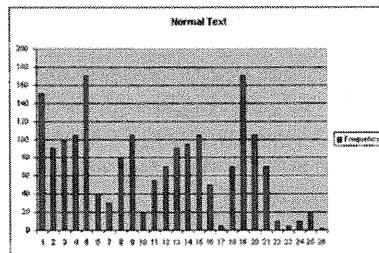
Both humans and computers can easily detect encryption because encryption increases file size and mathematically normalizes the occurrence of data. When viewing a document or e-mail comprised of garbled characters, one might infer that it is actually encrypted.

Histograms



The histogram for "normal" text is very non-uniform and easy for an automated program to distinguish between encrypted and unencrypted information.

The histogram for encrypted text is very flat and easy for an automated program to detect.



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Histograms

Histograms are graphical representations of the number of occurrences of data in a given distribution of such data. For example, a histogram of a text document would show the number of occurrences of each character that appears in the document. A normal text document would generate a histogram that shows that the frequency of characters varies greatly. In a histogram for an encrypted document, the frequency of characters is normalized. The same factor that helps prevent encryption from being interpreted makes it easier to detect.

How Steganography Works

- Stego requires a host (to carry the data) and the hidden message
- Host (usually a file) can be generated on the fly or use existing data
- Message can be hidden in certain parts of existing file, or can cause a new file to be generated

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

How Steganography Works

The principle behind steganography is simple—hiding data within data. This can be done in many different ways. The only limiter is the steganographer's creativity. Despite the seemingly endless possibilities for stego, there are some commonalities that can be found in its operation. There are several basic components that are common to all stego and several general types of operations that all stego can be categorized into. In the following sections, we explore these tenants of basic steganography.

For the purposes of this section, we focus on steganography as it relates to files. Other non-file carriers are possible with stego including data streams and correlative messages.

The Components of Stego

There are two general components of standard steganography. The first is the carrier or host file. This is the medium used to hold the hidden data. The carrier can be almost any type of file imaginable. Some popular examples of such hosts are:

- Images: bmp, gif, and jpeg
- Word documents
- Sound files
- Movies: mpeg
- Text documents
- Machine-generated images: fractals
- HTML files

General Types of Stego

- There are many ways to hide information; lesson in creativity
- General methods:
 - Injection
 - Substitution
 - File generation

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

General Types of Stego

Information can be hidden in many ways. In ancient times, information was placed on wooden tablets that were then covered with wax to hide the message. Messages were also tattooed onto messengers' bare heads. (Hair growth covered the message and their head needed to be shaved so the recipient could read it). In more recent history, messages were written with invisible inks that appeared only after they were heated. Messages were written with these inks in the margins or between lines in false documents to hide the fact that a hidden message existed.

In the information age, there are many new creative ways to hide information in an electronic carrier. Most of the techniques can be summed up in one of three general stego types:

- Injection
- Substitution
- File generation

Injection

- Most file types have ways of including information that will be "ignored"
- For example, hidden form elements or comments in HTML; GIF comments
- Word documents also have hidden information—holes in the data:
 - Create a large document, save, and then remove data. File size is still very large
- Increases the size of the file

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Injection

With most file types, there are ways to include information within them that will be ignored when the file is processed. This is the basis for injection stego. We place the information into "holes," or unused areas of the file. For example, with HTML, informational tags that tell how it should be processed must precede all characters. Web browsers ignore data that is formatted with certain HTML tags. However, if you examine the same HTML file with a text or HTML editor, the added characters will be fully visible. Another example is the comments that can be inserted in files, such as those that can be placed in a GIF image or MP3 sound file. These comments do not appear when you view or play the file, though they still physically exist in the body of the file if you know what to look for.

Even Microsoft Word documents contain areas (or holes) where information can be hidden. This can be demonstrated by creating a large document, saving it, and then by "cutting" a large portion of the document out. Even after the data is removed, the file size is still very large. The slack that is left in the document can also have data inserted into it.

The greatest problem with the injection type of stego is that as data is added, the file size of the carrier increases. This makes detection easy if the original file can be found, or if the size is increased outside of the norm for its type. For instance, if an MP3 file was injected into a document file, the increased size of the document will most likely be noticed.

One example of a tool that utilizes the injection type of steganography is Snow (<http://www.darkside.com.au/snow/>). Snow is a command-line program that allows the encryption and injection of a hidden message into an ASCII text file as "white space," which consists of extra spaces and tabs. It uses the ICE encryption algorithm that was authored by the same person, Matthew Kwan, as Snow and is generally untested and should not be used for high security purposes. Ice supports up to a 1024-byte encryption key. To encrypt a message, the command is as follows:

```
snow -C -m "This is the message" -p "secret password"  
<input text file name> <output stego text file name>
```

-C compresses the information, -m specifies the message to be encoded (in quotes), -p specifies the password to encrypt the document (in quotes), followed by the name of the text file to encode the information into, and finally the output text file that will hold the stego payload.

To extricate the encrypted payload from the text file, simply use the following:

```
snow -C -p "secret password" <stego text file name>
```

-C uncompresses the information, -p specifies the password that was used to encrypt the file, and finally the text file name is listed, which holds the stego payload. The hidden message is output directly to the screen. It can be redirected to a file by adding "> filename.txt" to the end of the command-line.

The data is hidden by adding a series of spaces and tabs to the end of each line, which in turn represents the bits of hidden information. This method can hold approximately three bits of information per eight columns in the document.

When comparing the original document and the stego carrier with most text editors, it is impossible to tell the two apart visually. However, viewing the same document through a file comparison utility or hex editor shows how very different the files actually are.

Injection stego is a viable method to hide small amounts of information in a carrier file.

However, because the information is added to the existing contents of the file, an increase in file size can be detected making it typically unsuitable for concealment of larger amounts of data. When large amounts of data need to be concealed, a method of stego where file size is not affected is advisable.

Substitution

- Data in a file can be replaced or substituted with hidden text
- Depending on the type of file and/or the amount of data, it could result in degradation of the file
- It usually replaces insignificant data in the host file

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Substitution

Substitution is the most popular stego method used to hide data in a host file. The concept is that elements are replaced on a bit-by-bit basis with information that is being hidden in the host document. Because the information is substituted in place of existing information, the file size of the carrier remains the same. However, noticeable file degradation can occur depending on the amount of information placed in the document. The goal with this technique is that only insignificant data should be overwritten to prevent degradation. It is important to have a suitably large carrier file when great amounts of information are being concealed. Typically, insignificant data is replaced with the information to be hidden. This insignificant data can take many forms, but one of the most common forms is the least significant bits (LSB) in the color table of a graphic.

Generate a New File

- The hidden data can also be used to generate a new file
- No host file is needed
- For example, the input text can be used to generate fractals or "human-like" text

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Generate a New File

Another method of stego that is growing in popularity is the actual generation of a new file from the data to be hidden. This is the only form of stego where a carrier isn't needed beforehand. A carrier file is needed, but it is generated on-the-fly by the stego program. The carrier file is actually created from the source information to be concealed. This can be used to generate such output as readable text or fractals. With each unique input file, a completely new and unique output file is generated.

Summary

- Concepts in Cryptography
- Symmetric (Private) Key Systems:
 - Triple DES
 - AES
- Asymmetric (Public) Key Systems:
 - RSA
 - ECC
- Hashing
- Stego

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Summary

Cryptography is essential for e-commerce, military communications, and the privacy of individuals on the Internet and other networks. A cryptographic algorithm must provide confidentiality, integrity of data, authentication, and non-repudiation. To achieve these goals, mathematical problems with a suitable computational complexity are used to create a cipher that takes too long to break using brute-force methods to be practical. Such problems are considered *intractable*, and though they cannot usually be proven to be secure, years of research and scrutiny leads us to believe they are.

Cryptography has been around for at least five millennia. Over the years, substitution and transposition ciphers have been used with increasing complexity. The German Enigma machine employed a very sophisticated substitution cipher. The Vernam one-time pad is perhaps the most difficult to crack cipher, but the fact that the key must be as long as the message makes it very difficult to use.

The Data Encryption Standard (DES), introduced in 1975, is too easy to brute-force for serious use today, but Triple DES effectively lengthens that key size. Triple DES employs the already available DES implementations and takes advantage of DES's decades of public scrutiny.

A permanent replacement for DES was needed to last for decades to come, and that was chosen as the Advanced Encryption Standard (AES) in October 2000. Rijndael, chosen from one of five finalists, is a symmetric cipher with three possible key sizes: 128-bit, 192-bit, and 256-bit.

Symmetric ciphers aren't right for every application. When the two parties, such as a merchant and a customer, have never met, secure key exchange is not easy. Public key cryptography solves this problem by allowing for separate keys for encryption and decryption. The famous RSA algorithm is one such cipher, and it is used by the Secure Sockets Layer (SSL) to provide secure communications on the Internet. Because symmetric cryptography is much faster, RSA is often used to exchange a session key for Triple DES, which then encrypts the transaction.

Proposed in 1985, elliptic curve cryptosystems (ECCs) offer the possibility of strong cryptography with low overhead. Thus, ECC lends itself to embedded applications in which memory and processing speed are at a premium.

When attacking a cipher, a cryptanalyst takes advantage of what information he has. Plaintext, ciphertext, and relationships between keys can all be useful. Being able to choose the text that gets encrypted or decrypted can be even more useful, because the cryptanalyst can deduce information based on his own input. For attacking hash algorithms, the birthday paradox makes it surprisingly likely to find collisions, two messages that hash to the same value.

Perhaps the most important lesson in this section is that ciphers should be developed in the open, taking advantage of the collective brainpower of cryptologists throughout the world. This kind of scrutiny reduces the likelihood that a weak algorithm is used and encourages cipher designers to place all of a cryptosystem's security in the key rather than the algorithm itself.

Module 20: Applying Cryptography

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Module 20: Applying Cryptography

This section intentionally left blank.

Applying Cryptography

SANS Security Essentials IV: Secure Communications

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Introduction: Applying Cryptography

As was discussed in the previous two modules, cryptography has many applications in information security. You can consider it the Swiss Army knife of information security because it has so many useful purposes. The top purposes are confidentiality, integrity, authentication, and non-repudiation. By encrypting information with a key that only the rightful users of the information possess, it can be used to protect the information from prying eyes (confidentiality). It can also be used to detect information tampering (integrity) by cryptographically hashing the information and then encrypting the hash. If the information is tampered with, the hash will not match proving that the information has been modified. Cryptography can be used to prove identity (authentication). This can be done by requesting that the user encrypts a test message. The encrypted test message is then decrypted using the user's stored key. If the message decrypts successfully, the user has proven his identity, or at least that he possess the right key! Non-repudiation allows someone to prove in a court of law that someone agreed to a contractual relationship.

Objectives

- Data in transit:
 - Virtual Private Networks (VPNs)
- Data at rest:
 - Pretty Good Privacy (PGP)
 - Full disk encryption
- Key management:
 - Public Key Infrastructure (PKI)

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Objectives

The abilities that cryptography offers us are great, but how are they being used in real-world networks? In this section, we discuss some of the practical applications of cryptography, including how cryptography can be used to protect communications across a network, protect information resting in storage, provide authentication services, and ensure the integrity of information.

Applying Encryption to Network Communications

Our discussion begins with one of the most common networking uses for cryptography: protecting information as it flows across a network. Information is very exposed when it leaves your PC to travel across a network to another PC or server. At any point between the source and destination of a message, a man-in-the-middle can capture or modify the information contained within the message. What does this mean in the real world? Well, without encryption, an attacker might be able to capture your credit card details as you provide them to an online retailer.

Potentially more damaging, many network protocols do not encrypt their session information. These protocols transmit your username and password information "in the clear." An attacker who can listen in on your network conversations when you use one of these unencrypted protocols will be able to impersonate you, gaining access to all of the network resources you have access to through that protocol. As a last example, consider the damage an attacker could cause by simply modifying the contents of the right network conversation. Changing an account number used during a banking transfer or the ship-to address during an online purchase could defraud you of potentially large amounts of money. Cryptography provides a powerful method to protect against these information security risks.

Where to Encrypt

A basic question when protecting network communications is where the protection should be performed. Should each application be responsible for protecting its own network communications, should cryptography be implemented as a service that applications can optionally use, or should it be included at the network level where all communication from and to particular locations can be protected? In practice, all of these methods are currently in use.

Application Specific

Replacing unencrypted protocols like telnet with secure alternatives can be an easy way to improve security, assuming that a secure replacement exists. Examples include replacing post office protocol (POP) with authenticated post office protocol (APOP), Network File System (NFS) with the Andrew File System (AFS), and (more commonly) telnet and ftp with Secure Shell. Keep in mind, however, that each application can implement different security enhancements and not all replacements protect all conversations. For example, APOP only improves on POP by protecting the authentication messages, not the e-mail messages themselves. APOP prevents casual eavesdropping of usernames and passwords but does not prevent an attacker from listening in or modifying e-mail conversations.

Protocol	Purpose
Telnet	Remote login
R* commands (rlogin, rsh, rexec, etc)	Remote login, and command execution
FTP	File transfer
POP, IMAP	Retrieval of e-mail from mail server
SMTP	Transfer of mail between mail servers

Secure Shell, on the other hand, provides several security enhancements to the protocols it replaces. Critical is its capability to support strong certificate-based authentication, its capability to encrypt all session traffic providing confidentiality, and its capability to authenticate both sides of a connection, server and client. A high-quality, free version is available from the OpenSSH organization making its use a no-brainer whenever telnet or ftp like services are required.

Transport Layer

Another option for protecting information transfer is to provide a secure communications service that many applications can use. The advantage of this method is that each application does not have to re-implement the same security services. This is the approach taken by Transport Layer Security (TLS).

The primary goal of the TLS Protocol is to provide privacy and data integrity between two communicating applications. It is comprised of two protocols: the record protocol, which is used to securely transfer application data, and the handshake protocol, which is used to negotiate the details of a secure session. The combination of these two protocols provide, in an application protocol independent way, confidentiality of the communications using symmetric key encryption, integrity of the communications using a cryptographic message authentication code, and mutual certificate-based authentication of client and server.

Many application protocols have been redesigned to utilize TLS security services. There are RFCs that define SSL's use for protecting the transfer of mail from server to server, for the secure retrieval of mail from a mail server (RFC 2595), and for authentication of PPP sessions (RFC 5216). Its most common use is to protect and authenticate web sessions.

You most likely have used TLS if you have ever purchased anything from a web site. Most e-commerce web sites use TLS to protect communication whenever sensitive information is being requested from you, such as your credit card numbers or your address. You can tell when the web site you are visiting has activated TLS because a symbol of a locked padlock appears at the bottom of your browser's window whenever an TLS session has been successfully set up.

Even though TLS is application independent, applications must be modified to make use of its services. Just as with application specific cryptography, if the applications you need to use do not support TLS, you cannot make use of its protection. This is where the last type of network cryptographic protection we will discuss comes in.

Network Layer

Network layer encryption protects network conversations whether the application using the network supports cryptography or not. Network layer encryption sits in-between the transmitter and receiver. It accepts in clear-text information, and then encrypts it prior to sending it out. At the receiving end, the information is decrypted and forwarded on to its final destination. This type of network encryption is called a “virtual private network (VPN).”

Virtual Private Networks (VPNs)

The student will have a high-level understanding of what VPNs are and how they operate.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Virtual Private Networks (VPNs)

This section intentionally left blank.

Confidentiality in Transit

- Private network:
 - **Pro:** Dedicated lines and equipment are not shared by others
 - **Con:** Dedicated lines are expensive, grow more so with distance, and are underutilized except at peak

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Confidentiality in Transit

Prior to the popularization of VPNs, companies wanting to protect network conversations between different locations purchased dedicated- leased lines, frame-relay circuits, ATM connections, or other types of private circuits that provided connectivity between the sites from a telecommunications company. They could be reasonably confident that their information could not be intercepted because these circuits allow only the two sides of the connection to exchange information. No third parties should be able to communicate over the private connection. This assumes that you trust the telecommunications provider, which might or might not be a good bet depending upon where in the world you are. Although secure, these circuits also tend to be slow and expensive and become more expensive as they get faster, or the distance increases between the sites that need to communicate. There is also a large lead-time between the decision to set up one of these connections and getting it running. It can take months for a telecommunications company to fulfill a new circuit order.

Virtual Private Network (VPN)

- Data is encrypted at one end of the VPN from "cleartext" into "ciphertext"
- Ciphertext is transmitted over the Internet
- Data is decrypted at the other end of the VPN from "ciphertext" back into the original "cleartext"

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Virtual Private Network (VPN)

VPNs are a perfect alternative to costly, inflexible private circuits. They give companies the option of setting up virtual circuits across public networks, such as the Internet. Encryption provides the confidentiality needed as the private information flows across the public network. This capability allows VPNs to establish secure communication between different remote offices and can also be used to establish remote access to internal network resources by employees from their homes or while they are on travel.

VPN Advantage—Flexibility

- VPNs are flexible
- A VPN "tunnel" over the Internet can be set up rapidly. A frame circuit can take weeks
- All you need is an Internet connection
- A good VPN also supports Quality of Service (QoS)

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

VPN Advantage—Flexibility

One of the biggest benefits of VPN technology is its flexibility. If you need a secure channel between two hosts only for a day, or even an hour, then a VPN might fit the bill. After you have all of the components to establish a VPN, setting one up requires only configuration. This makes the technology far more flexible than private circuits, which must be ordered far in advance of their use and might also require additional hardware. This flexibility lends itself to creating new business solutions. For example, it's not cost-effective to wire a T1 for every employee who works from home. It is practical, however, to load software on an employee's laptop and let her connect to the home office via a VPN over the Internet. This assumes that the home users already have connections to the Internet.

VPN Advantage—Cost

- Connect easy and cheap to the Internet
- Create two Internet connection points
- Encrypt traffic over the Internet
- Pro: A lot more bandwidth cheaper
- Con: Don't get dedicated bandwidth across the Internet
- Only 1-2 second delay compared to private network:
 - Can be critical for some operations

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

VPN Advantage—Cost

Another significant advantage of using VPN technology is the cost savings. It is simple and affordable to get Internet access these days, and in many cases that is all the connectivity that a VPN client will need. A VPN tunnel is then established between two Internet connection points, and the traffic between them is encrypted.

Generally speaking, this provides organizations with more bandwidth for less money (depending on the speed of the remote Internet connection). One downside to this, however, is that there is no way to offer dedicated bandwidth over the Internet. There are a number of factors that might influence a VPN client connection, most of which are out of the organization's control. Another possible negative aspect of VPN connections is a 1-2 second delay in communications compared to private networks. Although this is acceptable to most organizations, some critical operations might not be able to afford the delay.

VPN Breakdown

- VPNs not ideal for financial, medical, and other real-time operations
- VPNs are ideal for file transfers, e-mail, and so on
- If time is not critical, VPNs can save a lot of money
- If time is critical, dedicated lines are recommended

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

VPN Breakdown

There are also some disadvantages to VPNs, the main one being the lack of performance guarantees. Most private circuits, such as leased-lines or ATM, have the capability to guarantee bandwidth and latency. Similar guarantees have been difficult to achieve with VPNs. TCP/IP, the networking protocol for the Internet, was not designed to provide quality of service (QoS) and improvements have been slow in coming. Providing QoS for VPNs is even more difficult because many QoS solutions require the service provider to look into the messages they are passing on to decide whether the message has higher priority than other messages. If the service provider cannot examine the information in a message (because of encryption), it makes it even more difficult to decide which network traffic should get priority.

There are solutions to these problems. Multiprotocol Label Switching (MPLS), an alternative over traditional layer three routing, is used to address these problems. It allows forwarding of messages across the Internet without requiring examination of the message contents. MPLS-based VPNs can be purchased from a wide variety of Internet service providers, though they are more expensive than standard IP services.

Types of Remote Access

- Client-to-site VPN (transport):
 - Example: Laptop dial-up connection to remote access server at HQ
- Site-to-site (tunnel):
 - Example: L.A. office connection to DC office location

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Types of Remote Access

There are two primary categories of VPNs to consider, *client-to-site*, and *site-to-site*. Client-to-site VPNs provide remote access from a remote client, such as a traveling sales rep or telecommuting employee to the corporate network. Such VPNs are normally established between the client's computer and a gateway device located at the border of the corporate network. The client's computer runs VPN software that allows it to establish the connection to the VPN gateway.

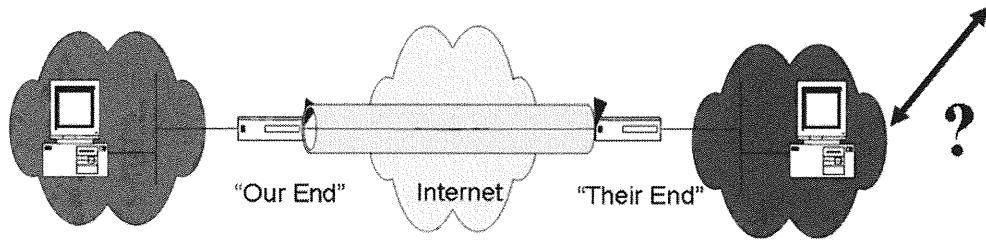
Site-to-site VPNs provide connectivity to networks, such as headquarters and a remote office. In these connections, gateway devices are located in *front* of both networks. Information needing to flow between the sites is directed to the local gateway, which then encrypts the contents of the message and forwards it to the other site's gateway. The remote site's gateway decrypts the message then sends it onto its final destination.

There is a third, less common type of VPN, the client-to-client VPN. These VPNs establish a protected link between two specific computers. As such, they could be considered the most secure of the VPN types, because in the client-to-site and site-to-site VPNs, part of the path between the transmitter of a message and the receiver of the message is unencrypted. For instance, in client-to-site VPN, the communication from the client's computer to the VPN gateway is protected, but the message travels unencrypted (and unprotected) from the VPN gateway to the internal corporate server the client is trying to communicate with. If an attacker inserts herself somewhere between the VPN gateway and this server, she would be able to eavesdrop or modify the contents of the message.

If client-to-client VPNs are more secure, why are they not used more often? The majority of the reason is the configuration required. Each pair of hosts wanting to communicate must be specifically configured to allow the communication. The most important part of this configuration is key installation. Each host must have a separate unique key that it can use to encrypt information to a particular destination host. Because of this, client-to-client VPNs between every two hosts would quickly become unmanageable as the number of hosts increases, if manual configuration is used. Public Key Infrastructure, which is discussed later in this module, is one way to address this key distribution problem.

Security Implications

- Be careful where encrypted tunnels are set up to avoid bypassing security devices
- Both ends must be trusted



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Security Implications

Many sites assume that because they have established a VPN, they are secure. This is a bad assumption because VPNs bring their own special security concerns into your network. One frequent error made with VPNs is to overly trust the other side of a VPN connection.

With site-to-site VPNs, it is common to see the VPN connection allowed into the network without applying any security restrictions to it. This might be appropriate if the other side of the VPN belongs to the same organization and is controlled by the same security policies and procedures. If the other side of the connection is another organization, such as a business partner, though, access through the VPN should be restricted. Most VPN gateways include firewall abilities allowing them to limit network traffic across the VPN. It is a best practice to restrict this traffic to the minimum necessary to fulfill the business needs of the connection.

Another potential security problem VPNs introduce is caused by the encryption VPNs use to protect the messages they exchange. As mentioned previously, this encryption prevents an attacker from eavesdropping, but it also prevents intrusion detection systems and anti-virus tools from examining the packets for malicious or inappropriate content. This reduces or eliminates the effectiveness of these security tools.

Last, client-to-site VPNs suffer from the trusted client problem. Many organizations have strict rules on the type of software allowed on corporate computers. Part of the reason for these controls is that unauthorized software might contain security vulnerabilities.

When allowing employees to use a VPN to access the corporate network, the organization might not be in the same position to dictate a tight configuration. In fact, most home computers are insecurely configured. If an attacker discovers the home computer and takes it over, he might be able to use his access to the computer to leverage access to the corporate network over the employee's VPN connection. For this reason, it is a good idea to recommend, or better yet, enforce the use of a personal firewall product and anti-virus software prior to allowing remote users to access client-to-site VPNs.

Now that we've discussed how encryption can be used to protect communications over a network, it's time to introduce some concrete examples of technology that implements these ideas. The first is IPSec, the current industry standard for setting up VPNs.

IPSec Overview

- Issued by IETF as an open standard (RFC 2401) thus promoting multi-vendor interoperability
- Can enable encrypted communication between users and devices
- Implemented transparently into network infrastructure
- Scales from small to very large networks
- Commonly implemented: Most VPN devices and clients are IPSec-compliant

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

IPSec Overview

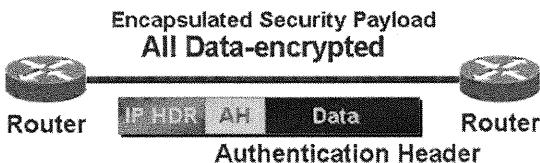
IP Security (IPSec) is an IETF standard for establishing virtual private networks. It is slowly replacing proprietary VPN protocols and becoming the industry standard. Many products on the market now support IPSec natively, such as Checkpoint Firewall-1, Cisco routers, and Windows XP/Vista.

Like the application-level and transport-level techniques we have discussed, IPSec provides data integrity, confidentiality, and authentication. IPSec also offers sophisticated replay attack prevention.

Attackers use replay attacks by copying a message as it goes across the network, then re-transmitting the copy to the destination. Even if the attacker cannot read the encrypted message, he can cause undesired results. For example, if the message was a request to transfer \$1,000, the replay might be able to cause an additional transfer making the total transferred \$2,000. IPSec includes specific mechanisms to detect and prevent replay. Replay attacks are often used to capture encrypted authentication sessions and replay them later to log on to a given system.

Types of IPSec Headers

- Authentication Header (AH):
 - Data integrity: No modification of data in transit
 - No confidentiality
 - Origin authentication: Identifies where data originated
- Encapsulated Security Payload (ESP):
 - Data integrity: No modification of data in transit
 - Confidentiality: All data encrypted
 - Origin authentication: Identifies where data originated



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Types of IPSec Headers

IPSec is actually a collection of protocols used singly or together to implement its various network security services. Primarily, IPSec is composed of two main modes: the Authentication Header (AH) protocol and the Encapsulated Security Payload (ESP) protocol. To understand how IPSec works, let's examine the abilities offered by each of these protocols.

Authentication Header (AH)

AH provides message integrity, anti-replay, and source authentication. It works by adding authentication information into each IP packet. To see how this works, we need to understand some of the information that goes into an IP packet.

IP packets are composed of many pieces of information, with each being important. One of the most important, from a security standpoint, is the source IP field. The source IP field is used to tell the recipient who sent the message. In a normal network conversation, the computer that is sending a message uses its own IP address as the source address. This is important to the security of the system because many firewall systems use source IP addresses to determine whether a message should be allowed into a network or not. If an attacker can choose to lie about his IP address, he could potentially use an address that the firewall does allow in, fooling the firewall into accepting a message that it should have denied. Without AH, there is nothing to prevent an attacker from lying about the source or any other field inside the packet.

To prevent this, AH adds a keyed hash of the message to the packet. This hash is referred to as the Integrity Check Value (ICV). In the ICV computation, AH includes every field that does not change during its trip from source to destination. This includes the source address, destination address, length, and the data. This information is inserted into the packet after the regular IP header, but before the data.

To verify that the packet has not been tampered with, the recipient recomputes the ICV. If any of the hashed fields, including the source address, have been changed, even by a bit, the hash will be different and the integrity check will fail. This provides both integrity checking and authentication. The integrity is guaranteed because the hash must match the message.

However, what about the authentication? Remember that this is a keyed hash. The key used is negotiated between the sender and recipient prior to the start of communications. You can compute only the hash if you know the right key. Thus, if a recipient can re-compute the hash using the key previously agreed upon with the sender, then the message has been authenticated as originating from that sender.

The algorithm used to create the ICV is configurable. The architects of the IPSec protocol endeavored to minimize any dependency between IPSec and the cryptographic algorithms that it relies upon. This is to prevent the standard from becoming out-of-date if a new cryptographic algorithm needs to be supported. Only two algorithms are required by the IETF for a particular AH implementation to be considered compliant to the protocol. These are MD5 and SHA-1. Both algorithms are used by AH for the same purpose, the creation of a hashed message authentication code (HMAC).

As mentioned previously, some fields have to be left out of the ICV computation because they change during transmission. An example of this is the time-to-live (TTL) field. The TTL field is used to limit how many different routers (or hops) a packet can pass through before it reaches its destination. Every time a packet arrives at a router, its TTL field is decremented. When it reaches zero, the packet is dropped and an error message is sent back to the source of the packet. You can see why this could never be included inside the hash computation. This field is guaranteed to be different by the time it arrives at the recipient. The recipient's hash computation would always fail!

There is one last feature worth mentioning about AH, its anti-replay capabilities. AH uses the sequence number to determine whether a packet has been seen before. The way it works is straightforward. When an AH connection is first established, the value is set to zero. Every time a packet is sent out, the number is incremented. So, the first packet has a sequence number of zero, the next 1, and so on. To prevent replay, the receiving system must make sure that it never accepts two messages with the same sequence number.

There is an additional wrinkle to this. The sequence number is a 32-bit value. This allows for over 4 billion different sequence numbers. Although this might sound like a large number, it is not inconceivable, given enough time, for it to be exceeded.

When this happens, the protocol specifies that the current key in use be renegotiated and that the sequence number value be reset to zero.

Encapsulated Security Payload (ESP)

ESP is the companion protocol to AH. Like AH, it offers message integrity, anti-replay, and authentication features, but it also offers confidentiality by providing the capability to encrypt the contents of the message. Its implementation differs from AH in the area within the packet that it concentrates on. ESP does not pay any attention to the IP header of the packet. It concentrates instead on the message contents.

Just like AH, ESP is designed to minimize its dependency on any particular encryption algorithm. To establish compliance with the IETF standard though, an implementation must support the following algorithms: Digital Encryption Standard (DES) for encryption, and HMACs based upon both MD5 and SHA-1 for authentication. Each implementation must also include the NULL algorithm for both encryption and authentication. The reason for the NULL algorithm will be explained shortly.

As stated previously, ESP provides confidentiality and authentication. You don't have to use both though. It is possible to use ESP to only perform authentication, or confidentiality, or both. Here's how.

When encryption is chosen, all of the information in the packet above the network level is encrypted using the selected encryption algorithm. This includes the embedded protocol header (for example, TCP, UDP, and ICMP) and all of the message data. The packet is then rewritten by replacing all of the transport data with the payload field of the ESP message.

If you do not need the message to be confidential, you can turn encryption off by using the NULL algorithm. This algorithm, as you might guess from the name, does nothing to the message. When used, an ESP message is still generated and placed into the outgoing packet. The only difference is that the message data contained within the ESP payload is still in its original form (for example, clear-text).

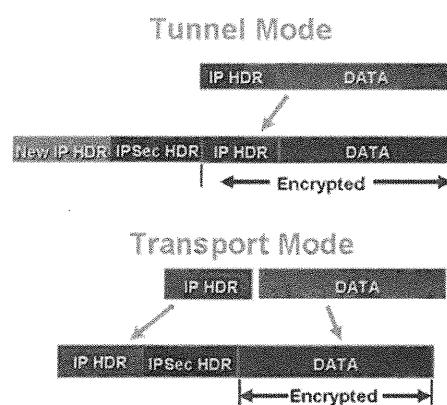
Authentication is performed similarly to the AH protocol, by creating and then verifying an ICV. The difference is what information is included in the ICV calculation. ESP authentication includes only the information in the ESP message, so the source and destination of the packet do not enter into the calculation. It does not matter whether the payload of the ESP message is encrypted or not. The calculation is the same.

Just as with ESP confidentiality, a NULL algorithm is available for ESP authentication. This algorithm acts differently than the NULL confidentiality algorithm. When it is called, instead of returning the same message that it was presented, it returns nothing. This results in the authentication field of the ESP message being empty.

There is one caveat worth mentioning about these NULL algorithms. You can use one or the other but not both. Using both would effectively disable ESP and for obvious reasons is not included in the standard.

Types of IPSec Modes

- Tunnel mode: Applied to an IP tunnel:
 - Outer IP header specifies IPSec processing destination
 - Inner IP header specifies ultimate packet destination
- Transport mode: Between two hosts:
 - Header after IP header, before TCP/UDP header



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Types of IPSec Modes

Both AH and ESP can operate in two modes: transport mode or tunnel mode. Transport mode is used to protect a conversation between two specific hosts on a network. For example, two hosts using ESP in transport mode are establishing a client-to-client or a client-to-site VPN. Up to now, all of our IPSec examples have been based upon transport mode. Tunnel mode is used to establish a site-to-site VPN. Let's look at how tunnel mode differs from transport mode for both AH and ESP.

How Tunneling Works

Tunnel mode, as the name implies, sets up virtual tunnels between gateways. Tunnel mode works by accepting an entire IP packet, which is then packaged in an IPSec packet. This new IPSec packet is not addressed to the destination of the packet it is carrying. Instead, its destination address is the address of the gateway system at the other side of the tunnel. When the destination gateway receives a tunnel packet, it un-packages it to get out the original packet. This packet is then routed onward to the host listed in its destination field. From this original packet's point of view, the trip across the tunnel represents just one hop, regardless of how many intermediate routers might have actually existed between the two gateways.

Tunnel Mode and AH

As in transport mode, AH provides authentication and integrity services for the packet. Implementation of tunneling mode AH is straightforward given our description of how tunneling works.

When a packet arrives at a gateway for passage across the tunnel, a new IP packet is created. This tunnel packet's header contains the source address of the gateway and the destination address of the remote side gateway. The data portion of the tunnel packet contains the original packet in its entirety.

Now, AH proceeds exactly the same as transport mode AH. An ICV is computed based upon the fields in the tunnel IP packet including the data field, which includes our original packet. The ICV is placed just after the new packet header and before the data field. When the packet arrives at the destination gateway, the ICV value is recomputed.

If it matches, it proves that the packet has not been tampered with while it traveled through the tunnel. This includes proving that the original packet has not changed, and that the fields of the tunnel packet are genuine. The gateway can now remove the original packet from the data field of the tunnel packet and send it on its way.

Tunnel Mode and ESP

ESP tunnel mode works similarly to AH tunnel mode. When a new packet arrives at a gateway, it is packaged inside a tunnel packet that is addressed to the remote gateway. Encryption and authentication algorithms are then run on this new packet's data field, thus protecting the original packet. Note that this does not protect the header of the tunnel packet. The resulting tunnel packet includes the new IP header addressed to the remote gateway, and an ESP message, which includes the cipher-text and authentication data for the original packet.

Session Establishment

There are many options available within IPSec. Before an IPSec connection can be created, the two sides of the connection must agree on what options they are going to use. In addition, many of the options require the exchange of other information, such as session keys and sequence numbers. Session establishment negotiates these details. The agreements from these negotiations are called "Security Associations."

Security Associations

Security Associations (SAs) are a critical part of IPSec. They document the security services (called "transforms") that a particular IPSec connection is using. These details include the IPSec protocol being used (AH or ESP), the authentication mechanism that is going to be used (for example, HMAC-MD5), which cryptographic algorithm to employ, the length of the key used in the cryptographic algorithm (for example, 56-bit), what security services are being applied (for example, authentication and confidentiality), and any other details necessary to fully describe the security services of the connection. Each IPSec connection must have an SA set up prior to beginning communication.

SAs are unidirectional. A single SA describes only transforms for one side of a network conversation. To establish a two-way conversation, two SAs are required: one to allow packets to be protected from point A to point B, and the second to allow packets to be protected from point B to point A. These SAs are normally set to use the same transforms, but this is not actually required.

There is nothing to stop an implementation from using different transforms on each side of the conversation: for instance, encrypting one direction with 3DES, but leaving the other direction unencrypted. This would normally be undesirable!

Internet Key Exchange (IKE)

Internet Key Exchange (IKE) is the protocol used by IPSec to negotiate the session details of a connection and then document them as SAs. IKE is a hybrid protocol composed of a key management framework and a key exchange protocol. These are the Internet Security Association and Key Management Protocol (ISAKMP) for key management and the Oakley Key Determination Protocol (Oakley) for key exchange. IKE is occasionally referred to as "ISAKMP/Oakley." Elements of a third protocol called "Secure Key Exchange Mechanism (SKEME)" are also used to extend the capabilities of Oakley.

The negotiation occurs in two phases. In phase one, a secure, authenticated connection is established to protect the conversations that will occur next. This is extremely important because the security of all future conversations relies upon the capability of the two sides of the connection to privately exchange keys and other security details. Phase one provides this privacy. The results of phase one are recorded in a special SA (ISAKMP-SA) that is used only to protect ISAKMP conversations. In phase two, the security services and details for an SA are negotiated over the ISAKMP-SA.

There are two methods that can be used to accomplish phase one, referred to as main mode and aggressive mode. The difference between them is that main mode checks the identity of the participants, and aggressive mode does not. Identity protection sounds like a good thing and it is. So why would we go without it? If public key cryptography is used to set up the ISAKMP-SA, identity can be inferred. If side A of a conversation can decrypt side B's messages using side B's public key, we can assume that the message was generated by B because only B should have B's private key. This provides the identity protection indirectly, making it unnecessary for ISAKMP to perform a special operation to check it.

Phase two also has multiple modes but the primary one is quick mode. This is the mode that is used to negotiate the security details for the ESP and AH SAs. This is also the mode that is used to re-key connections when the keys have been in use for too long.

SSL VPNs

- Fastest growing, have less operational problems than IPSec, cryptographically equivalent, but from an application perspective not quite as secure
- Ideal if you have multiple vendors and all you need is a browser for client side. Portal VPNs work with almost any browser. SSL Tunnel VPNs require modern browsers that can handle active content.
- Problems include opening firewall ports, application vulnerabilities, authentication, and the attack surface of the browser

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

SSL VPNs

If your organization is considering purchasing a VPN, don't ignore SSL-based technologies. In the past, IPSec VPNs were the overwhelmingly dominant standard. Today, SSL-based VPNs are more often the choice. They tend to have lower operational cost and fewer problems. An AES-based IPSec VPN is possibly stronger from an application security perspective; however, from a cryptographic standpoint, they are equivalent.

There are security issues related to SSL VPNs. These include the fact that you have to open ports in your firewall, probably 443 and 80. Because SSL VPNs use web technology, security weaknesses in web browsers, and web servers could affect the VPN; however, if the primary purpose of the VPN is remote dial in and protection from casual eavesdropping, SSL is sufficient. There are also potential problems with the authentication that can never be solved until we teach the user to never accept an unknown certificate. SSL portal VPNs work with essentially any modern web browser. Specifically, they work with browsers whether or not the browsers allow (or support) active content. Thus, SSL portal VPNs are accessible to more users than SSL tunnel VPNs.

An SSL tunnel VPN allows a user to use a typical web browser to securely access multiple network services through a tunnel that is running under SSL. SSL tunnel VPNs require that the web browser be able to handle specific types of active content (for example, Java, JavaScript, Flash, or ActiveX) and that the user be able to run them.

(Most browsers that handle such applications and plug-ins also allow the user or administrator to block them from being executed.)

The "tunnel" in an SSL tunnel VPN is both similar and quite different from the tunnels seen in typical IPSec VPNs. The two types of tunnels are similar in that almost all IP traffic is fully protected by the tunnel, giving the user full access to services on the network protected by the VPN gateway. The tunnels are quite different in that SSL/VPN tunnels are usually created in SSL using a non-standard tunneling method, whereas IPsec tunnels are created with methods described in the IPsec standard.

At the completion of each session, sensitive information might remain on the user's computer in temporary Internet files. If you are purchasing a commercial SSL VPN, make sure that it has the technology to clean up after sessions.

If you are considering the purchase of SSL VPNs, consider these procurement requirements:

- SSL VPN manageability features such as status reporting, logging, and auditing should provide adequate capabilities for the organization to effectively operate and manage the SSL VPN and to extract detailed usage information.
- The SSL VPN high availability and scalability features should support the organization's requirements for failover, load balancing, and throughput. State and information sharing is recommended to keep the failover process transparent to the user.
- SSL VPN portal customization should allow the organization to control the look and feel of the portal and to customize the portal to support various devices such as personal data assistants (PDA) and smart phones.
- SSL VPN authentication should provide the necessary support for the organization's current and future authentication methods and leverage existing authentication databases. SSL VPN authentication should also be tested to ensure interoperability with existing authentication methods.
- The strongest possible cryptographic algorithms and key lengths that are considered secure for current practice should be used for encryption and integrity protection unless they are incompatible with interoperability, performance, and export constraints.
- SSL VPNs should be evaluated to ensure they provide the level of granularity needed for access controls. Access controls should be capable of applying permissions to users, groups, and resources, as well as of integrating with endpoint security controls.
- Implementation of endpoint security controls is often the most diverse service among SSL VPN products. Endpoint security should be evaluated to ensure it provides the necessary host integrity checking and security protection mechanisms required for the organization.
- Not all SSL VPNs have integrated intrusion prevention capabilities. Those that do should be evaluated to ensure they do not introduce an intolerable amount of latency into the network traffic.

Pretty Good Privacy (PGP)

The student will understand the functionality of the PGP cryptosystem and how they operate.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Pretty Good Privacy (PGP)

This section intentionally left blank.

Confidentiality in Storage

Pretty Good Privacy (PGP)

- Started out as a way to bring privacy to public communication medium:
 - Protects files on hard drives
 - Protects files transferred via e-mail
 - Provides file/folder level encryption

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Confidentiality in Storage

PGP is a great example of an application-specific use of cryptography. The current commercial version of PGP supports encryption and the creation of digital signatures for files, provides IPSec-compliant VPN capabilities, and acts as a host firewall, but PGP's original purpose was to protect e-mail.

PGP provides two main protections for e-mail. First, it supports strong encryption of the e-mail message. This encryption is implemented using a combination of public key and symmetric key cryptography. The second protection is digital signature of e-mail messages, providing non-repudiation and integrity verification.

In this section, we concentrate on PGP as an e-mail security tool. We discuss why it was created and how it works. We also discuss one of the most interesting parts of PGP: how trust is established between PGP participants.

History and Purpose

In 1991, Phil Zimmerman created PGP. He was motivated by the potential passage of a bill that would have forced providers of cryptographic systems to include back doors into their products to allow the Government the ability to decrypt anyone's private messages. Although eventually defeated, the bill provided Zimmerman the incentive to create the first version of PGP, a program designed to provide confidentiality for e-mail messages. This political stand was to come at a high price.

In 1991, the U.S. government considered cryptographic systems munitions and enforced strict export controls on them. PGP falls into this category. Its export was punishable by up to ten years in jail and a fine of \$1 million, but nonetheless, it was exported sometime in 1991. The question though was by whom?

As a protective measure, prior to the vote on the previously mentioned bill, friends of Zimmermann who had copies of PGP given to them began posting PGP on Internet sites and Bulletin Board Systems (BBSs). These were U.S. sites, making the activity legal. At some point, though, someone downloaded the product and transferred it overseas using the Internet. The cat was out of the bag.

Two years later, in February 1993, just after the release of PGP version 2.0, U.S. Customs got interested in investigating PGP's release to the world. A grand jury was created to investigate the charge that Zimmermann allowed his creation to be exported. This was despite the fact that it was clear that Mr. Zimmermann did not perform the (legal) uploading of PGP to the BBSs and did not condone or participate in its eventual transfer outside the country.

The government's investigation caused a huge outcry on the Net. Many people spoke out in support of Zimmermann and a defense fund was set up to help defray the sizable legal costs he incurred. The investigation dragged on for close to three years, but was eventually dropped.

Today, most export restrictions for PGP have been lifted, making PGP legally available both inside and outside the U.S. Commercial versions are available from the PGP Corporation (<http://www.pgp.com>), which has an export license allowing them to sell to most countries. Open-source versions of PGP are also available (<http://www.pgpi.org>), including a version called "GNU Privacy Guard," which is released under the GNU General Public License making it free for both private and commercial use.

How PGP Protects E-Mail

PGP provides two security services for e-mail messages: confidentiality through encryption, and message integrity and source identification through digital signatures.

Encryption

A problem with using encryption for e-mail is that encryption requires some shared information between sender and receiver. Using symmetric key algorithms, both participants need to share a secret key. This key needs to be private to the two participants; otherwise, a third-party would be able to decrypt the exchanges between them. Establishing a shared secret key prior to sending a message can be inconvenient when sending a message to someone you know, but can be impossible if you need to send a message to someone you might never have met. This makes a purely symmetric key system a bad choice for e-mail.

Public key is a better choice for this key exchange. Because public key systems separate the key into two pieces: a public piece, which you can safely distribute to the world, and a private piece, which you do not reveal, it becomes possible to exchange messages with anyone as long as both know each other's public key. Sounds better, but there is a major downside to public key cryptography. It's *slow!*

On-the-fly Encryption (Full Disk Encryption)

- Encrypted files are decrypted to read, and then encrypted back to hard drive
- If system is turned off and computer is stolen, no one can read your encrypted messages
- If computer is on, anyone can decrypt your encrypted messages
- You should know what threat you are protecting against

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

On-the-fly Encryption

When a user accesses an encrypted file, the file is decrypted for reading. When the user closes the file (in other words, it is no longer being accessed), the file is re-encrypted back to the hard drive of the system. What this means is that the system, and consequently the encryption routines employed on the system, is accessing the file based on the context of the user. If the system is turned on, and the user is logged in, the user's permissions will dictate whether the file can be decrypted or not. If the user is authorized to decrypt the file, the file will be readable to anyone who accesses the system while the user is logged into it. If the system is turned off and gets stolen, no one can access the file without logging in as the authorized user.

Establishing a Key

- Generate a public/private key pair:
 - Diffie-Hellman/DSS or RSA
 - Key length/size
 - Key expiration

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Establishing a Key

To avoid the performance penalty of public key cryptography, while still allowing its use, PGP takes a hybrid approach. It creates a random symmetric key that it uses to encrypt the message, and then encrypts this key with the recipient's public key. Upon receipt, the recipient can decode the message by decrypting the symmetric key with his private key, and then using the symmetric key to decrypt the message. This provides a fast solution that allows easier establishment of trust between sender and receiver.

Digital Signatures

PGP can also digitally sign a message, verifying the integrity of the message, and the identity of who sent it. PGP digital signatures are created in a two-step process. In the first step, the information being signed is submitted to the SHA-1 cryptographic hash algorithm. The resulting hash is then encrypted using the sender's private key. The result is the digital signature, which can be sent with the original message allowing recipients to verify the validity of the message. Verification of the message is performed by decrypting the digital signature using the sender's public key to get the SHA-1 hash. A new hash is then computed on the received message and compared to the decrypted hash. If they match, then the message is genuine.

Choosing a Passphrase

- Most critical part of key generation
- Use strong password principles:
 - Many characters
 - Mixed case, alphanumeric, special characters
 - Easy to remember, hard to guess

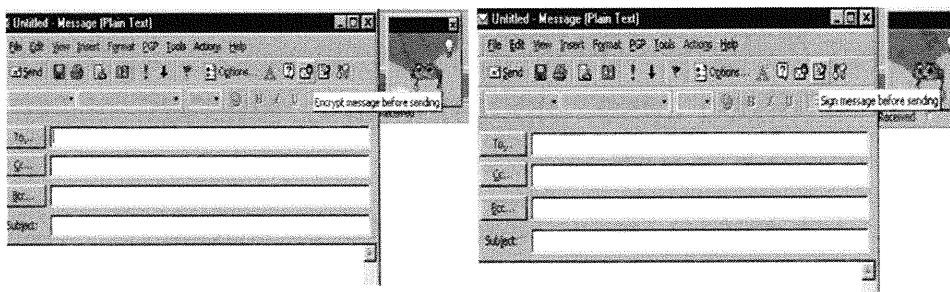
SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Choosing a Passphrase

The last piece of information needed is a passphrase. The private portion of the key, which will be generated, needs to be stored on the disk of the computer that will be sending protected e-mails. Without additional protective measures, anyone with access to the computer would be able to copy it. Compromise of a user's private key would allow the compromiser to decrypt every message ever encrypted using the key. Because of this, PGP takes the extra step of encrypting the private key, using the passphrase that you supply. The passphrase should be composed of letters, numbers, and symbols and should be fairly long. Take the time to choose a good passphrase, but take even more time to make sure you are not going to forget it. Without the passphrase, any data you have encrypted with your key will be inaccessible to everyone, *including* you!

Encrypting Outbound E-Mail

- To encrypt or sign e-mail, it is as easy as clicking an icon before you send:



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

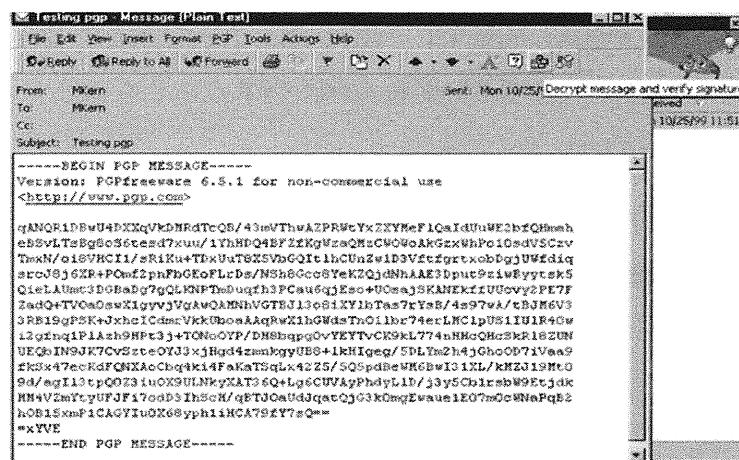
Encrypting Outbound E-Mail

Depending on which e-mail program you use, encrypting mail with PGP can be as simple as clicking a button, or as hard as externally encrypting the message, and then pasting the resulting ciphertext into the body of the e-mail. Programs that currently support PGP natively include Microsoft Outlook and Lotus Notes. For these programs, encryption and signing is automatic (depending on how you configure it), as long as you possess valid certificates for the recipients.

Other programs can still be used to send PGP messages. They just do not automatically perform the PGP operations for you. The following steps can be used to manually encrypt your e-mail messages, regardless of which mail program you are using:

1. Click the *encryption* icon of the PGPtols application.
2. From the clipboard, click *encrypt*.
3. Select and copy the text you want to encrypt into the clipboard. This can be done by highlighting all of the text you want encrypted and clicking *Edit, Copy*.
4. Select encryption from the PGPtols application by clicking the *encrypt* icon (the icon with the envelope covered by a closed lock). PGPtols is part of the commercial release of PGP and is used as its control panel for performing encryption, decryption, signing, and verification.
5. From the clipboard on the dialog that displays after you select encryption, click *encrypt*.
6. Select the certificates of your intended recipients from your public key ring. It is always a good idea to include yourself as one of the recipients.
7. Click the *OK* button to perform the encryption. PGP replaces the cleartext contents of the clipboard with the encrypted ciphertext.
8. To finish, paste the encrypted message into your e-mail document by highlighting all of the text in the e-mail message again and clicking *Edit, Paste*. This replaces the original text with the ciphertext. The message can now be mailed normally.

Sample PGP-Encrypted E-Mail



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Sample PGP-Encrypted E-Mail

This section intentionally left blank.

Decrypting Inbound E-Mail

- To decrypt, either copy to clipboard or double-click the attachment
- PGP window opens
- Enter your passphrase
- The net of the two actions, the encryption with the public key and decryption with the private key, results in the cleartext message

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

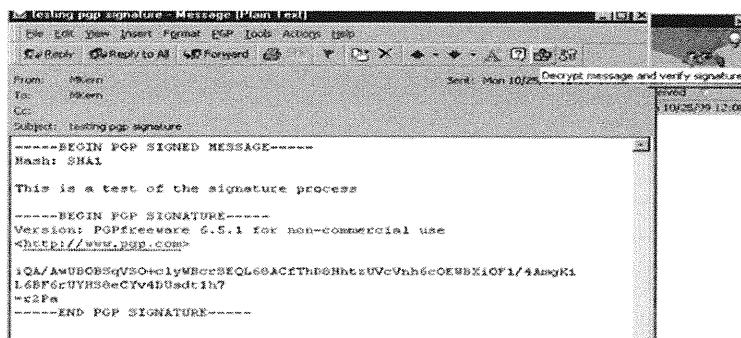
Decrypting Inbound E-Mail

Manually decrypting the message is performed similarly:

1. Highlight the PGP message and copy it to the clipboard.
2. Using PGPtols, click the *decrypt* icon. This icon appears as an envelope covered by a letter and an open lock. Clicking it opens a dialog box asking for a file to decrypt/verify.
3. Click the *decrypt from clipboard* button. Clicking this button brings up a window showing the list of people the message was encrypted to, and a request that you supply your pass phrase.
4. Type your passphrase. Assuming you type your passphrase correctly, the message is decrypted and displayed in a new window.

Signing Outbound E-Mail

- Use the passphrase to access the private key
- Apply a signature based on that private key



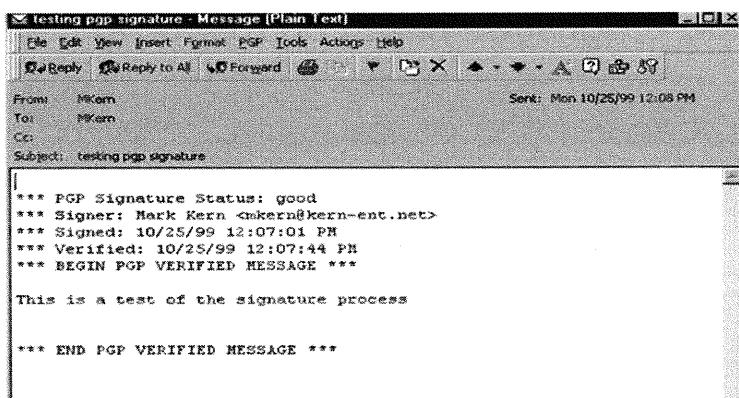
SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Signing Outbound E-Mail

Creating a signature for a message follows the same process that you use to encrypt one. The only difference is the result. The message that you want to sign needs to be sent to PGP either through the clipboard or by placing it in a file that PGP can read. Using PGPtols, you sign by clicking the icon that appears as a letter with a pencil pointing at the bottom. This causes the Select File dialog box to display. Choosing the clipboard button causes the passphrase dialog to display. Typing the correct passphrase for your certificate allows PGP to compute a signature for the current contents of the clipboard, which will be replaced by the signed message. The signed message can now be pasted into your e-mail window.

Confirming a Signed E-Mail

- Choose to "verify" the signature



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Confirming a Signed E-Mail

Verification occurs just like decryption. It even uses the same PGPtols button. The result is slightly different, though. The resulting window shows the message and information about its validity, including the results of the verification, the name and e-mail address of the sender, the date the message was signed, and the date it was verified by you.

The combination of PGP's capability to protect e-mail (and other) messages while allowing convenient key distribution has made it one of the most popular encryption tools available. It is easy to use, widely available, and secure. If you have not already experienced it, you should strongly consider giving it a test drive.

Public Key Infrastructure (PKI)

The student will have a high-level understanding of how PKI cryptosystems are used for secure communications.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Public Key Infrastructure (PKI)

This section intentionally left blank.

What is the Business Value of a Public Key Infrastructure?

- PKI provides a technical mechanism for encrypting an organization's data
- A hierarchy of infrastructure systems is used to create digital certificates
- Digital certificates are used to encrypt data
- A PKI provides a managed infrastructure for:
 - Creating certificates
 - Maintaining certificates
 - Revoking certificates

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

What is the Business Value of a Public Key Infrastructure?

Public Key Infrastructure (PKI) is the tool most often used for ecommerce and business-to-business (B2B), and it allows users to exchange encrypted information over a public network. When you purchase goods on the Internet, you privately and securely exchange data and currency (like a credit card number) with an online vendor through the use of a public and a private cryptographic key pair. That cryptographic key pair is obtained and shared through a trusted authority.

Familiar trusted authorities include RSA, which has developed the main algorithms used by PKI vendors, Verisign, which acts as a certificate authority and sells software that allows a company to create its own certificate authorities, and Thawte, which offers certificates for online merchants, and e-mail certificates.

We are familiar with PKI on public network like the Internet, but PKI can also be utilized inside of organizations. Those students who are involved with the U.S. military are familiar with the Common Access Card (CAC). These physical cards not only provide general identification, they are used for authentication to enable access to Department of Defense (DoD) computers, networks, and certain DoD facilities. CACs contain a certificate for each authorized user that facilitate the use of PKI authentication tools, and establish an authoritative process for the use of identity credentials. These cards also enable encrypting and cryptographically signing e-mail, but we are focusing on PKI right now.

Think about the sheer volume of people associated with the United States Department of Defense. We have:

- Active service members
- Inactive service members

- Reserve personnel
- Civilians
- Authorized contractors

A sophisticated and manageable infrastructure is needed to provide access to information and ensure only the right people have access to the right information (think concept of least privilege). Imagine that you are a system administrator at DoD trying to manage access of service members stationed across the globe!

A PKI infrastructure allows an organization to create certificates to facilitate authorized access. A hierarchical certificate structure simplifies maintaining certificates as well as removing access when a user changes jobs or leaves an organization.

Certificates

- An essential part of PKI
- Digital document attesting the binding of an entity to a public key
- Unique to each entity
- Equivalent to a passport or driver's license
- Mitigates impersonation

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Certificates

The cornerstone of the public key technology is the ability to distribute public keys to large populations while conveying trust that the certificates are associated with a user and the user's public key. A *certificate* is the way by which trust is distributed appropriately throughout the environment.

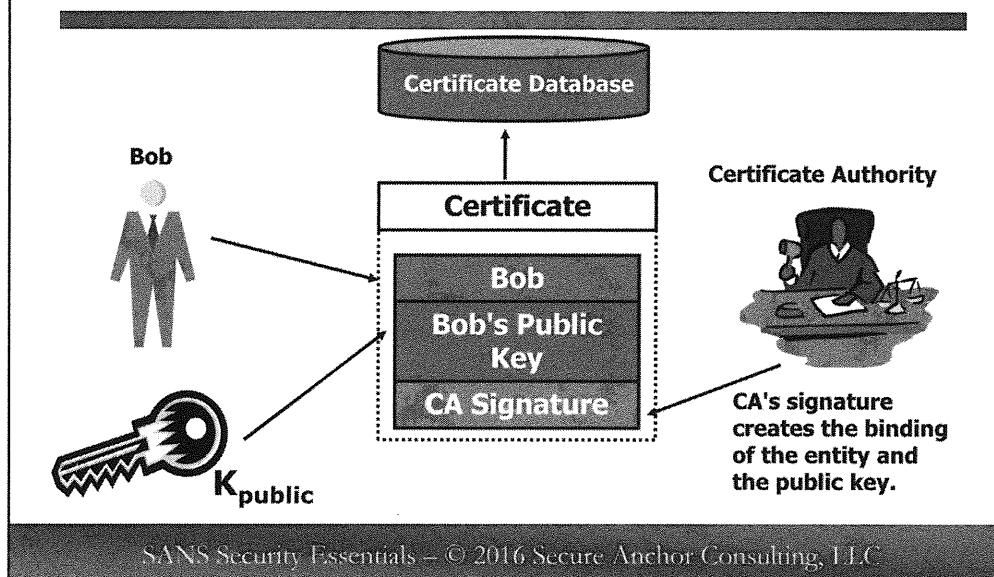
The certificate itself is signed by a PKI authority, or certificate authority (CA), which "everyone" has agreed to trust. Everyone, as it is used with quotes in the previous sentence, means those individuals within a given operating domain, such as an industry, company, organization, or agency that have agreed upon a common law and trust system. Where there is no agreement, problems arise with interoperability of public key infrastructures, as discussed previously.

X.509 stipulates the format and structure of the certificate and what the certificate must contain. The X.509 certificate format is used in Secure Sockets Layer/Transport Layer Security (SSL/TLS), Secure Multipurpose Internet Mail Extensions (S/MIME), Internet Protocol Security (IPSec), and Secure Electronic Transaction (SET), to name a few.

Certificates are meant to be equivalent to a passport or driver's license, at least in the domain for which it is issued. Passports are usually issued by nation states, which implies an overarching governing mechanism over a large geographic region. PKI currently does not share this type of overarching reach. In many ways, PKI represents islands of self-regulation where either a company or a collection of companies agrees on some sort of trust mechanism. A planet-wide PKI, or even a nationwide PKI, does not currently exist.

Certificates are intended to mitigate impersonation. A third-party signs a user's certificate with its private key, in effect stipulating that the third-party has done a thorough background check of the entity to make sure she is who she says she is.

Certificate —The Easy Picture



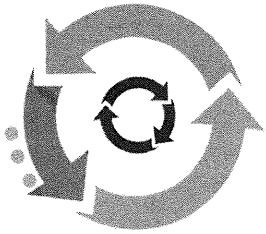
SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Certificate —The Easy Picture

So far we've been referring to a third-party that signs the user's certificate. That third-party is called a *certificate authority* (CA). A certificate authority has many responsibilities, not least of which is issuing certificates signed with the CA's private key. In the preceding illustration, Bob and Bob's public key are included in a certificate. After the CA has established the validity of Bob's identity, it signs the binding, thus creating the public certificate. The certificate is typically stored in a publicly accessible directory.

Operational Goals of PKI

- Operationally, a PKI must manage the certificates that it issues
- The traditional PKI certificate lifecycle includes:
 - Certificate registration
 - Certificate creation
 - Certificate distribution
 - Certificate validation
 - Certificate key recovery
 - Certificate expiration
 - Certificate revocation



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Operational Goals of PKI

Key Management and Certificate Lifecycles

There are many steps along the lifetime of a certificate. Each is important to the maintenance of security within the PKI. On this slide, we cover:

- Registration and initialization
- Certification
- Storage
- Escrow

Registration is the process that occurs before a certificate is issued. It involves the person or entity who wants the certificate providing his identification information in the form of a Distinguished Name (DN), and some definitive proof that he is indeed the person represented by the DN.

Next comes initialization. This step provides the person the details he needs to communicate with the PKI, including a copy of the root CA's certificate. Initialization is also where the client's public/private key-pair is generated. Depending upon the policy being followed, this key generation might be performed by the person or by the CA. If performed by the person, the public key needs to be sent to the CA. If performed by the CA, the keying material (public and private) needs to be carefully sent to the person. In either case, the public key becomes associated with the person, and the person must be validated for the key to be valid.

Certification occurs when the CA actually issues the certificate, which includes the user's DN, public key, and certificate details such as validity period, protected by a signature generated by the CA. At this point, the certificate can be stored in a certificate server, such as an LDAP, or simply issued to the person to use and share as he wants. There are several facets of a key storage discussion:

- Public keys
- Private client-side keys
- Private server-side keys
- Private CA Root and Subordinate keys

First, public keys are just that—public. It is not only okay to share public keys, but it is encouraged. The public key can be used to determine authenticity of messages, can serve as part of a non-repudiation scheme, and can be used to encrypt messages, which only the owner of the key can decrypt. The success of the entire infrastructure is based on the availability of the public keys, so they must be stored where everyone can get to them. It is important to note that a certificate ties a public key to an individual or a single entity, so that the certificate/public key becomes an identifier. Public keys/certificates are, therefore, often stored in registries so that others can look up another user's certificate.

For PKI to be trustworthy—in other words, for it to work at all—adequately controlled secure key storage is critical for each client's private keys. As new client-side private keys are imported into a key store, users can protect their certificates and private keys with passwords. These passwords can be used simply to keep someone else from exporting (or stealing) their certificate and/or private key. Or the certificate can be stored in such a way that a password is required before the key can even be accessed and used. In fact, certificates can be stored as non-exportable, so that no one can export the certificate or key once it is installed. All of this makes compromise and masquerading more difficult for the malicious user. In any event, it is absolutely critical that a user's private keys be protected at all costs. They must always remain in the user's possession. Through revocation, discussed next, the victim can have her certificate revoked and obtain a new certificate so that the original is no longer useful to anyone, even if the password is cracked.

Users should be aware of changes in the local environment where their keys are stored. If anything causes a user to be concerned that her certificate or key might be compromised, the best solution is to have the certificate revoked and obtain a replacement certificate. Server-side private keys, such as those associated with SSL, must also be protected adequately to preserve the integrity of the messages between the client and the server. If the private key is known by another entity, it is possible for a man-in-the-middle attack to take place where the encrypted stream of data can be intercepted and decrypted by a third-party without the knowledge of the two parties who originated the conversation.

Perhaps the most important facet of key storage pertains to the private keys used to create the Root and Subordinate Certificates for the Certificate Authority. The entire infrastructure becomes useless if these private keys are not carefully protected. If a CA's private keys are compromised, certificates created by that CA cannot be trusted. Anything signed by a compromised key is invalidated and unreliable. For this reason, reputable and trusted Certificate Authorities will actually conduct a "Key Ceremony" where multiple parties witness the creation of and physical protection of the new keys for Root and Subordinate Certificates. It is not uncommon for a CA to even videotape the proceedings to ensure the greatest control possible, up to and including the deposit of the private keys physically into a safe or other secured physical location.

Finally, will the keys be stored by software or hardware?

Software key stores, for example, include client browsers. These software key stores are adequate for keys at lower risk, such as an individual user's private key. There have been demonstrations that software key stores can be attacked, such as the research done at Princeton and referred to as the "cold boot attack." Higher risk private keys are commonly protected by hardware. Examples include the Trusted Privacy Module (TPM) that is supplied on modern computers and networking gear. It is tamper resistant and has a security focused protocol to retrieve or store keys. Another common hardware storage method is a smart card.

The U.S. Department of Defense calls the smart cards they use “Common Access Cards (CAC)”[2]. Even higher risk private keys, especially for Root and Subordinate Certificates, might be stored in hardware modules created for that purpose. These hardware modules, typically an add-on component for a computer, are the electronic equivalent of a safe. Other controls might be used to further increase the security of such keys, such as requiring that multiple persons each maintain only a portion of the passphrase required to access the keys, and that no one person knows the entire passphrase. The controls used to protect private keys should be commensurate with the value of the information protected by those keys. The more valuable the information, the more resources the data owner should be willing to spend to protect it.

Key escrow is the storage of keys with some trusted third-party for it to hold, in case the keys are needed but are otherwise inaccessible. This might also be referred to as “key backup.” Key escrow could also be requested by law enforcement so that it can access encrypted information as needed. Key escrow with law enforcement, therefore, is not a popular concept among civil libertarians because of the juxtaposition of public interests versus privacy and individual freedoms.

Key Management and Certificate Lifecycles

In the case of a certificate expiration, the CA need only issue a new certificate for the person. Most CAs set certificate expiration at 1 or 2 years, although shorter time periods can be specified or requested for special purposes. Certificate lifetime is kept this short for a purpose. If we extend the lifetime, we increase the risk that the certificate could be compromised. By expiring certificates on a regular basis, we ensure that users who no longer need access to the data will not have that access after a specified time. In this way, the PKI system cleans up after itself in a mandatory way that cannot be altered or bypassed. Expired certificates are known by all PKI participants to be invalid because today's date is beyond the expiration date on the certificate. But what about certificates that need to be changed before the expiration date?

Certificates may be revoked for a number of reasons. Here are a few:

- User terminated from employment
- User moves to a new position no longer requiring the access provided by the certificate
- User changes e-mail address or name or other important information
- Suspected key compromise

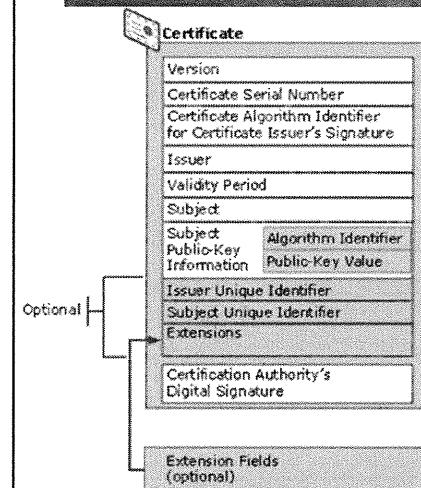
To revoke a certificate, the CA maintains a Certificate Revocation List (CRL). The CRL consists of a list of the certificate serial numbers for all of the certificates that have been revoked by the CA. This list needs to be regularly updated and sent to each of the PKI participants.

When checking a certificate's status, the first check should be to verify that the certificate's serial number is not listed on the latest CRL. One problem with this is the frequency of CRL distribution. When a certificate is revoked, there will be some period of time between its invalidation and the receipt of all of the PKI members of a CRL, which references the certificate. During this period, it is possible that the certificate might be accepted when it should not have been. The solution is to increase the frequency of CRL distributions, but distributing it too much might consume too many network and system resources, so a balance must be made between security and operations.

Key Management and Certificate Lifecycles

Key recovery is also an important part of many PKIs. Remember that if you lose your private key, all of the information encrypted with that key is lost as well. To prevent this, some CAs store a copy of the person's private key. Although this does somewhat undermine the non-repudiation of the key, it does allow the key to be recovered if the person loses it. Key recovery is particularly important in organizational settings where the information that is being protected is owned by the organization, not the individual. If the individual leaves the company, or is simply unavailable, the backup key can be used to recover the materials the individual was working on. Other reasons for key recovery include forgotten password for an encrypted file, death of an employee who has encrypted data, or someone attempting to hide criminal activity from law enforcement.

Digital Certificates



- Standard for digital certificates is the x.509 certificate
- Each certificate contains:
 - Demographic data
 - Validity period
 - Supported encryption algorithm
 - Public/private key
 - Signature by issuing CA
- Public or private keys can be used for multiple forms of encryption

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Digital Certificates

A digital certificate is a credential used to help someone decide whether a key is genuine. It works by binding a public key with identification information such as name and e-mail address. This information is then signed by at least one third-party. As long as you trust the opinion of one of the third parties that signed the certificate, you should be able to trust the validity of the certificate.

Digital certificates bind an individual's identity to the public key. With PKI systems, the purpose is the same, but the process used to produce the certificate is more formal. Most PKI systems do not allow the user to create certificates themselves like PGP does. Instead a certificate authority creates the certificate and issues it to the user. The care at which the CA performs this role directly affects how secure the overall PKI is. If the CA issues a digital certificate to anyone without requesting proof of identity, the confidence you should have in the certificate is low. If instead, the CA requires that you show up, in person, with two forms of government issued ID before issuing you a certificate, your confidence can be high in that CA's certificates.

Most current PKI systems produce certificates in the X.509 certificate format. This specification is published by the International Telecommunications Union (ITU), an international standards body. Most certificates follow the X.509 version 3 standard. Each X.509 certificate includes two sections: the data section and the signature section. The data section holds all of the details associated with the certificate, including the following fields:

- X.509 version number
- Serial number
- Identity information of the certificate's owner in the form of a distinguished name (DN)
- Owner's public key, and the algorithm used to generate it.
- Period that the key is valid (for example, 12:00 midnight Nov 1, 2002 through 12:00 midnight Nov 30, 2004)
- Identity information of the issuing CA

The certificate can also include other details, sometimes referred to as “certificate extensions,” that are application dependent. An example is X.509 certificates used in SSL connections. With SSL, the X.509 extensions include a certificate type used to distinguish between certificates issued to browsers and certificates issued to servers. The documentation that specifies how certain certificates are to be used is called the *Certificate Policies document*.

Secure Web Traffic (SSL)

- One use of PKI is to encrypt messages between a Web server and a web browser
- This is accomplished by the use of either:
 - Secure Sockets Layer (SSL)
 - Transport Layer Security (TLS)
- Client and server use a PKI certificate (asymmetric) to negotiate a session key (symmetric)
- PKI certificate is used for secure key exchange
- Session key is used to encrypt data between systems
- SSL/TLS is expanding today into more than web sites

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Secure Web Traffic (SSL)

Cryptographic protocols can provide security and data integrity over TCP/IP networks. Two such protocols, TLS and SSL, encrypt the segments of network connections at the Transport Layer.

Secure Socket Layer (SSL)

The Secure Sockets Layer (SSL) protocol was developed by Netscape Communications in 1994 to provide application-independent secure communications over the Internet. SSL procedures are most commonly employed on the web with the Hypertext Transfer Protocol (HTTP) for ecommerce transactions, although SSL is not limited to HTTP. SSL uses cryptography to provide message privacy, message integrity, and client and server authentication, and operates on TCP port 443.

Transport Layer Security (TLS)

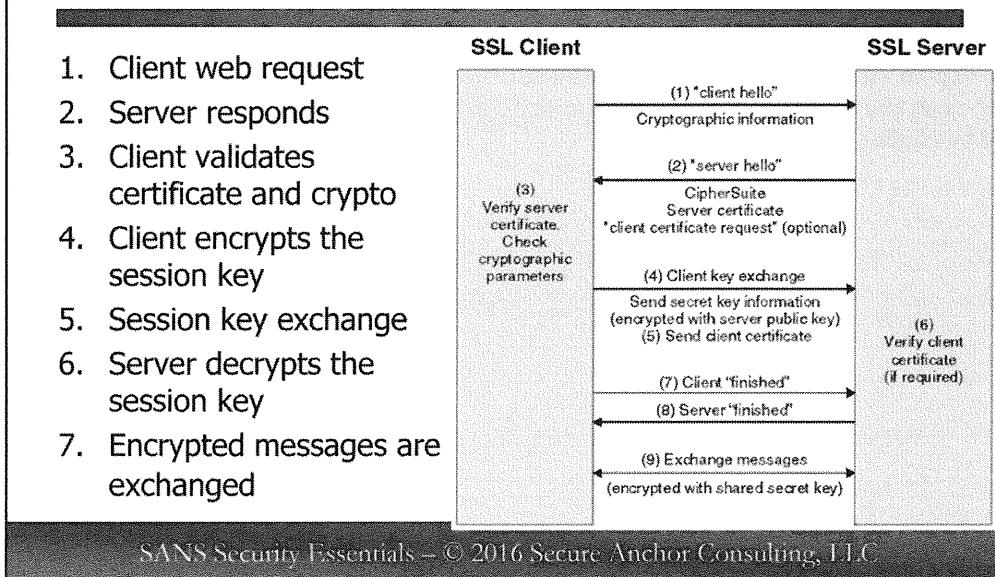
TLS protocol allows client/server applications to communicate across a network in a way designed to prevent eavesdropping/ tampering by providing endpoint authentication and communications confidentiality over the Internet.

Both SSL and TLS protocols provide for:

- Key Establishment
- Confidentiality with Triple DES-EDE-CBC (and other protocols considered less secure) and TLS supports AES
- Signature (RSA, DSA) and TLS are considering Elliptic Curve with a draft Internet standard
- Hash MD5, SHA-1

In most web-browsing situations, communications between the web browser and the server is unilateral, meaning the client knows the server's identity, but not the other way around. The Web-browser client is unauthenticated or anonymous. Server authentication means that the browser has validated the server's certificate (for example, checked the digital signatures of the server certificate's issuing CA-chain). When validated, the browser might display a security icon. In Microsoft Explorer, this is a "closed padlock." But validation does NOT "identify" the server to the end-user. For true identification, an end-user has to scrutinize the identification information contained in the server's certificate (and indeed its whole issuing CA-chain). The locked padlock icon has no relationship to the URL, DNS name, or IP address of the server. Such a binding can be securely established only if the URL, name, or address is specified in the server's certificate itself. This distinction makes it very difficult for end-users to properly assess the security of web browsing. This is not a shortcoming of the TLS protocol—it is a shortcoming of PKI.

PKI SSL Crypto: An Illustration



PKI SSL Crypto: An Illustration

PKI SSL Handshake

At the beginning of an SSL session, an SSL handshake is performed. An HTTP-based SSL connection is always initiated by the client using a URL starting with <https://> instead of <http://>. This handshake produces the cryptographic parameters of the session.

1. **Client Web request:** “Hello, let me tell you about myself. I will tell you about my version of SSL, the cipher protocols I can support, and data compression methods I understand.” The message also contains a 28-byte random number.
2. **Server responds:** “Hello there. I can understand many different cipher protocols. I will pick the one we both can understand, and a data compression method. I will also provide a session ID, and another random number. I sending you my public key, NOT my private key. You can’t see that.”
3. **Client validates certificate and crypto:** The client reviews the information sent by the server to authenticate the server. The client looks at the public key and sees the signature from the CA. If the server can be successfully authenticated, the client proceeds to step 4.
4. **Client encrypts the session key:** Using all data generated in the handshake to this point, the client (using cipher suggestion of the server) creates the pre-master secret for the session, encrypts it with the server's public key (obtained from the server's certificate, sent in step 2), and then sends the encrypted pre-master secret to the server.
5. **Session key exchange:** Both the client and the server use the master secret to generate the session keys, which are symmetric keys used to encrypt and decrypt information exchanged during the SSL session and to verify its integrity.

6. **Server decrypts the session key:** The client sends a message to the server informing it that future messages from the client will be encrypted with the session key. It then sends a separate (encrypted) message indicating that the client portion of the handshake is finished. The server sends a message to the client informing it that future messages from the server will be encrypted with the session key. It then sends a separate (encrypted) message indicating that the server portion of the handshake is finished.
7. **Encrypted messages are exchanged :** The SSL handshake is now complete and the session begins. The client and the server use the session keys to encrypt and decrypt the data they send to each other and to validate its integrity.

Other Uses of PKI

PKI can be used for more than secure web traffic. It can also be used for:

- Secure e-mail
- Partial or whole disk encryption
- Code and driver signing
- General user authentication
- IPSec and VPN authentication
- Wireless authentication
- Network Access Control/Protection (NAC/NAP)
- Digital signatures
- And much more...

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Other Uses of PKI

In addition to uses for PKI such as e-mail encryption, Web encryption via SSL, and disk-based encryption, there are many other uses of a standard PKI. Some of the other reasons why an organization might need to implement a PKI and issue certificates are in order to implement:

- Code and driver signing
- General user Authentication
- IPSec and VPN authentication
- Wireless authentication
- Network Access Control/Protection (NAC/NAP)
- Digital signatures

Again, as was mentioned previously, an organization needs to determine what its business goals are for implementing a PKI solution. Those business drivers are important for determining what type of PKI will be required, who will manage the PKI, and what types of certificates are issued by the PKI. Regardless of what your organization is using this for today, it should be assumed that there will be even more uses for it in the future and it should be designed with flexibility and expansion in mind.

Problems with PKI

- Competing/incomplete standards
- Certification of CAs:
 - Important issue but easy to overlook
- Cross-certification between CAs
- Do-it-yourself or outsource?
- Extensive planning requirement
- User education and/or perception

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Problems with PKI

A basic question when establishing a PKI is who is going to use it. If only a small group is going to share a single CA, management can be relatively simple. Trying to establish a PKI for a large organization can be demanding. Extending it out to other organizations is even more so. As the number of people and groups who will participate increases, so does the need for tight standardization and management, but these agreements will be increasingly difficult to arrive at because there are more participants. Issues that still need to be addressed before wide-scale deployment of PKI include the following:

- Competing standards, or standards still in flux: Until most applications support a common PKI standard, interoperability will continue to hamper large PKI deployments. Before a PKI can be useful, the applications that you rely on need to be able to make use of the PKI.
- Certification of certificate authorities: The policies that a particular CA uses, and how well those policies are enforced, directly affects how secure the entire PKI based on them will be. Especially when establishing common PKIs with other organizations, common certification standards will need to be agreed to in order to make it possible to understand how trust between different groups should be maintained.
- Cross-certification between CAs: Standards for determining rules of conduct between cross-certifying CAs are still being worked out.
- Do-it-yourself or outsource it is a key question—Allowing a third-party who specializes in PKI management to run your PKI infrastructure might be cost effective, but is only possible if you completely trust the third-party.
- User education or perception: Any large deployment of software can succeed or fail based on user reaction to the system. Because a properly implemented PKI can become essential to the operation of the entire network, it is imperative that users understand and accept their role within the PKI.

- Lack of critical mass: PKIs are large systems needing careful planning and deployment to succeed. Getting enough of the components established can be challenging and the PKI will be useless until they are. This can make it difficult to justify the creation of the PKI. The high cost of establishing the PKI prior to receiving any of its benefits has cooled many organizations' interest in establishing their own PKIs.

Even with these problems, it is likely that PKIs will eventually be ubiquitous. Their advantages are too clear for them to remain on the sidelines. Many organizations are working hard to develop technical and management standards for PKI, especially the U.S. government, who is working hard to deploy a government-wide PKI. As these standards evolve and become more robust, the deployment risks will be reduced, encouraging pervasive use of PKI.

Applying Cryptography: Summary

- Cryptography can be deployed at many levels across a network:
 - Application level
 - Transport level
- Network level (VPNs, IPSec, SSL).
- Many choices but not all are compatible
- PKI is used to establish trust and is an important aspect of key distribution

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Summary

As standards related to cryptography become more prevalent, there might come a day when almost every piece of information that is processed by a computer system is protected by some form of cryptography. In the meantime, this section has provided a look at some of the current methods for cryptographic protection of our information systems. To organize our discussion, we showed how cryptography can be applied at several levels of a network including the application level, the transport level, and the network level. Each of these levels brings with it advantages and disadvantages.

At the application level, each application must provide its own cryptographic services. This allows the application developers to closely match the services to the needs of the application. The downside is that each application might need to replicate similar cryptographic services and some applications might have implemented the services better than others. Still, replacing insecure applications such as telnet with applications that support cryptography such as SSH can provide an immediate increase in security.

Applying cryptography at the transport level allows many applications to share a uniform set of cryptographic services, reducing the problem of inconsistency. Applications must still be written to use the transport level services, but because the security services themselves are not being duplicated, there is little possibility of irregularity.

If your application neither supports application level nor transport level encryption, you can still take advantage of network level cryptography. Applying encryption at the network level, referred to as virtual private networking, addresses both consistency and availability issues. Any information that flows across the network can be protected. The downside here is that individual application needs might not be taken into account.

This module also included descriptions of current protocols and products that implement cryptographic security. This included a detailed discussion of how IPSec, the current standard for implementing VPNs, can be used to protect all information that flows across an untrusted network. At the application level, we described PGP, one

of the first widely deployed public key based applications. Finally we discussed what many consider the holy grail of cryptographic protection, PKI systems, and how they can be used to establish trust, even between people who have never interacted with each other before.

Although it will be a long time before we reach a point where all of our information assets are protected at all times by cryptography, there are many current applications and protocols available that support cryptography. Using available applications and protocols such as SSH and IPSec can provide an immediate improvement in your organization's security.

Module 21:

Critical Security Controls

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Module 21: Critical Security Controls

This section intentionally left blank.

Critical Security Controls

SANS Security Essentials – © 2014 SANS

Critical Security Controls

This section intentionally left blank.

Critical Security Controls: What's the Point?

- Government and private sector organizations are being attacked and compromised daily
- What we're doing today to defend systems is mostly not working!
- Advanced vectors are proving that traditional security is not effective
- We need priorities and someone to take a stand and provide the industry with a set of real priorities for defense.
- Goals of the Critical Security Controls:
 - Those with knowledge of threats and attacks help the groups defending systems as a part of a community risk assessment model to secure systems

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Critical Security Controls: What's the Point?

Probably the best way to describe this purpose is from the authors of the guidelines themselves, who in version 3.0 of the CSC have the following to say:

"Securing our nation against cyber attacks has become one of the nation's highest priorities. To achieve this objective, networks, systems, and the operations teams that support them must vigorously defend against a variety of threats, both internal and external. Furthermore, for those attacks that are successful, defenses must be capable of detecting, thwarting, and responding to follow-on attacks on internal networks as attackers spread inside a compromised network."

"A central tenet of the US Comprehensive National Cybersecurity Initiative (CNCI) is that "offense must inform defense." In other words, knowledge of actual attacks that have compromised systems provides the essential foundation on which to construct effective defenses. The US Senate Homeland Security and Government Affairs Committee moved to make this same tenet central to the Federal Information Security Management Act in drafting the U.S. ICE Act of 2009 (the new FISMA). That new proposed legislation calls upon Federal agencies to (and on the White House to ensure that they):

"monitor, detect, analyze, protect, report, and respond against known vulnerabilities, attacks, and exploitations" and "continuously test and evaluate information security controls and techniques to ensure that they are effectively implemented."

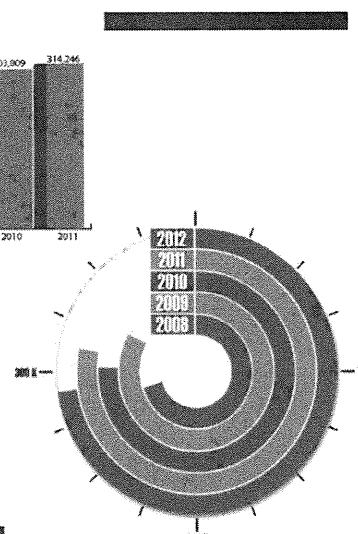
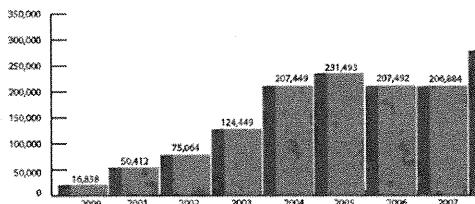
"Because federal agencies do not have unlimited money, current and past federal CIOs and CISOs have agreed that the only rational way they can hope to meet these requirements is to jointly establish a prioritized baseline of information security measures and controls that can be continuously monitored through automated mechanisms."

"This consensus document of 20 crucial controls is designed to begin the process of establishing that prioritized baseline of information security measures and controls."

The consensus effort that has produced this document has identified 20 specific technical security controls that are viewed as effective in blocking currently known high-priority attacks, as well as those attack types expected in the near future. Fifteen of these controls can be monitored, at least in part, automatically and continuously. The consensus effort has also identified a second set of five controls that are essential but that do not appear to be able to be monitored continuously or automatically with current technology and practices. Each of the 20 control areas includes multiple individual sub-controls, each specifying actions an organization can take to help improve its defenses.” (<http://www.sans.org/critical-security-controls/>)

FBI Annual Internet Crime Complaints

Yearly Comparison of Complaints¹



http://www.ic3.gov/media/annualreport/2011_IC3Report.pdf
http://www.ic3.gov/media/annualreport/2012_IC3Report.pdf

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

FBI Annual Internet Crime Complaints

Since the year 2000, the U.S. Federal Bureau of Investigations (FBI) has tracked the number of cyber complaints it has received each year. And each year, it publishes a report of its findings. The graphs in the slide represents the data sets provided. The number of cyber crime complaints has been steadily growing since the year 2000, until it started to peak in 2008-2009 at around 300,000 reported cases of cyber crime in the United States each year. Most would also agree that a good majority of these cases are never actually reported for many reasons.

For information that has been made available for the last few years, please see the full report at:

http://www.ic3.gov/media/annualreport/2011_IC3Report.pdf
http://www.ic3.gov/media/annualreport/2012_IC3Report.pdf

Reported Losses by State (2012)

Rank	State	Loss	Percent
1	California	\$68,160,064.06	16.48%
2	Florida	\$34,419,348.21	8.32%
3	Texas	\$30,445,492.21	7.36%
4	New York	\$28,108,596.87	6.80%
5	Illinois	\$14,316,107.72	3.46%
6	Pennsylvania	\$14,301,577.27	3.46%
7	Georgia	\$12,150,521.46	2.94%
8	Virginia	\$12,111,408.23	2.93%
9	New Jersey	\$11,933,510.08	2.88%
10	Washington	\$11,515,862.19	2.78%

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Reported Losses by State (2012)

In the same FBI report mentioned previously, the FBI also reports the annual losses reported by each state for the year along with the dollar value of the hard losses incurred by organizations and individuals as a result of those losses. So although there was just over 300,000 incidents reported in the year 2012, each state shouldered a different percentage of loss, leading to losses reported as high as \$70 million in California in 2012. Again, it should be remembered that these are hard losses, not soft losses, that are being reported by the FBI, and these are only those hard losses actually reported. Because we know many cases simply are not reported each year, one has to assume that the actual losses are much, much higher.

Examples from the News

- PrivacyRights.org (updated weekly)
- Here are some that are reported (most are not)
- Many APTs are not detected in a timely manner
- Just a small sample (organization/records breached):
 - Public Broadcasting Service (69,000)
 - RxAmerica and Accendo Insurance (175,000)
 - Sega (1.29 Million)
 - S. California Medical-Legal Consultants (300,000)
 - Citibank (360,000)
 - Sony Pictures (1 Million)
 - Sony Playstation Network (101.6 Million)

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Examples from the News

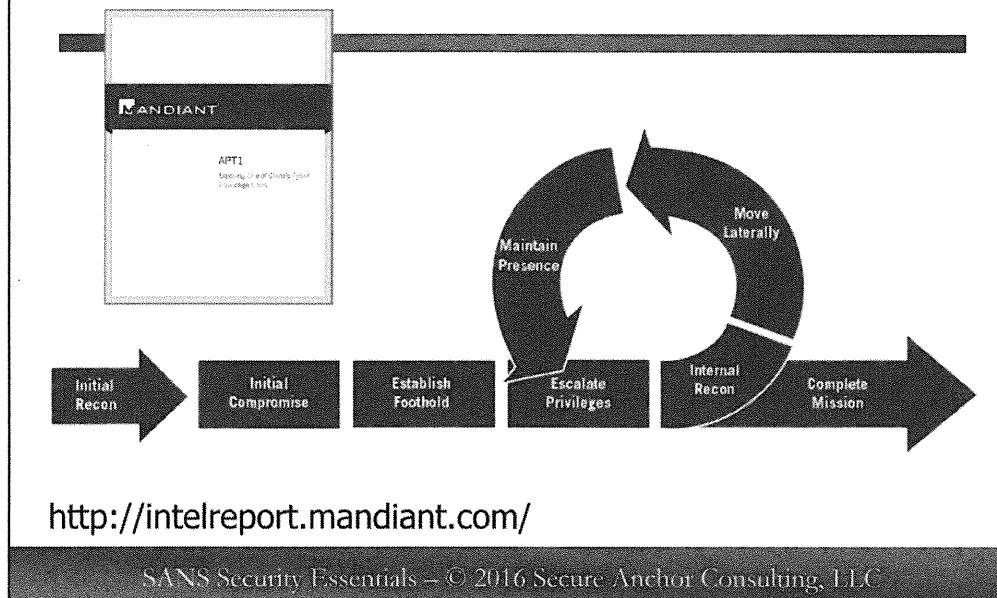
We could spend the next week of courses simply discussing examples from the news of organizations that have not followed best practices for information security and have as a result been compromised or specifically had data breaches, which have led to information being leaked to the unauthorized individuals.

One web site that I like to refer people to because it is normally very up to date as to data breaches that have occurred recently is the PrivacyRights.org web site that is updated at least weekly with new data breaches that have been reported in the news. It's important to note that some of the worst breaches, especially those of U.S. government federal systems (military and otherwise), those from the Defense Industrial Base, and from many other private sector organizations never make this list. Instead they simply go unreported and the outside world often never knows how badly a compromise has taken place. And of course all of this assumes that an organization has any idea a breach has occurred. Again, many never know what is happening on their systems.

But even from 2011, a simple visit to the PrivacyRights web site indicates the following as just a small subset of the breaches that have occurred:

- Public Broadcasting Service (69,000 – 6/2011)
- RxAmerica and Accendo Insurance (175,000 – 6/2011)
- Sega (1.29 Million – 6/2011)
- S. California Medical-Legal Consultants (300,000 – 6/2011)
- Citibank (360,000 – 6/2011)
- Sony Pictures (1 Million – 6/2011)
- Sony Playstation Network (101.6 Million – 4-6/2011)

Mandiant's Attack Lifecycle Model



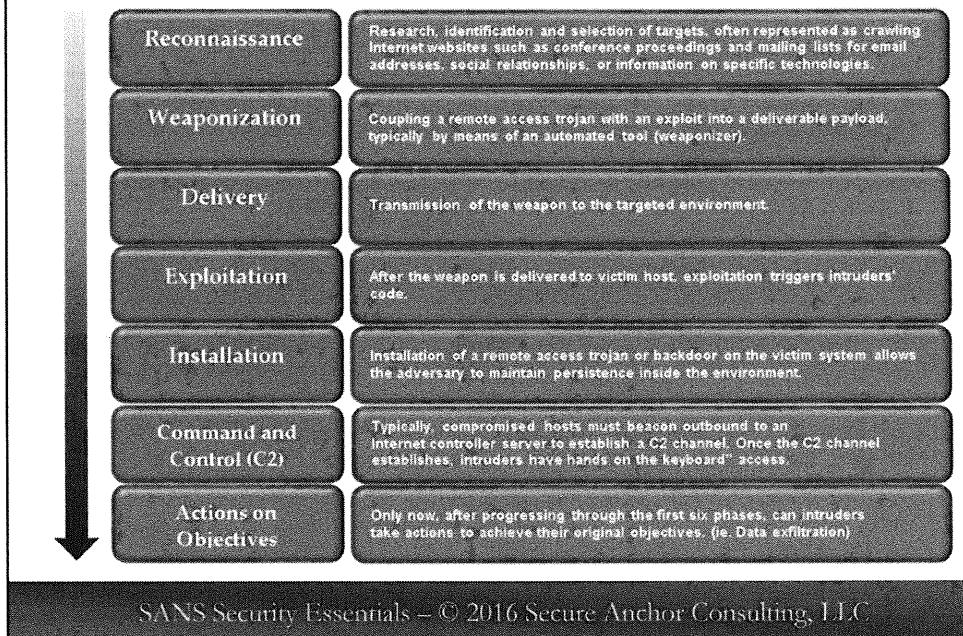
Mandiant's Attack Lifecycle Model

Many research groups have created models to help organizations better understand the attacks performed against them by bad actors. One such report is the APT1 report written by Mandiant (now FireEye), which can be found at <http://intelreport.mandiant.com/>. In this report, one of the things Mandiant describes is the average lifecycle of an advanced or dedicated attack against a copy. In this lifecycle, Mandiant has identified eight steps that are generally followed by bad actors:

1. Initial recon
2. Initial compromise
3. Establish foothold
4. Escalate privileges
5. Internal recon
6. Move laterally
7. Maintain presence
8. Complete mission

Steps 4–7, of course, often repeat themselves for a period of months or years, especially as the nature of the bad actor's mission changes over time.

Lockheed Martin Cyber Kill Chain



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Lockheed Martin Cyber Kill Chain

Another organization that has developed a lifecycle for attacks is Lockheed Martin. In a relatively well-known paper titled, “Intelligence-Driven Computer Network Defense,” Lockheed Martin describes the seven step process for how attackers compromise and attack networks. A full discussion of the paper can be found at <http://www.lockheedmartin.com/us/what-we-do/information-technology/cyber-security/cyber-kill-chain.html>.

The seven steps that Lockheed Martin identifies are:

1. Reconnaissance
2. Weaponization
3. Delivery
4. Exploitation
5. Installation
6. Command and Control (C2)
7. Actions on Objectives

Critical Security Controls Versus Intrusion Kill Chain

Critical Security Control	Actions on Objectives					
	Command & Control	Installation	Exploitation	Delivery	Weapponization	Reconnaissance
CSC #1: Inventory of Authorized and Unauthorized Devices	X		X	X	X	X
CSC #2: Inventory of Authorized and Unauthorized Software			X	X	X	
CSC #3: Secure Configurations for Hardware and Software on Mobile Devices, Laptops, Workstations, and Servers			X	X	X	
CSC #4: Continuous Vulnerability Assessment and Remediation			X	X	X	

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Critical Security Controls Versus Intrusion Kill Chain

Whether one examines the model promoted by Mandiant/FireEye or the model promoted by Lockheed Martin, the effect is the same. There appears to be a pattern of attack that cyber attackers follow when attempting to compromise a victim organization. For any control model then, the controls selected must be able to map to the attacks identified. If the offense is choosing a particular way to attack their victims, then successful defenders must understand that strategy and be able to implement defenses/controls that have the ability to directly stop the attackers' advances.

In the graphic, one can see how the Critical Security Controls map to the offensive strategies defined by Lockheed Martin's research. Clearly if an organization can defend itself, it should focus on the attack methodologies observed in the real world.

Council on CyberSecurity

- Official home of the Critical Security Controls
- CEO: Jane Lute, former Deputy Secretary of DHS
- Director of the controls: Tony Sager
- Not-for-profit group responsible for managing the controls
- Mission:

"The Council on CyberSecurity is an independent, global organization committed to an open and secure Internet."

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Council on CyberSecurity

The official home of the Critical Security Controls is the Council on CyberSecurity (<http://www.counciloncybersecurity.org/>). The CEO of this effort is Jane Lute, the former Deputy Secretary of the U.S. Department of Homeland Security. This not-for-profit group's stated mission is:

"The Council on CyberSecurity is an independent, global organization committed to an open and secure Internet. We contribute to this vision by mobilizing a broad community of stakeholders who are willing to bring their knowledge, experience, and commitment to a common goal: to identify, validate, promote, and sustain the adoption of cybersecurity best practice—by people, with technology, and through policy—to create a world in which best practice becomes common practice."

The Critical Security Controls is simply one of many cybersecurity projects managed by this council. The Critical Security Controls themselves are managed by Tony Sager, formerly of the U.S. National Security Agency, and a board of advisors and volunteers. This is the group that manages the actual documentation and updates to the controls themselves.

Document Contributors

- Blue team members inside the Department of Defense
- Blue team members who provide services for non-DoD government agencies
- Red and blue teams at the U.S. National Security Agency
- US-CERT and other non-military incident response teams
- DoD Cyber Crime Center (DC3)
- Military investigators who fight cyber crime
- The FBI and other police organizations
- U.S. Department of Energy laboratories
- U.S. Department of State
- Army Research Laboratory
- U.S. Department of Homeland Security
- DoD and private forensics experts
- Red team members in DoD
- The SANS Institute
- Civilian penetration testers
- Federal CIOs and CISOs
- Plus more than 100 other collaborators

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Document Contributors

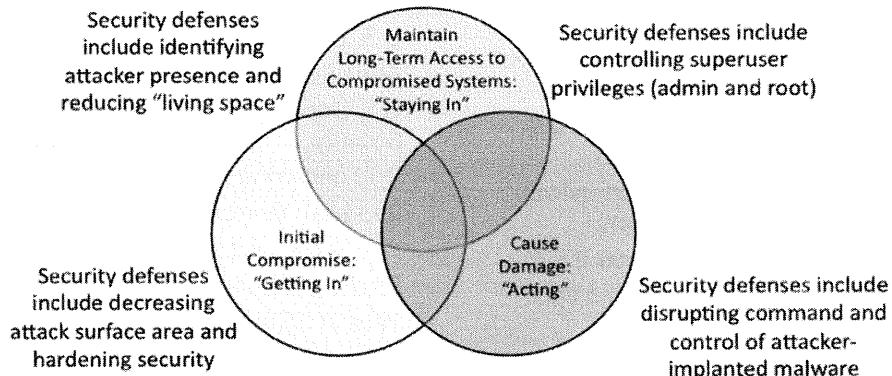
One of the best things about these controls is that they were not created in a vacuum. Instead they are the result of numerous entities working together to provide feedback into the attacks that they are seeing and the controls that they've found, which are helpful in truly combating these threats. The individuals that contributed to the project have experience in fighting the threats directly and have given insight from their experiences.

Some of the groups that helped participate and contributed to this project are:

- Blue team members inside the Department of Defense
- Blue team members who provide services for non-DoD government agencies
- Red and blue teams at the U.S. National Security Agency
- US-CERT and other non-military incident response teams
- DoD Cyber Crime Center (DC3)
- Military investigators who fight cyber crime
- The FBI and other police organizations
- U.S. Department of Energy laboratories
- U.S. Department of State
- Army Research Laboratory
- U.S. Department of Homeland Security
- DoD and private forensics experts
- Red team members in DoD
- The SANS Institute
- Civilian penetration testers
- Federal CIOs and CISOs
- Plus more than 100 other collaborators

Types of Computer Attacker Activities These Controls are Designed to Help Thwart

Computer Attacker Activities and Associated Defenses



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Types of Computer Attacker Activities These Controls are Designed to Help Thwart

These controls are not limited to blocking only the initial compromise of systems but also address detecting already compromised machines and preventing or disrupting attacker's actions. The defenses identified through these controls deal with decreasing the initial attack surface through hardening security, identifying already compromised machines to address long-term threats inside an organization's network, controlling super-user privileges on systems, and disrupting attackers' command-and-control of implanted malicious code. This slide illustrates the scope of different kinds of attacker activities these controls are designed to help thwart.

The rings represent the actions computer attackers often take against target machines. These actions include initially compromising a machine to establish a foothold by exploiting one or more vulnerabilities. Attackers can then maintain long-term access on a system, often by creating accounts, subverting existing accounts, or altering the software on the machine to include backdoors and rootkits. Attackers with access to machines can also cause damage, which include stealing, altering, or destroying information, impairing the system's functionality to jeopardize its business effectiveness or mission, or using it as a jumping-off point for compromise of other systems in the environment. Where these rings overlap, attackers have even more ability to compromise sensitive information or cause damage. Outside of each set of rings in the figure, various defensive strategies are presented, which are covered throughout the controls described in this document. Defenses in any of the rings helps to limit the abilities of attackers, but improved defenses are required across all three rings and their intersections. It is important to note that the CSC is designed to help improve defenses across each of these rings, rather than merely preventing initial compromise.

Project Guiding Principles (1)

- Defenses should focus on addressing the most common and damaging attack activities occurring today, and those anticipated in the near future
- Enterprise environments must ensure consistent controls across an enterprise to effectively negate attacks



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Project Guiding Principles (1)

In the course of developing the Critical Security Controls, there were a number of guiding principles that were also developed to guide the project and give direction to the writing and implementation of each of the controls. A couple of these guiding principles are:

Defenses should focus on addressing the most common and damaging attack activities occurring today and those anticipated in the near future. In other words, the defensive mechanisms that are implemented today should be based on actual attacks that have been seen in the field during incident response activities. They should not be academic, but real attacks that have been seen and the controls that are developed should be based on tactics that can stop these attacks from being successful.

Enterprise environments must ensure consistent controls across an enterprise to effectively negate attacks. In addition to implementing these controls, they need to be implemented in a consistent manner across the enterprise. If they are not implemented consistently across the enterprise, then the organization is opening the door for risk. It is the equivalent to building a house and putting locks on “almost” all of the exterior doors. The thought was good, but the home is still left partially unprotected.

Project Guiding Principles (2)



- Defenses should be automated where possible, and periodically or continuously measured using automated measurement techniques where feasible
- To address current attacks occurring on a frequent basis against numerous organizations, a variety of specific technical activities should be undertaken to produce a more consistent defense

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Project Guiding Principles (2)

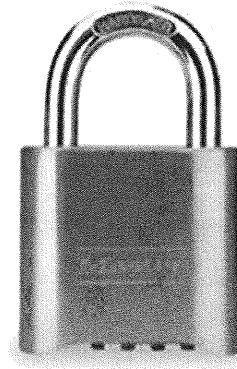
In addition to the two principles mentioned on the previous slide, the following two principles were also developed for the same purpose:

Defenses should be automated where possible and periodically or continuously measured using automated measurement techniques where feasible. Although having controls is good, having controls that can be automated is even better. If an administrator has to constantly or manually work with the controls, then the likelihood of the controls being successful goes down. By adding the human resource element into the equation, it makes the likelihood of the control being circumvented higher due to neglect, mistakes, or simply lack of organizational resources to follow through on the control.

To address current attacks occurring on a frequent basis against numerous organizations, a variety of specific technical activities should be undertaken to produce a more consistent defense. There is certainly nothing wrong with operational controls. They bring value to an organization and can help the organization to better defend its systems. However, they are not the focus of the Critical Security Controls. These are a set of technical controls that can help defend systems. There are other models that focus on process and operational tactics; this is not one of them.

Project Guiding Principles (3)

- Root cause problems must be fixed to ensure the prevention or timely detection of attacks
- Metrics should be established that facilitate common ground for measuring the effectiveness of security measures, providing a common language to communicate about risk



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Project Guiding Principles (3)

In addition to the two principles mentioned on the previous slides, the following two principles were also developed for the same purpose:

Root cause problems must be fixed to ensure the prevention or timely detection of attacks. We are not simply attempting to address surface issues with these controls. We are trying to get to the heart of the issue. More money or more personnel are not always the solution to this problem. There might be other underlying causes that need to be addressed before we start to see success.

Metrics should be established that facilitate common ground for measuring the effectiveness of security measures, providing a common language to communicate about risk. This is an area of maturity that many organizations are just starting to consider. Many organizations have heard the call for developing metrics, but actually developing information assurance related metrics might be well into the future. Organizations need to determine whether they follow the Critical Security Controls or not, which set of metrics they plan on following to measure their effectiveness. SANS certainly has their preferences, but a model needs to be considered.

Understanding the Controls

- The controls are prioritized based on the NSA attack mitigation scores
- A few controls cannot be automated and labeled with (validated manually)
- Key rules when the controls were chosen:
 - Each control has to map to a actual known attack
 - If a known attack does not exist, it cannot be a control
 - “Offense must inform defense”

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Understanding the Controls

In addition to everything else that's been stated so far about the controls, there are a few other principles and details that everyone should be aware of when implementing or evaluating based on these controls.

First, the listing of the controls are based on the NSA attack mitigation scores.

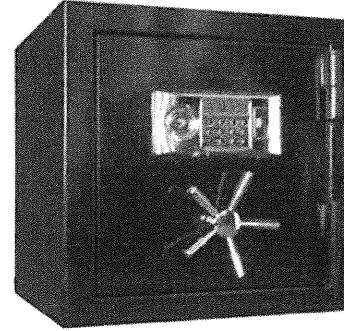
Note that there is a distinction between some controls. Most of the controls were meant to be automated. These defenses should be automated defenses that do not require constant attention on the part of system administrators. Although any system requires care and feeding, these defensive systems should not require manual efforts on the part of administrators in order to function.

And, as has been said before, there are a few key rules that were established when creating these controls:

1. Each control has to map to an actual known attack.
2. If a known attack does not exist, it cannot be a control.
3. “Offense must inform defense.”

Categories of Sub-Controls

- Quick Wins (QW)
- Improved Visibility and Attribution (Vis/Attrib)
- Hardened Configuration and Improved Information Security Hygiene (Config/Hygiene)
- Advanced (Adv)



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Categories of Sub-Controls

In the Critical Security Controls, there are four categories of sub-controls used to describe the type of sub-control that is specified as a part of the overall control itself. Remember, each of the Critical Security Controls is further broken down into sub-controls or tasks that need to be accomplished to meet the overall goals of the control. Each of these sub-items or tasks is categorized into one of these four areas.

Because the critical control documentation itself is probably the best source for how these are defined, let's hear what that documentation has to say. In the documentation, it describes each category in the following way:

“Quick Wins: These fundamental aspects of information security can help an organization rapidly improve its security stance, generally without major procedural, architectural, or technical changes to its environment. It should be noted, however, that a Quick Win does not necessarily mean that these sub controls provide comprehensive protection against the most critical attacks. The intent of identifying *Quick Win* areas is to highlight where security can be improved rapidly. These items are identified in this document with the label of QW.

“Improved Visibility and Attribution: These sub controls focus on improving the process, architecture, and technical capabilities of organizations so that organizations can monitor their networks and computer systems, gaining better visibility into the IT operations. Attribution is associated with determining which computer systems, and potentially which users, are generating specific events. Such improved visibility and attribution support organizations in detecting attack attempts, locating the points of entry for successful attacks, identifying already-compromised machines, interrupting infiltrated attackers' activities, and gaining information about the sources of an attack. In other words, these controls help to increase an organization's situational awareness of its environment. These items are labeled as Vis/Attrib.

"Hardened Configuration and Improved Information Security Hygiene: These aspects of various controls are designed to improve the information security stance of an organization by reducing the number and magnitude of potential security vulnerabilities as well as improving the operations of networked computer systems. This type of control focuses on protecting against poor security practices by system administrators and end users that could give an adversary an advantage in attacking target systems. Control guidelines in this category are formulated with the understanding that a well managed network is typically a much harder target for computer attackers to exploit. Throughout this document, these items are labeled as Config/Hygiene.

"Advanced: These items are designed to further improve the security of an organization beyond the other three categories. Organizations already following all of the other controls should focus on this category. Items in this category are simply called Advanced.

The Critical Security Controls

- | | |
|--|--|
| 1. Inventory of authorized and unauthorized devices | 11. Limitation and Control of Network Ports, Protocols, and Services |
| 2. Inventory of authorized and unauthorized software | 12. Controlled Use of Administrative Privileges |
| 3. Secure configurations for hardware and software on laptops, workstations, and servers | 13. Boundary Defense |
| 4. Continuous Vulnerability Assessment and Remediation | 14. Maintenance, Monitoring, and Analysis of Audit Logs |
| 5. Malware Defenses | 15. Controlled Access Based On Need to Know |
| 6. Application Software Security | 16. Account Monitoring and Control |
| 7. Wireless Device Control | 17. Data Protection |
| 8. Data Recovery Capability (validated manually) | 18. Incident Response Capability (validated manually) |
| 9. Security Skills Assessment and Appropriate Training To Fill Gaps (validated manually) | 19. Secure Network Engineering (validated manually) |
| 10. Secure configurations for network devices such as firewalls, routers, and switches | 20. Penetration Tests and Red Team Exercises (validated manually) |

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

The Critical Security Controls

So what actually are the Critical Security Controls? We spend the remainder of this course evaluating these controls, discussing how to implement these controls, and learning how to evaluate an organization based on these controls. However, to get started, let's at least take a brief look at each of these to get an overview of where we are going this week.

The controls are:

1. Inventory of authorized and unauthorized devices
2. Inventory of authorized and unauthorized software
3. Secure configurations for hardware and software on laptops, workstations, and servers
4. Continuous Vulnerability Assessment and Remediation
5. Malware Defenses
6. Application Software Security
7. Wireless Device Control
8. Data Recovery Capability (validated manually)
9. Security Skills Assessment and Appropriate Training to Fill Gaps (validated manually)
10. Secure configurations for network devices such as firewalls, routers, and switches
11. Limitation and Control of Network Ports, Protocols, and Services
12. Controlled Use of Administrative Privileges
13. Boundary Defense
14. Maintenance, Monitoring, and Analysis of Audit Logs
15. Controlled Access Based On Need to Know
16. Account Monitoring and Control
17. Data Protection
18. Incident Response Capability (validated manually)
19. Secure Network Engineering (validated manually)
20. Penetration Tests and Red Team Exercises (validated manually)

Why are the Controls Important? (1)

- Cyber security is complex and becoming even more complicated every day
- Organizations are being compromised, even after spending large portions of their budget on infosec
- CIOs and CISOs need prioritized controls to get the most return from their investment
- More controls rarely hurt, but how do we decide which controls to start with?
- **It's critical that we have priorities!**

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Why are the Controls Important? (1)

The reality is that everyday defending information systems is becoming more and more complex as technologies expand and new features and systems are being added to our networks at a growing rate. And it seems like no matter how much money is spent on cyber-defense, each year the problem keeps getting worse. Even for those organizations that are spending a larger and larger portion of their budgets on controls to protect their information systems, it seems like it is a losing battle against those that would try to compromise their data.

As a result, it seems like there is a real need, which has been identified by CIOs and CISOs—they need a prioritized set of controls and associated metrics that will give them the biggest return on their security investment and give them a fighting chance at defending their systems. The reality of the issue is that rarely do more controls hurt the situation, but there are so many possible controls these days that it is difficult to know which controls make the most sense and will return the biggest gain for the investment. How do we know which controls to start with when defending our systems?

It is absolutely critical that we have priorities!

Why are the Controls Important? (2)

- We need agreement among:
 - Inspectors General (IGs – auditors)
 - Operations (sys-admins)
 - Security engineers
- We need metrics and measurements that everyone can agree to use
- We need to stop people from violating systems and compromising the confidentiality and integrity of our data

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Why are the Controls Important? (2)

We also need to consider that there needs to be agreement between the people who are implementing the controls and the people who are auditing for the controls. For years, system administrators have implemented one set of controls thinking that was the right thing to do, and auditors have evaluated the organization based on another set of controls. The worst part of this issue is that the controls both groups were working with were not necessarily the right controls, and more often than not these controls were not in sync. We need to have metrics and measurements that both groups agree, based on business goals, are the right controls for the organization.

The bottom line is that organizations need to stop unauthorized individuals from violating systems and compromising the confidentiality, integrity, and availability of the data sets that reside on these systems. Unless we are all in agreement as to what the necessary controls are, it will be very difficult for us to protect our systems and have any level of consistency in our approach to defense.

Critical Security Controls Versus NIST

- These controls are NOT meant to supersede NIST guidelines (800-53 or others)
- The Critical Security Controls are a subset of the Priority 1 items in NIST 800-53
- The Critical Security Controls are technical only in nature, they do not address personnel and physical security controls.
- These controls are a technical “high water mark”
- Implementation and auditing priorities should be:
 - Implement and audit for the Critical Security Controls first
 - Implement and audit for the remaining 800-53 controls next

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Critical Security Controls Versus NIST

One question that has oftentimes been raised with the Critical Security Controls themselves is, “How do these controls compare to the controls listed in NIST 800-53?”

First of all, it has to be said that these controls are not meant to supersede the guidelines in NIST 800-53. In fact, if you compare the Critical Security Controls to the Priority 1 items that are listed in the NIST 800-53 guidance, you’ll notice that the Critical Security Controls are simply a subset of this list. These controls, however, are only technical controls, not operational controls, and they’re meant to be priorities that an organization can follow when getting started defending their systems. They’re meant to be a technical “high water mark” of controls that absolutely must be in place if we are to defend our systems.

Therefore, if you are an organization that is already used to implementing the controls in NIST 800-53 (or even if you are not), the controls listed in the Critical Security Controls should be seen as a starting point for implementing controls and measuring your organization to see whether you are in line with them. Implement these controls first and then move on to the remaining controls in the NIST 800-53 guidance. The people at NIST have done a great job of identifying important controls that should be in place. We are in no way recommending that these controls be ignored. We’re simply saying that if you have to start with a subset of controls, those listed in the Critical Security Controls are a good place to start.

Critical Controls Versus Other Standards

- NIST SP 800-53rev3 describes a comprehensive security program, covering almost all aspects of information security a typical enterprise faces
- The same theory of mapping the Critical Security Controls to other regulations could be applied
- The goal is the same, regardless of the standard, to prioritize technical controls for protecting systems
- These controls are just as applicable to both public and private sector organizations
- **The focus of the controls is assurance, not compliance!**

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Critical Security Controls Versus Other Compliance Regulations

When talking to U.S. Federal Government CIOs and CISOs, everyone determined that NIST 800-53 was their baseline for technical controls in their information systems. In fact, this was such a vital concern that in the Critical Security Controls documentation, there is even a mapping between the NIST 800-53 documentation and the Critical Security Controls themselves. But what if you aren't in the U.S. Federal Government?

Regardless of the industry where you find yourself, there is likely some industry or government regulation that you are responsible for implementing. Although there is not a direct mapping that's been made between the Critical Security Controls and other compliance regulations, motivated individuals certainly could come up with a mapping between the controls and other regulations. Regardless of the regulations, the goal is still the same—to have prioritized controls. These priorities are vital in being successful at implementing the right controls to protect information systems.

The one thing everyone needs to remember, though, is that the focus of these controls is assurance, not compliance! Just because you're able to create a mapping, it does not mean that the goals of both lists of controls is the same. The purpose of these controls is information assurance and protecting data, not simply creating another checklist to be used for compliance purposes.

Revision History

- Version 1.0: Original rough draft of controls
- Version 2.0: Major revision of sub-controls based on community and agency feedback
- Version 2.1: Minor revision of sub-controls based on community and agency feedback
- Version 2.3: Addition of metrics and core evaluation methodologies
- Version 3.0: Major revision of sub-controls and addition of standards mappings and sensors
- Version 3.1: Reordering of controls based on priority of controls
- Version 4.0: Revision of sub-controls, removal of sensors, addition of ERDs
- Version 4.1: Minor edits and further alignment with Aus DSD Top 4
- Version 5.0: Updates to sub-controls and cleaning controls

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Revision History

In each version of the controls, there has been something new added or clarified. Each time a version is released, it is given to the community for their feedback. Each version has a story and new contributors often have given feedback based on the attacks that they have seen at their organization.

To this point, the versions of the controls that have been released are:

Version 1.0 – Original rough draft of controls

Version 2.0 – Major revision of sub-controls based on community and agency feedback

Version 2.1 – Minor revision of sub-controls based on community and agency feedback

Version 2.3 – Addition of metrics and core evaluation methodologies

Version 3.0 – Major revision of sub-controls and addition of standards mappings and sensors

Version 3.1 – Reordering of controls based on priority of controls

Version 4.0 – Revision of sub-controls, removal of sensors, and addition of ERDs

Version 4.1 – Minor edits and further alignment with Aus DSD Top 4

Version 5.0 – Updates to sub-controls and cleaning controls

Module 22:

IT Risk Management

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Module 22: IT Risk Management

This section intentionally left blank.

IT Risk Management

SANS Security Essentials III: Internet Security Technologies

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Introduction: IT Risk Management

This module focuses on risk management: The art of analyzing threats and vulnerabilities, and determining the impact these risks can have on your enterprise. Risk management is much more than just determining the various risks you are exposed to. It is an exploration of the various approaches and techniques for managing these risks.

You might be asking yourself: Why is risk management so important? It is because every computer hardware or software implementation has some security risk associated with its use. Take for instance the situation where your company wants to implement a wireless LAN architecture to co-exist with the wired network. There are documented (and some undocumented) risks associated with wireless LAN (WLAN) technology. Do you just ignore these risks and implement WLAN without any worries? This is where risk management techniques are used to determine the level of risk, and if we can live with that risk level.

Risk management's main focus is to reduce the risk until it is at an acceptable level. The actual acceptable level will vary from company to company. However, risk management means that we need to identify, control, and minimize the loss associated with each risk. We begin by understanding the risk management process, the concepts of threats and vulnerabilities, and their relationship to risk assessments.

Objectives

- Risk management overview
- Best practice approach to risk management
- Threat assessment, analysis, and report to management

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Objectives

Risk management involves an understanding of how security is implemented in your organization, and how security threats affect your business operations. As a general rule, before you can begin managing risks, you need to understand your business operations and the types of risks that they might be exposed to.

Why is risk management so important to an organization? The fact is there are risks all around us. Some risks are not that damaging, although some can cause catastrophic results. The question is whether you know what those risks are. More important, what will you do if they become real?

Take the example of someone living in a three million dollar home on the beach in Malibu, California. Every few years, a big storm hits the West Coast and causes the ocean to generate some pretty awesome waves that hit the coast. Every year, many homes are destroyed by the storm. You might ask yourself why anyone would choose to live in an area where their house will most likely be destroyed by the storms. Most likely the owners have done some form of risk assessment and risk management. That is, they have determined the risks associated with having a house on the beach (the storms), analyzed the impacts of these risks (house could be destroyed), and determined a course of action on how to handle these risks (purchase insurance). In this example, we have addressed the cause, the effect, and the response to a risk condition.

As you might imagine, every industry has its share of operational risks. The information technology field is the same. Any computer or system on the Internet or another network is vulnerable to an attack. Having a system on the Internet is like taking a martial arts class—you are going to get hit. The questions you need to ask yourself are: How hard are you going to get hit? What is the damage if I do get hit? What can I do to minimize the damage? Remember, in risk management, we are concerned with the cause, the effect, and our response to the risk incident.

In this chapter, we structure our definitions and assumptions about risks around the concepts of the information security triad: confidentiality, integrity, and availability. We should keep these concepts in mind when performing risk assessments and subsequent risk management decisions. In risk management, we are looking at ways to minimize the impact that could affect the confidentiality of our information, the integrity of our systems and data, and the availability of our infrastructure.

Risk management helps information systems (IS) management strike a balance between the impact of risks and the cost of protective measures. The goal of risk management is to identify, measure, control, and minimize or eliminate the likelihood of an attack.

Risk Management Overview

The student will understand the terminology and basic approaches to risk management.

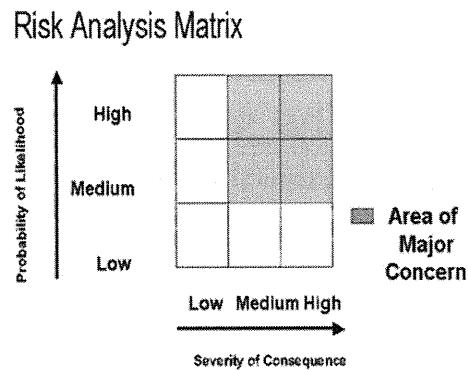
SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Risk Management Overview

This section intentionally left blank.

IT Risk Management: Where Do I Start?

- Identify threats and vulnerabilities, and analyze risks
- Validate due care by using industry best practices



SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

IT Risk Management : Where Do I Start?

The objective of risk management is to identify specific areas where safeguards (or countermeasures) are needed to prevent deliberate or inadvertent unauthorized disclosure or modification of information.

The steps for an effective risk management process are:

- Conduct a rapid assessment of risks so you know what your security policy needs to cover. This forms the basis for your security policy, with input from various business departments.
- Fully analyze risks, or identify industry practice for due care; analyze vulnerabilities.
- Set up a security infrastructure.
- Design controls; write standards for each technology.
- Decide which resources are available, prioritize countermeasures, and implement the top priority countermeasures you can afford.
- Conduct periodic reviews and possibly tests.
- Implement intrusion prevention and incident response.

We need to start with policy because it will dictate the security posture the company wants to take with respect to protecting its resources.

If you have a very open security policy (for example, you allow anything in and anything out of your corporate network), and you are concerned about the risks to your network, then your desired policy does not match your implementation. The security policy will point you to areas of your business operation that require protection. It is not possible to implement 100% protection for your enterprise. The best approach is to concentrate first on protecting those areas of your organization that if compromised, could incur the most damage.

The second step in risk management is to analyze risks and determine their impact to your organization. This also involves looking at the industry's best practice for maintaining security.

As we stated previously, risk analysis involves determining the risks and determining their impact to the infrastructure. The figure in the slide is a risk analysis matrix. The X-axis is the severity of consequence, rated from low to high. That is, as the risks or the degree of severity increases, so does the damage it does. The Y-axis is the probability of likelihood that the risk could really happen, also rated from low to high. The goal is to concentrate on those areas that result in a medium-to-high severity of consequence and a medium-to-high likelihood that it would actually occur. For example, the severity of consequence of a huge meteor hitting the earth is high, but the probability of likelihood is low. This scenario would not be an area of concern. However, putting our e-commerce on the Internet and not protecting it with a firewall could result in a high probability that the system would be compromised and a high severity of consequence.

Define Risk

- $\text{Risk} = \text{Vulnerability} \times \text{Threat}$
- **Vulnerability:** A weakness in a system that can be exploited
- **Threat:** Any event that can cause an undesirable outcome

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Define Risk

Now that we have discussed in detail the risk management process, we should define the concepts of threats and vulnerabilities, and how they relate to risk analysis. Risk identification involves understanding the associated threats and vulnerabilities you might be exposed to.

What is the definition of risk?

The classical definition of risk is:

Risk = Threat x Vulnerability

Vulnerability is defined as a weakness in a system that could be exploited. You have heard this one before, "A vulnerability has been found in the XYZ FTP client service that if exploited, could result in a buffer overflow," or something like that. The vulnerability is the fact that the FTP client has a flaw, a weakness that could lead to a system compromise. The danger lies in the fact that these hidden vulnerabilities are discovered and subsequently exploited.

A threat is any event that can cause an undesirable outcome. A threat could be the exploitation of a vulnerability. The threat is that someone could actually exploit this weakness and compromise your system.

Risk Management Questions

(Risk Requires Uncertainty)

- What could happen? (What is the threat?)
- If it happened, how bad could it be? (Impact of threat)
- How often could it happen? (Frequency of threat—annualized)
- How reliable are the answers to these three questions? (Recognition of uncertainty)

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Risk Management Questions

To decide among accepting, mitigating, or transferring the risk, we need to better understand the risk and how it affects us.

When evaluating risk, it is helpful to ask yourself some key questions:

1. What could happen?
2. If it happened, how bad could it be?
3. How often could it happen?
4. How reliable are the answers to the previous questions?

The answers to these questions help us focus on the actual threats and gain a better understanding of their impact if they were to actually happen. The first question is to ask ourselves: What exactly are we afraid of? What is the actual threat? Is the threat something tangible? Can we accurately define the threat?

And if we can define the threat, what damage could it cause? What is the probable extent of the damage? For instance, the damage could be anything from a few corrupted files to a complete deletion of all critical files. In other words, what is the impact of the threat? Another variable to consider is the frequency of the threat. How often could this threat happen? Is it just once, or can it occur more often?

The last question relates to the recognition of uncertainty. That is, how sure are you of the answers to the three questions? Can you validate and prove your answers? This might be a difficult question to answer, because it might be hard to accurately perform our risk calculations on operating systems or new programs when new vulnerabilities are constantly being discovered.

SLE Versus ALE

- Single Loss Expectancy (SLE):
The loss from a single event
- Annualized Loss Expectancy (ALE):
Annual expected loss based on a threat

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

SLE Versus ALE

When all is said and done, in the end, it all comes down to money. What management will be considering is, "How much financial loss are we willing to accept in a single (threat) event?" If a company's database is compromised and that database contains your proprietary (and valuable) secret formula for your next revolutionary drug, then you could not afford even one risk to your system that might lead to the theft of this formula. Remember we stated that risk involves uncertainty. The uncertainty here is that we cannot accurately determine the exact value of the formula (it might make millions of dollars, or it might not make any money at all because the formula might not work).

Single Loss Expectancy (SLE—One Shot)

- Asset value x exposure factor = SLE
- Exposure factor: 0 – 100% of loss to asset
- Small focused 509c conference, one event/yr
- Terrorist event causes 50% drop in turnout
- Revenue \$100 k
- $\$100,000 \times .5 = \$50,000$ loss expectancy

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Single Loss Expectancy

What this leads to is the calculation of the Single Loss Expectancy (SLE). The SLE is the dollar value that is assigned to a single event. That is, it is the organization's loss from a single event. The formula is:

Single Loss Expectancy = Asset Value (\$) X Exposure Factor (EF)

The Exposure Factor (EF) is the percentage of loss a threat event would have on the asset. The EF is expressed in terms of 0 to 100% loss to an asset. In the conference business, you have to guarantee a certain number of hotel rooms, whether or not people show up to stay in them. Event planners use classic risk management techniques to help them understand their overall risk. As we all learned from 9/11, there was a major downturn in travel. This significantly impacted a number of small-focused conferences because people just did not show up. For example, if terrorist event cause a 50% drop in turnout to a small-focused 509c conference, which had a value of \$100,000, the single loss expectancy would be \$50,000, because we can assume that a terrorist event would result in a 50% loss.

Annualized Loss Expectancy (ALE: Multi-Hits)

- SLE x Annualized rate occurrence = Annual Loss Expectancy (ALE)
- Annual loss: The frequency the threat is expected to occur
- Example, web surfing on the job:
 - SLE: 1000 employees, 25% waste an hour per week surfing, $\$50/\text{hr} \times 250 = \$12,500$
 - ALE: They do it every week except when on vacation: $\$12,500 \times 50 = \$625,000$

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Annualized Loss Expectancy

What happens when the event occurs more than once? We then calculate the Annualized Loss Expectancy (ALE). The ALE is the annual expected financial loss from a threat. The formula is:

Annual Loss Expectancy = Single Loss Expectancy x Annualized Rate of Occurrence (ARO)

The Annualized Rate of Occurrence (ARO) is the estimated frequency at which a threat is expected to occur. Its value can range from zero to a large number. Sometimes the ARO is easy to calculate. Other times, it is very difficult to compute; in fact, many times this number represents the uncertainty factor in our risk management calculation.

As a real-case scenario, imagine you need to calculate the amount of revenue loss because of your employees' web surfing during work hours (not work-related, of course). We start by calculating the SLE. For this, we need the asset value and the exposure factor.

If 25% of your 1,000 employees waste one hour of their time each week surfing the web and the cost per hour is \$50, then the formula becomes:

SLE = $\$50/\text{hr} \times 250$ or \$12,500 per week

That cost is significant.

If we want to calculate the annualized cost, the formula becomes:

ALE = $\$12,500 \times 50$ weeks (assuming a two-week vacation) or \$650,000 per year.

Quantitative Versus Qualitative

Quantitative Risk Assessment

- Far more valuable as a business decision tool because it works in metrics, usually dollars

Qualitative Risk Assessment

- Asset value and safeguard cost can be tied to monetary value, but not the rest of the model
- Easier to calculate but results are more subjective
- Results typically categorized as low-, medium-, or high-risk events
- Succeeds at identifying high-risk areas

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Quantitative Versus Qualitative

There are two risk assessment approaches: qualitative and quantitative. In quantitative risk assessment, we try to assign an objective numeric value; typically this value represents a monetary loss value. Qualitative risk assessment, on the other hand, deals with more intangible values, and focuses on variables and not just the monetary losses.

Quantitative risk assessment is a far more valuable business tool, because it works on metrics—usually in dollars. And the bottom-line cost in dollars is what management is looking for when trying to understand the implications of how a risk can affect the organization.

Qualitative risk assessment is much easier to perform and can identify high-risk areas. For instance, you need to perform a risk assessment to determine the impact of installing a wireless LAN access point in your organization. The first order of business is to determine the vulnerabilities, threats, and therefore the risks of using a wireless LAN. Then, you determine whether those risks apply to your organization and determine the likelihood that you are at risk. One of the risks of using a wireless LAN is the possibility of someone sniffing the wireless network traffic, and that a misconfigured access point can allow rogue client connections. These are real risks that need to be addressed. Can you put a monetary value to these risks? If someone does connect to your network via the open access point, how much is that going to cost your company in lost revenue?

As you can see from this example, quantitative risk analysis in this situation does not quite work. A qualitative approach is much better, because we can arrive at a more subjective result. In qualitative risk assessment, the results are typically categorized as low-, medium-, or high-risk events. A person operating a wireless LAN access point in the house in the country, where the nearest neighbor is 5 miles away, is at a low risk of having someone trying to connect to the network. A company in the middle of a high-tech park, with an access point that allows rogue connections, has a high risk.

Threat Assessment, Analysis, and Report to Management

The student will be able to identify each step in the Threat Assessment and Analysis process and how to report findings to management.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Threat Assessment, Analysis & Report to Management

This section intentionally left blank.

Business Case for Risk Management

- Use qualitative, quantitative, or best practice/checklist risk measurement to define the gap between our current risk status and where we want to be
- After the gap analysis, we select safeguards such as:
 - Host-Based Intrusion Prevention with operating system shims
 - NIPS/Antivirus solutions
 - Application-aware firewalls

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Business Case for Risk Management

It all eventually comes down to making the presentation to management and the need to convey the big picture. It is not enough to understand the core technologies we use for our countermeasure controls: host- and network-based intrusion prevention systems, vulnerability scanners, honeypots, and firewalls. The question is, can you show them how these technologies work together to produce the results needed?

Every enterprise has different needs and diverse expectations. A financial institution has different priorities than a military organization. A pharmaceutical company's valuable assets could be the formula for a new drug. A financial institution's assets could be client lists and account numbers. Everyone has something different to protect and a different tolerance for risk. It is interesting to note that banks lose large amounts of money every year and cannot account for how it disappears. To a bank, losing \$1 million per year might not be a big issue. For it, this is an acceptable and tolerable risk level—the cost of doing business.

Business Case: Applications

- Business case should always map back to risk:
 - Organization has no intrusion prevention and you are presenting the case for standing up a capability
 - Organization has rudimentary capability and you want to upgrade
 - Organization has central monitoring and you are presenting the case for a departmental capability
- If you cannot provide proof that systems are at risk, it will be harder to get additional funds for the countermeasures you recommend

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Business Case: Applications

Now that we have introduced the basic risk assessment process, let's apply this process to the business case for an intrusion prevention system.

First, let's consider the different scenarios we might be working with:

- The organization has no intrusion prevention, and you are presenting the case for adding this capability.
- The organization has a rudimentary intrusion prevention system, and you might recommend upgrading the system.
- The organization has a central intrusion prevention monitoring system, and you are presenting the case for departmental capability.

One of the problems you might face is that many managers are uncomfortable when confronted with actual data about attacks and vulnerabilities. They might clearly see this as a weakness on their part to do their job. Even as an outside consultant, you might face the same roadblock. In fact, as a consultant, you might feel a lot of resistance, even from the system administrators. This is because they might feel that you will show management that they have not been performing their jobs adequately.

It could also be the case that managers just don't understand the severity of the situation. They might not really believe that there is a problem. If you cannot provide proof that their systems are at risk, it will be harder to convince them to spend additional funds for the countermeasures you will recommend. You can often use existing sources of data, such as firewall and system logs, to leverage additional intrusion prevention financing by showing them a "smoking gun."

Step 1: Threat Assessment and Analysis

- Identify the types of threats
- Look for evidence that these threats are actually in use and remember the threat vectors:
 - Outsider attack from network
 - Outsider attack from telephone
 - Insider attack from local network
 - Insider attack from local system
 - Attack from malicious code

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Step 1: Threat Assessment and Analysis

Any system connected to the Internet is vulnerable to possible hacker and worm compromise. These attacks can come from many sources. If we do not install any security protection on our systems, how much system compromise (remember to think in terms of confidentiality, integrity, and availability) can we withstand? If we have valuable assets on our information systems to protect (and we all do), then what countermeasures should we install that will protect us from outside attacks?

The threat of a destructive worm is currently one of the most likely negative events and potentially most catastrophic. Though there are thousands of threats, we will use that specific one to illustrate the risk management process.

When determining what types of threats your enterprise could be exposed to, it is vital that information security professionals spend time thinking about how they might be attacked. That is, you need to enumerate all possible threats you might have to deal with. After the list is compiled, you can then look for evidence that these threats are actually viable threats to your enterprise. For example, your initial list of threats includes destruction of the data center by an earthquake. After investigating further, you determine that earthquakes are not prevalent in that area and the likelihood of an earthquake is nonexistent.

The process of determining what is at risk and what is the impact if the identified threats materialize is known as “risk analysis.”

The purpose of risk analysis is to:

- Identify existing countermeasures, threats, and vulnerabilities.
- Support the expenditure of resources and to determine the most cost-effective safeguards to offset the risks.
- Aid in the selection of cost-effective countermeasures that will reduce existing risks to an acceptable level.

The best way to focus on the real threats is to focus on the threat vectors as highlighted previously in the Risk Analysis Matrix figure.

Outsider Attack: Internet

- Newspaper, web articles on attacks at other places, Internet resources
- Hacking web sites
- Firewall/Intrusion Prevention logs are an excellent source for specific threats
- Scan your network with SNMP for cheaper routes to outside addresses than your firewall
- Internally, try traceroutes to private addresses
- Try to connect to your wireless networks from the parking lot

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Outsider Attack: Internet

Attacks coming from the outside to your internal network from the Internet have been well documented.

We hope you have implemented a firewall and an intrusion prevention system. If you have, the best sources of information on what is coming into your system are the firewall logs. If you do not have an intrusion prevention system, you might want to try one for a month or so. Place the IPS in your DMZ for a few days and log everything. This log tells you about the actual threats directed against your site.

Another effective method of determining what threats or attacks you are vulnerable to is to scan your own network. One example is a scan that looks for significant problems such as back doors. If you scan your site for SNMP agents that answer to the community string, public, you might be surprised how many there are. Look for broadcast packets from your internal/private address space coming in from the Internet. If you find evidence of this, it is most likely someone trying to get into your internal network from the outside.

Also look for ways into your network that might bypass the firewall, such as wireless LAN access points. In fact, check out any network device installed with its default settings. More times than not, the default settings are very weak and are easily exploitable.

I remember the first situation where I installed a personal firewall on my laptop. One day I was staying at a hotel in Florida that had Internet access. I turned on the log feature on my firewall because I was curious to see if anyone would be trying to get into my system once it connected to the hotel network. I was not online for more than 5 minutes when my firewall started beeping at me. Looking at the logs I found out that another machine was trying to do a network scan! Unfortunately, I had not put the firewall at its highest setting — the stealth mode. So they were able to see me on the network. I immediately configured it so I was completely hidden from network view, and added the machine trying to scan me to the list of blocked IP addresses.

Insider Attack: Internal Net

- Insider attacks are fairly advanced, subtle, and there is a potential for trouble — especially with a false alarm. Be careful!
 - Deploy network IPS on internal nets
 - Deploy host-based IPS on internal hosts
- Unix Xinetd, TCPwrappers, or Windows event logger reporting to central Syslogs is simple and effective

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Insider Attack : Internal Net

Using firewalls and intrusion prevention systems to monitor activity coming from outside your network is vital. But that does not mean you need to neglect looking for and monitoring your system from attacks from the inside. Insider attacks are often fairly advanced or subtle, and there is the possibility of extensive damage. This is because in many cases, security inside the network is more relaxed. Remember, if an attacker can penetrate your perimeter—via a backdoor, through a hole in the firewall, or using malicious code delivered by an e-mail—you will not see his activity if you are not monitoring your internal network. It is considered a good practice to:

- Deploy network-based IPS on high-value internal networks
- Deploy host-based IPS on high-value internal hosts

Many commercial and freeware products can help you monitor your internal network. In a Unix environment, use a host-based intrusion prevention system to report system events to a central location. Just make sure the central log system is well secured. If an attacker can modify your primary source of information, you might never figure out how he is getting in.

Another popular host intrusion prevention system is a product called Tripwire (<http://www.tripwire.com>). It is available for Unix and Windows. Tripwire can be configured to monitor critical system files.

It creates an MD5 hash value of the file and if the file is modified, it can detect a change because the new MD5 hash of the modified file is different than the original value. Although personal firewalls are ideal for laptop computers that are used on the road, they are ideal for use on internal host systems as well. Some organizations even implement keystroke monitors. One caveat is that the use of keystroke monitors requires a wiretap authorization.

The main issue that you will have to deal with is that you will have to review all of these audit logs to find out what is going on with your systems. The native Windows and Unix logs are not easy to work with; they are hard to parse and they don't always give you exactly the information you need. But there are many third-party tools that can make your life a lot easier.

Insider Attack: Honeypot

- Although insider attacks are tough to detect, it is pretty easy to instrument attractive systems or programs and watch for access:
 - Contract management systems
 - Pay and leave systems
 - Advanced research systems

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Insider Attack : Honeypot

Another way to monitor insider attacks is by using a honeypot system. The point of a honeypot is to confuse an attacker, causing him to waste time trying to break in to a system of no value. Although he is attempting to break in, you can collect information about his tools and techniques. A honeypot system can be implemented to lure and monitor insider attacks. You can configure a honeypot that looks like a payroll system or perhaps an advanced research system. Then, you sit back and wait to see who, without the proper authorization, tries to gain access.

There is a collaborative project on honeypots at <http://project.honeynet.org>.

Malicious Code

- Virus-scanning software:
 - How many viruses are being detected?
 - How are they getting in?
- Tripwire Unix and Windows (Host IDS):
 - Monitors the malicious modification of critical system files
- Egress (outbound) filtering:
 - Can detect systems infected with malicious code because they often use spoofed IP addresses

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Malicious Code

Malicious code is one of the more significant problems organizations face today. Virus-scanning software can be configured to generate reports as to how many viruses are being detected. If you see an increasing amount of viruses reaching some of your internal systems, perhaps you should look at how they are getting in. Many e-mail attachments are in reality Trojan horses that might install software on your systems to facilitate unauthenticated remote access. Installing a host IDS product such as Tripwire will help monitor the malicious modification of critical system files.

You should be concerned with malicious code entering your systems and make sure that your systems are not the launching pad for attacks to other systems. Many system administrators are great in implementing ingress (inbound) filtering, but forget all about egress or outbound filtering, or are not permitted to implement it. Egress filtering consists of setting your routers and firewalls not to pass traffic outbound from your site to the Internet if the traffic does not come from your address space. This is a way of keeping spoofed packets from entering the Internet. In fact, egress filtering is an excellent way to detect systems infected with malicious code because they often use spoofed IP addresses.

Step 2: Asset Identification and Valuation

We already spent \$25K on a firewall and now you tell me we need Intrusion Prevention?

But, boss, the organization's net worth is over \$4 billion and our Web-based sales last year alone exceeded \$45 million.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Step 2: Asset Identification and Valuation

Although management might have a good understanding of the cost of hardware, software, maintenance, and licensing fees, it might not be aware of the value of information assets. It might be a good idea to document the valuation of the assets you want to protect. If you know what your assets are worth, it is easier to justify the increased cost of the security controls. For instance, you might find it easier to justify spending \$25,000 on a firewall when you are protecting a system that generates more than \$2.5 million in revenue per year.

Assets come in many shapes and forms. If your e-commerce web site processes credit card orders and maintains customer records, your most valuable asset might be the client database system. Or perhaps your organization is involved in developing a new drug or vaccine. Your asset would be the systems that contain information about this new product or discovery. If this information fell into your competitor's hands, it could be catastrophic to your bottom line.

The asset valuation might take the form of hard monetary values. Management understands the quantitative analysis of asset valuations. For instance, your company's web-based sales generate over \$2.5 million in sales. This high dollar loss in revenue could be the catalyst needed to convince management to spend more money on security controls. The difficulty comes in the valuation of intangible assets, such as projected income. Nevertheless, anything that is central to a revenue center will have a higher bottom-line importance in management's eyes.

Step 3: Vulnerability Analysis

- Vulnerabilities are the gateways by which threats are made manifest:
 - Doorways for the use of exploit code or techniques
 - Increase frequency of threat event
 - Increase impact of threat event

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Step 3: Vulnerability Analysis

As stated earlier in this chapter, vulnerabilities are weaknesses that could be used by an attacker to compromise a system. Every day, new vulnerabilities are uncovered. There are several web sites that can keep you informed of the latest vulnerabilities.

Although it is critical to learn about newly discovered vulnerabilities, it is just as critical to know about older vulnerabilities. It is important because many systems do not have the latest hotfixes or patches installed. Many attacks exploit older vulnerabilities because they know that some systems are not patched correctly.

It is ironic that in many cases, the most critical systems in our enterprise are the ones that do not have the latest patches. This is because the application of hotfixes and patches typically requires the system to be rebooted. These critical servers (for example, web servers, domain controllers, e-mail servers, and so on) need to be operational 24/7, and bringing these systems down for any amount of time is not always an option. So, these critical servers remain unpatched and, therefore, vulnerable to well-known attacks.

Step 4: Risk Evaluation

- Match threats and known vulnerabilities, calculate ALE
- Estimate risk from unknown (not yet discovered) vulnerabilities
- Risk might be expressed monetarily (preferred) or qualitatively

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Step 4: Risk Evaluation

Understanding the risks to the enterprise is only the beginning part of the process. The next step is to perform a risk evaluation. Risk evaluation is the process of taking the vulnerabilities identified in the previous step, and determining the impact levels (if any) they will have on your enterprise if exploited.

Not all vulnerabilities will impact your organization. Remember our discussion on vulnerabilities, where we might see a notice that says, "A vulnerability has been found in the XYZ FTP client service that if exploited, could result in a buffer overflow"? It could very well be that, although this vulnerability could cause major damage to our system if exploited, this vulnerability might not even affect us whatsoever (that is, we do not have the FTP client installed so that vulnerability cannot be exploited). Risk evaluation involves looking at the vectors that you know apply to your organization, and then list some known vulnerabilities that can be exploited via these vectors, regardless of any countermeasures installed.

After the threats are matched to known vulnerabilities, you can then calculate the annualized loss expectancy (ALE). Remember that the ALE is calculated by multiplying the single loss expectancy (SLE) by the annualized rate of occurrence (ARO). The SLE is calculated by multiplying the asset value (AV) times the exposure factor (EF). When calculating the value of the assets, you should consider the investment value, the cost of the hardware and software, the time your organization has invested, and also the potential loss of revenue.

Step 5: Interim Report

- Project summary: To evaluate the possible need for an IPS capability, we decided on possible threats and tested for evidence that they were actually in use. We also ran vulnerability tests to determine our degree of exposure.
- Asset identification and valuation report: Present what was found and its value. As for threats, give some worst cases and probable long-term sustainable losses, in dollars, if possible.
- Plan to make things better: NEVER brief senior management without a plan in your back pocket

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Step 5: Interim Report

The interim report is essentially your pitch to management. This report should contain a project summary, which addresses all of the steps you took to arrive at the decision in the report. For example, let's say your task was to determine whether a new IPS is needed for your enterprise.

The project summary might look something like this:

"To evaluate the possible need for an IPS capability, we decided on possible threats and tested for evidence that they were actually in use. We also ran vulnerability tests to determine our degree of exposure...."

The report should also contain an asset identification and valuation report. This information is critical, because management will then have a better understanding of what are the valuable assets the company has, and help better justify the cost of the countermeasures to protect them. This section contains the information of what assets were identified and their value. When relating to the threats to these assets, give some worst cases and long-term sustainable losses.

The interim report should always have a plan to improve the situation. Never brief senior management without a plan on how to solve the problem. Be prepared to recommend solutions and more importantly, justify your recommendations.

Acceptable Risk—Who Decides?

Who in your organization is actually authorized to decide what level of risk the organization will accept?

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Acceptable Risk— Who Decides?

An important question you need to answer is, "Who in your organization is actually authorized to decide what level of risk the organization will accept?" An issue that comes up from time to time is that lower-level managers make risk decisions that can potentially adversely affect the entire organization. This is partly a function of not understanding the technologies and risks involved. One method of mitigating this is with awareness training.

Cost Benefit Analysis

- Comparison of the cost of implementing countermeasures with the value of the reduced risk:
 - Easy countermeasures generally show cost benefit
 - Hard effective countermeasures (for example, host-based IPS) also should show cost benefit

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Cost Benefit Analysis

Another consideration factor is the cost of the safeguard versus the actual value of the asset. It makes sense that the cost of the protection should not be more than what the asset is worth. Would you buy a \$5,000 safe to protect a ring that is worth only \$100?

Cost benefit analysis is the comparison of the cost of implementing countermeasures with the value of the reduced risk. Can you accurately determine whether the countermeasure is 100% effective? Anti-virus software is known to fail against unknown viruses, especially if the virus signatures are not up to date. Companies with firewalls are not 100 percent protected, because firewalls can be compromised (or bypassed), and because traffic containing attacks might legitimately pass through the firewall.

Benefits are the reduction in the risks your company is exposed to. Keep in mind that the biggest benefits to the organization might be the countermeasures that protect the revenue flow. This is especially true if your organization is involved in e-commerce. The cost of a countermeasure is more than just the initial cost. There is the labor cost of monitoring the devices and the lifecycle cost.

"Final" Report

- Includes the interim report results
- Safeguard selection:
 - Including easy-to-do tasks that have already been implemented
- Risk mitigation analysis
- Cost benefit analysis
- Recommendations

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

"Final" Report

The final report given to management includes the interim report results, as well as the safeguard selection. These countermeasures include the actual technology control selections (for example, IPS, firewalls, and anti-virus software) and might also include specific tasks. Sample tasks might be to review the IPS logs more frequently or increase the anti-virus software signature update interval. If these easy-to-do tasks have already been implemented, state so in the final report. Other components of the report include the risk mitigation analysis, the cost benefit analysis, and your recommendations.

When discussing the recommendations, it is best to include more than one option or solution to the problem. Make sure to map out the entire recommended architecture. Sometimes the report can outline short-, medium-, and long-term solutions. Short-term solutions are those that can be implemented in a short amount of time and typically for little or no cost. Long-term solutions could involve a complete re-design of the infrastructure and an increased safeguard solution cost. For instance, the short-term solution could be to use VLAN ACLs to segment the network. If this is a concern, you can prepare the policies in advance and have them ready to load if a worm crisis occurs. Although the best long-term strategy is probably to implement host-based and/or network-based IPS.

Business Case: Summary

- Threat assessment and analysis
- Asset identification and valuation
- Vulnerability analysis
- Risk evaluation
- Interim report
- Establish risk acceptance criteria
- Safeguard (countermeasure) selection with risk mitigation analysis
- Cost benefit analysis
- Final report

A security professional must know the threat, understand the fundamental information security tools, and be able to apply this knowledge in risk management.

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Business Case: Summary

This module discusses the process of risk management. We discussed how there are vulnerabilities in the software and hardware systems we use. Some of these vulnerabilities are documented and some have yet to be found. Vulnerabilities are weaknesses—security holes—that can be exploited and used to compromise the system. These vulnerabilities, coupled with threats, add up to the risks we are exposed to. Risk management is needed to make sure we are prepared just in case those threats become a reality.

We learned how we do not need to be concerned with all of the vulnerabilities in our systems. We need to look at it in terms of what vulnerabilities are those we should be concerned with (the ones that directly affect you), what is the likelihood someone could exploit the vulnerability, and what would be the impact if the threat did become a reality. Calculating the single loss expectancy (SLE) uses the value of the asset you are protecting and the exposure (to the risk) factor. The annualized loss expectancy (ALE) uses the SLE value and the annualized rate of occurrence for that particular risk. These results are used to enable either a quantitative or qualitative risk assessment approach. Quantitative assigns a numeric value to the risk whereas qualitative produces a more subjective risk assessment (for example, low-, medium-, or high-risk factor).

An effective risk management process involves working within the security framework of the security policy, identifying the various risks the organization might be exposed to, implementing a security infrastructure to handle those risks, deciding what controls should be used to build the infrastructure, and deciding what countermeasures to implement, based on the impact and severity of the risk.

Keeping track of vulnerabilities in operating systems, applications, and security devices is a full-time job. Luckily, there are many Internet sources of information, including e-mail notification subscription services that can inform you of new vulnerabilities found, and what you can do to fix the problem.

Further Reading

Keeping your systems configured correctly is also a monumental task. Relying on one person or company to provide the definitive security configuration is not always the recommended approach. You should use recommendations developed by a consortium of individuals and corporations. This method provides best practices security recommendations from a more diverse group of experts that come together to share their knowledge and experience.

Risk analysis is the process of analyzing the threats to and vulnerabilities of an information system and the potential impact the loss of information or capabilities of system would have on national security and using the analysis as a basis for identifying appropriate and cost-effective measures. Risk assessment provides a concise analysis of the IS security posture and identifies significant vulnerabilities, safeguards, and countermeasures. It is used to support the expenditure of resources and determine the most cost-effective safeguards.

Organizations can also use NIST SP 800-30 – “Risk Management Guide for Information Technology Systems” as a reference.

SEC401 Lab Tools: Secure Communications

SANS Security Essentials - © 2016 Secure Anchor Consulting, LLC

SEC401 Lab Tools – Secure Communications

This section intentionally left blank.

S-Tools

S-Tools is a steganographic tool
used to hide data in
BMPs, GIFs, and WAV files
on Windows systems.

SANS Security Essentials - © 2016 Secure Anchor Consulting, LLC

S-Tools

S-Tools can be used to hide messages inside BMP, GIF, and WAV files. Depending on the options that you choose, the output file that contains the hidden data might have different properties than the original file.

S-Tools Details

- Name: S-Tools
- Operating system: Windows
- License: Freeware
- Protocol used: NA
- Category: Steganography
- Description: S-Tools is a GUI tool used to hide data in multiple file types.

SANS Security Essentials – © 2016 Secure Author Consulting LLC

S-Tools Details

The following topics and action items are covered in this chapter:

- Installing S-Tools
- Running S-Tools
- Hiding files in images
- Hiding files in WAV files

S-Tools Background

- S-Tools version 4 is now a drag-and-drop Steganography tool for use with BMP, GIF, and WAV file types

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

S-Tools Background

This section intentionally left blank.

S-Tools' Purpose

- Like all Steganography tools, the main purpose of S-Tools is to hide data in multiple data types
- It gives the user the ability to encrypt and hide data in a manner to hide itself through obscurity
- It utilizes symmetric cryptography for its encryption method

SANS Security Essentials – © 2010 Secure Anchor Consulting, LLC

S-Tools' Purpose

This section intentionally left blank.

S-Tools Architecture

- S-Tools is a GUI-based application for use on Windows platforms
- It has the ability to both hide and retrieve information
- It can retrieve only data that was hidden by S-Tools itself

SANS Security Essentials – © 2010 Secure Anchor Consulting LLC

S-Tools Architecture

The file in which data is hidden is called the “carrier file.” After hiding a file within a carrier file, you can send the carrier file to another person who knows to use S-Tools with the appropriate password; he or she will be able to view the contents of the hidden file, while unsuspecting others view a regular image file.

Installation

- Like many Steganographic tools,
there is no installation required

SANS Security Essentials - © 2016 Secure Anchor Consulting LLC

Installation

S-Tools does not require installation. The files are already extracted so just copy the *S-tools* directory from the course media to *C:\tools* directory.

Running S-Tools

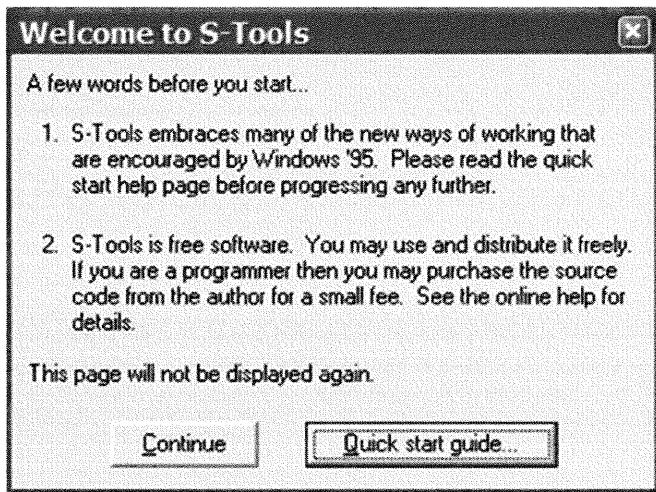
- To run S-Tools, simply double-click the *S-Tools.exe* executable provided on the DVD



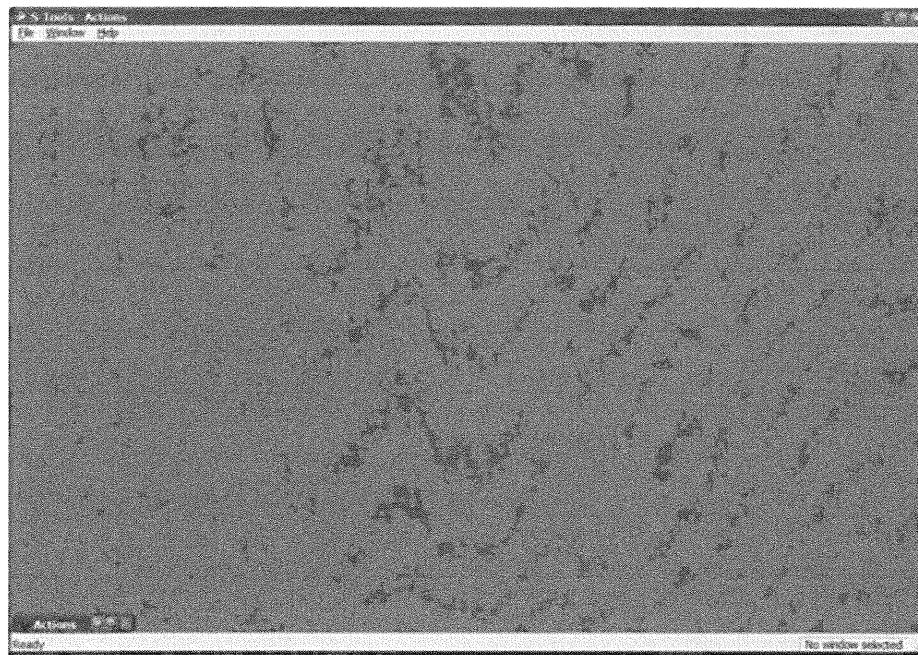
Running S-Tools

The following are the steps that need to be performed in order to install S-tools on your system.

1. Start S-Tools by double-clicking *S-Tools.exe*, which is located in C:\tools\s-tools\. The interface for S-Tools launches and you are presented with the following screen.



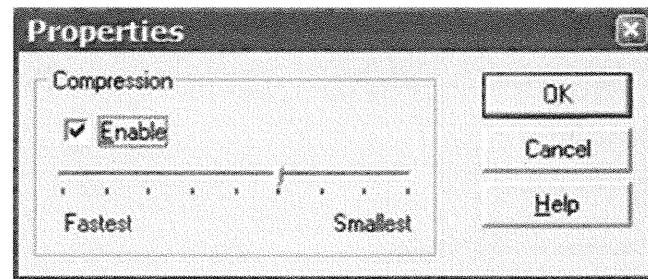
2. Click *Continue* to get to the main screen. As you can tell, there are not many options in the S-Tools interface.



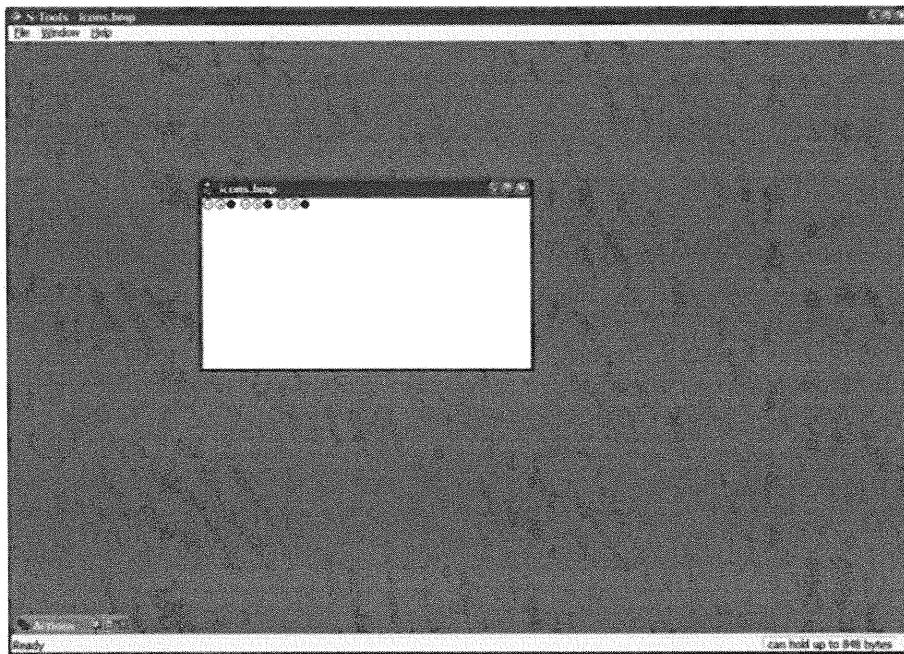
Hiding Files in Images

To hide files in images with S-Tools, perform the following steps:

1. Click *File, Properties* to view the compression ratio for the file that you are hiding. The higher the compression, the longer it takes to hide a file; however, you will be able to hide larger files when the compression is cranked up. Click *OK* to close the Properties dialog box.

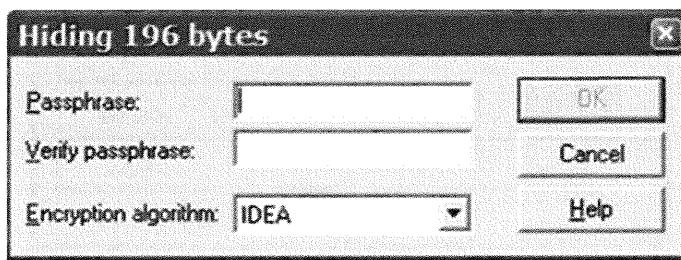


2. Start a Windows Explorer session and find one of several BMP files that are on your system. Search for any file ending in .bmp in your *c:\Windows* or *c:\Program Files* directories. Copy that file to your *s-tools* directory and then drag the BMP file onto the
3. *S-Tools* window. For this example, *c:\icons\icon.bmp* is used.



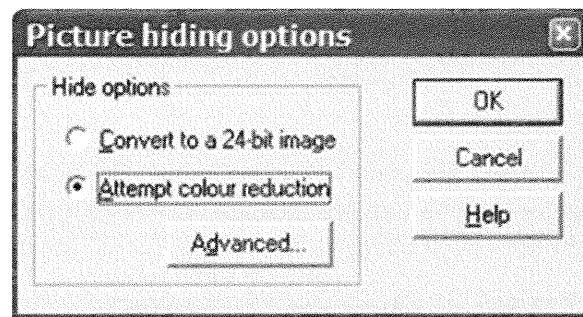
4. Open *WordPad* and type **Secret_data.txt** to create a document. Now drag **Secret_data.txt** on top of the BMP image that is in the S-Tools window. You will receive a passphrase prompt, as shown in the following screen. Enter and confirm the passphrase that you will use to secure the file. The dialog box also details the amount of data that you are hiding; in this case, you are hiding 196 bytes of data inside the BMP file.

S-Tools natively supports a number of encryption algorithms, including IDEA (the default), DES, Triple-DES, and MDC. Accept the default value of IDEA and click *OK*.



5. As shown in the following screen, the next prompt deals with the quality of the output file. If you convert it to a 24-bit file, the output file will be larger than the original due to the storage of the extra hidden data. The other option, Attempt color reduction, reduces the quality of the picture in an attempt to keep the output file size close to the original. Accept the default value of *Attempt color reduction* and click **OK**.

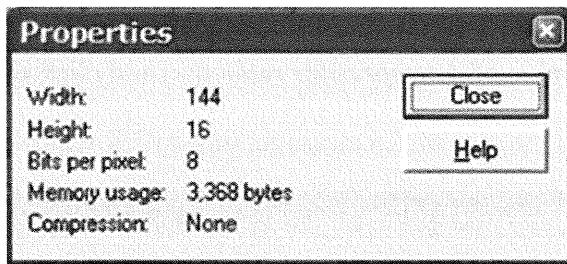
Note: Depending on the size of the message, this option window might not be displayed.



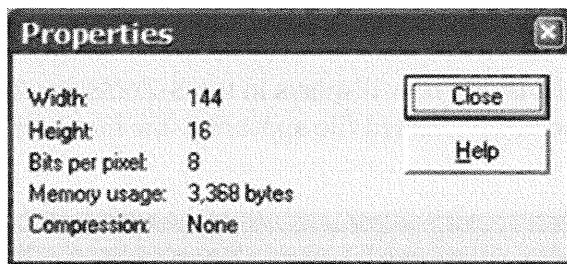
6. The newly created .BMP file now displays in the S-Tools interface. Notice that in the following screen, the newly created file appears to be the same as the original file.



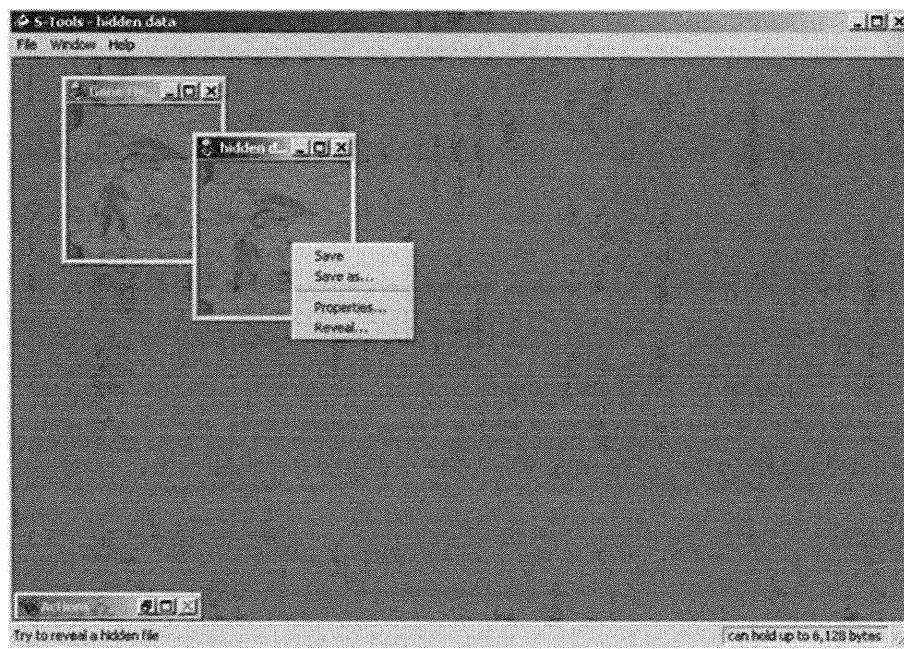
7. Right-click the picture that includes the hidden file, and then click *Properties*. Notice the dimensions of the file as well as the memory usage.



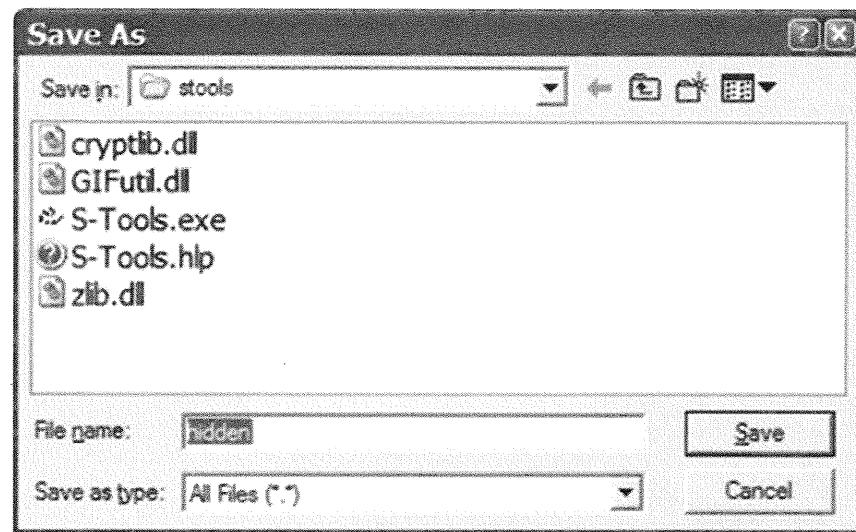
8. Now right-click the original file and click *Properties*. The dimensions are the same. Note: depending on the size of the file you are hiding and whether color reduction has been performed, you might see slight changes in the results that are displayed.



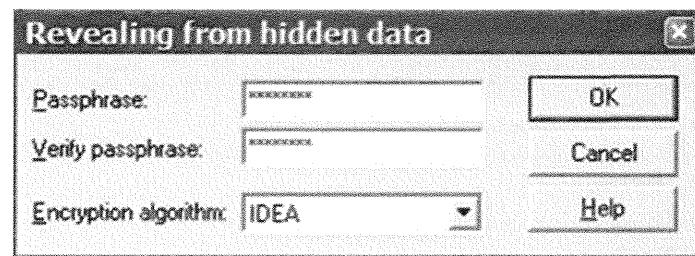
9. To save the newly created hidden file, right-click it and click *Save as*.



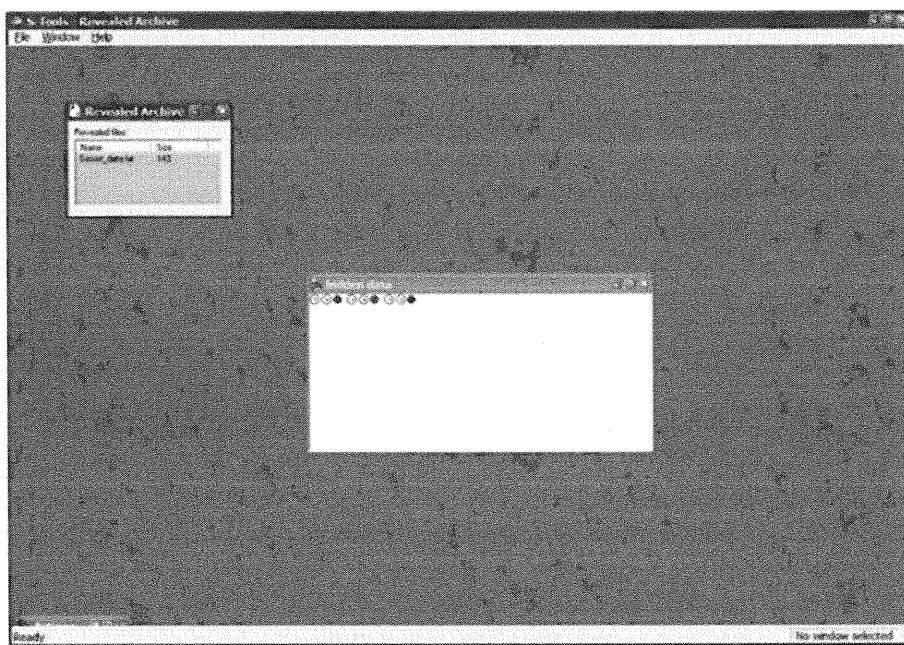
10. Save the file as **hidden.bmp**.



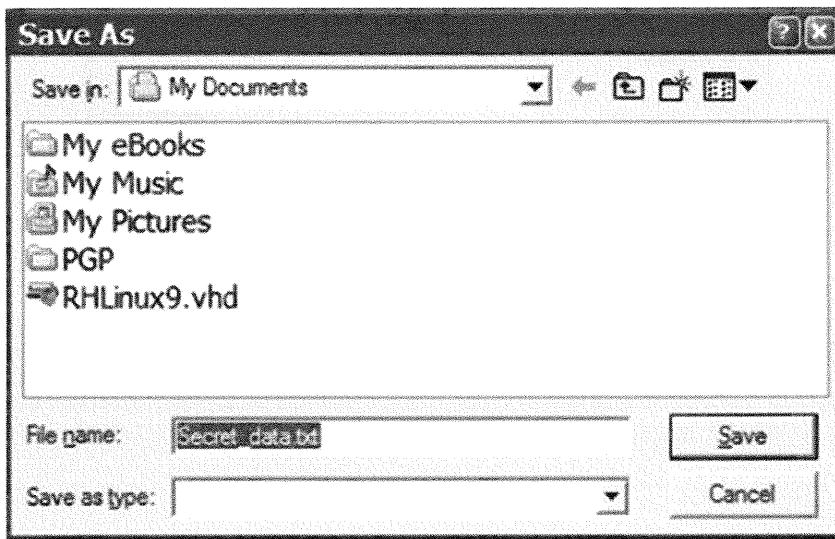
11. After you have saved the file, close S-Tools and then restart it. Drag the newly created *hidden.bmp* file onto the S-Tools window. Then right-click the picture and click *Reveal*.
12. You are now prompted for the passphrase in order to reveal any data hidden in *hidden.bmp*. Enter and confirm the passphrase that you've selected, and click *OK*.



13. The Revealed Archive window (and the hidden file *Secret_data.txt*) now displays.



14. Highlight *Secret_data.txt*. Then, right-click it and click *Save as*, as demonstrated in the following screen. Save the file as **Secret_data_after_steg.txt** so it will be easy to compare it to *Secret_data.txt*.

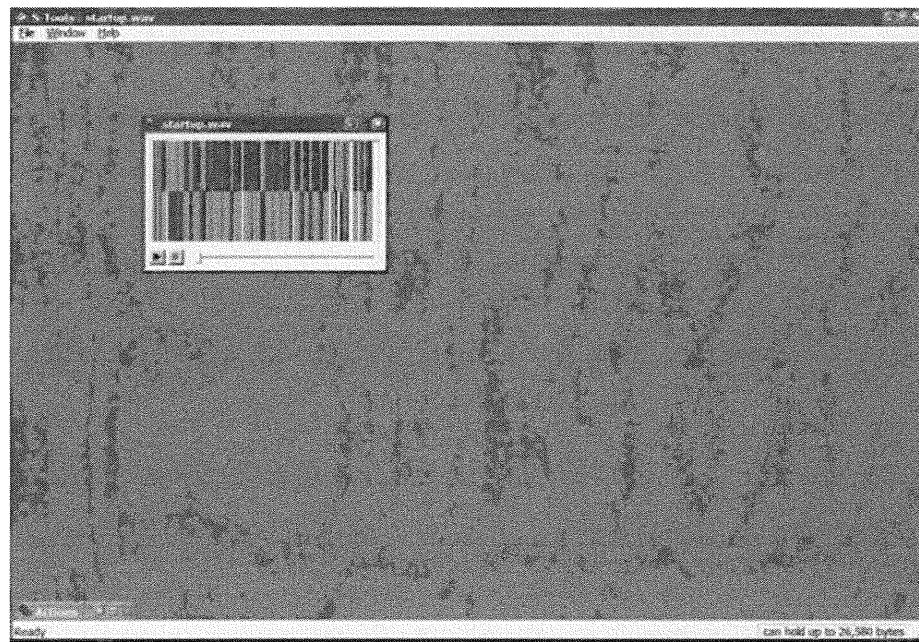


15. You can now open the new text file to ensure that your message is still intact.

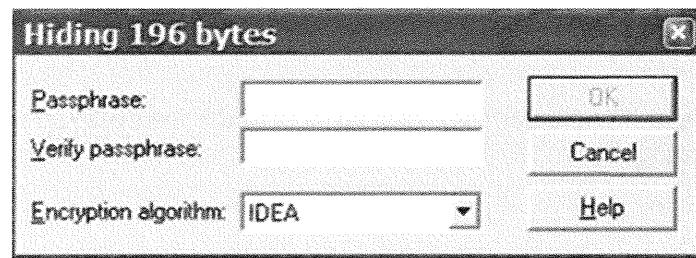
Hiding Files in WAV Files

Images are not the only things that S-Tools can use for hiding files. This section describes how you use the same steps in the previous section to hide files; however, this time, you use a WAV file as the carrier file. Follow these steps:

1. Using Windows Explorer, find a WAV file on your system and drag the file onto the *S-Tools* window. For this example, you can use *C:\WINDOWS\Media\Windows Startup.wav*.

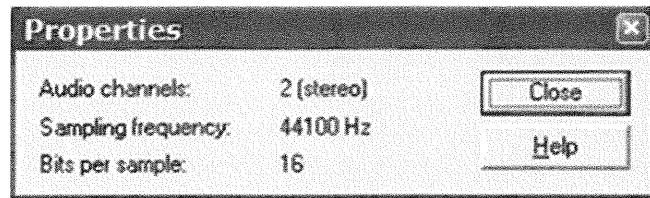


2. Again, drag *Secret_data.txt* onto the WAV file. Enter and confirm the passphrase with which you want to secure the hidden file, and then click *OK*.

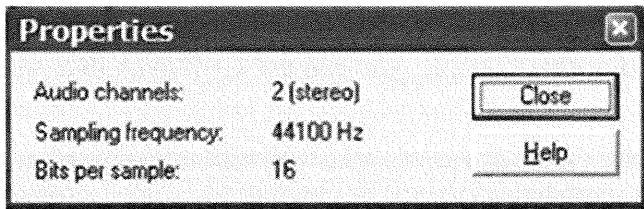


The new file containing the hidden data is now present in the S-Tools window. Viewing the properties of these two files, as shown in the following screen, reveals that the audio properties are the same, but the file sizes and dates are different.

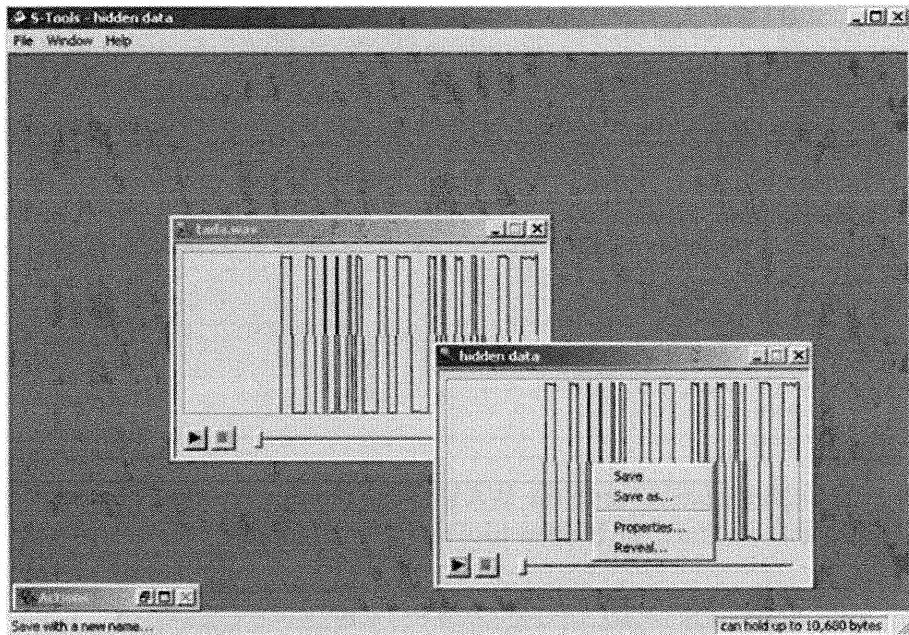
The following screen displays the hidden file's properties:



The following screen displays the original file's properties:



3. Next, right-click within the hidden data window and click *Save as*. Accept the default file name *hidden.wav*.



4. After saving the newly created file, you can reveal the information in the same way that you did for the .bmp file.

Exercise: S-Tools

1. Can you use S-Tools to hide files within .mp3 files?
2. Will the size of the new carrier file be different than the original file?
3. Does the recipient of the carrier file need to have S-Tools to view the hidden data?
4. Will a WAV file that contains hidden data still play?

SANS Security Essentials - © 2016 Secure Analysis Consulting, LLC

Exercise: S-Tools

The following questions are answered in the following section:

1. Can you use S-Tools to hide files within .mp3 files?
2. Will the size of the new carrier file be different than the original file?
3. Does the recipient of the carrier file need to have S-Tools to view the hidden data?
4. Will a WAV file that contains hidden data still play?

Exercise Solutions: S-Tools

1. No, S-Tools supports BMP, GIF, and WAV.
2. Yes, the file will get either larger or smaller depending on a number of variables, including the size of the hidden data, the quality of the picture, and the compression settings.
3. Yes, S-Tools will need to be installed and the passphrase must be known (if one was set during creation).
4. Yes, a WAV file containing hidden data will still play.

SANS Security Essentials – © 2016 Secure Anchor Consulting LLC

Exercise Solutions: S-Tools

The following are the answers to the questions:

1. No, S-Tools supports BMP, GIF, and WAV.
2. Yes, the file will get either larger or smaller depending on a number of variables, including the size of the hidden data, the quality of the picture, and the compression settings.
3. Yes, S-Tools will need to be installed and the passphrase must be known (if one was set during creation).
4. Yes, a WAV file containing hidden data will still play.

Summary

- S-Tools provides an easy way to disguise sensitive data in various types of carrier files
- Experiment with S-Tools compression settings to get the best results

SANS Security Essentials - © 2016 Secure Anchor Consulting LLC

Summary

S-Tools provide an easy way to disguise sensitive data within media files. Practice is the best way to gain experience to know what compression setting works best for stealthily hiding data and still appearing as the original file.

Invisible Secrets

Invisible Secrets is a steganographic tool that allows people to hide text in multiple file types.

SANS Security Essentials – © 2016 Secure Anchor Consulting LLC

Invisible Secrets

This chapter introduces Invisible Secrets. We start off by using Invisible Secrets to hide information in a file.

Invisible Secrets Introduction

If you want a wider range of carrier file types and more options than S-Tools provides, then look no further than Invisible Secrets. Invisible Secrets is a security suite that lets you, among other things, hide files within the following file types:

- JPEG
- PNG
- BMP
- HTML
- WAV

Invisible Secrets Details

- Name: Invisible Secrets
- Operating system: Windows
- License: Trial version
- Protocol used: NA
- Category: Steganography
- Description: Invisible Secrets allows users to hide encrypted information in pictures.
- URL: <http://www.invisiblesecrets.com/>

SANS Security Essentials - © 2010 Secure Anchor Consulting LLC

Invisible Secrets Details

The following topics and action items are covered in this chapter:

- Installing Invisible Secrets
- Running Invisible Secrets

Invisible Secrets Background

- Invisible Secrets is an application created by Neobyte Solutions
- It is a full-featured stego tool that provides multiple encryption methods for protecting your hidden data
- It allows users to put sensitive data in seemingly normal files such as pictures of your latest vacation

SANS Security Essentials – © 2016 Secure Anchor Consulting LLC

Invisible Secrets Background

This section intentionally left blank.

Invisible Secrets' Purpose

- Hides data in pictures
- Encrypts hidden data
- Properly deletes information on your hard drive with its integrated DOD-compliant shredder

SANS Security Essentials - © 2010 Secure Anchor Consulting LLC

Invisible Secrets' Purpose

This section intentionally left blank.

Invisible Secrets' Architecture

- It is a Windows-based GUI application used to hide information in multiple file types
- It can use multiple symmetric encryption algorithms to secure hidden data
- It incorporates a DOD 5220.22-M compliant data shredder

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Invisible Secrets' Architecture

In this chapter, you use Invisible Secrets as a stenographic tool. If you do not need a covert method of disguising information but still need to store sensitive information securely, Invisible Secrets allows you to encrypt and decrypt files using a number of algorithms. Of course, these added features cost money. This chapter uses the 30-day demo version of Invisible Secrets.

Installation

- To install, make sure you are logged in as the administrator
- Locate the *invsecr.exe* file on your CD-ROM and copy it to your local drive
- Double-click *invsecr.exe* to begin the installation

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Installation

This section intentionally left blank.

Running Invisible Secrets

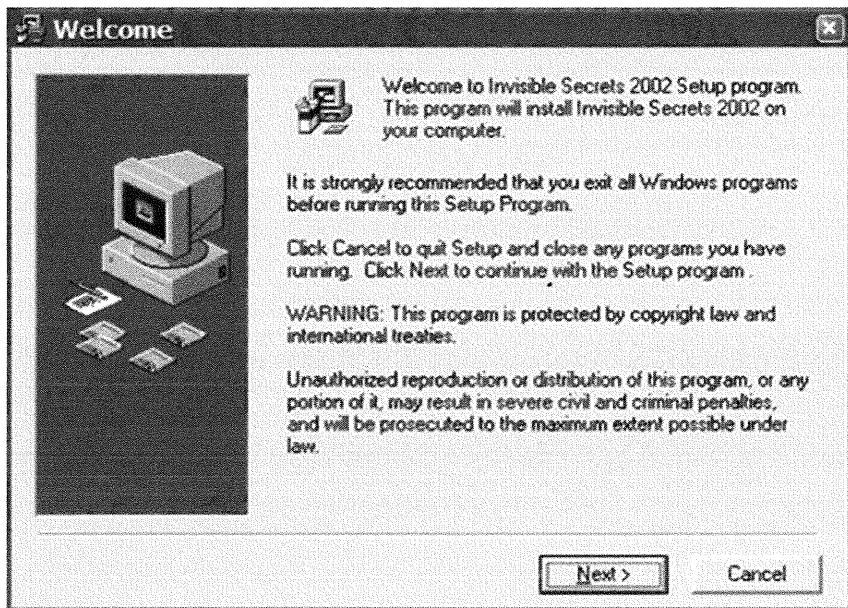
- To run the application, go to *Start, Programs, Invisible Secrets*
- Make sure that you have the appropriate authority to use this tool and that it does not conflict with any of your company's policies

SANS Security Essentials – © 2016 Secure Anchor Consulting LLC

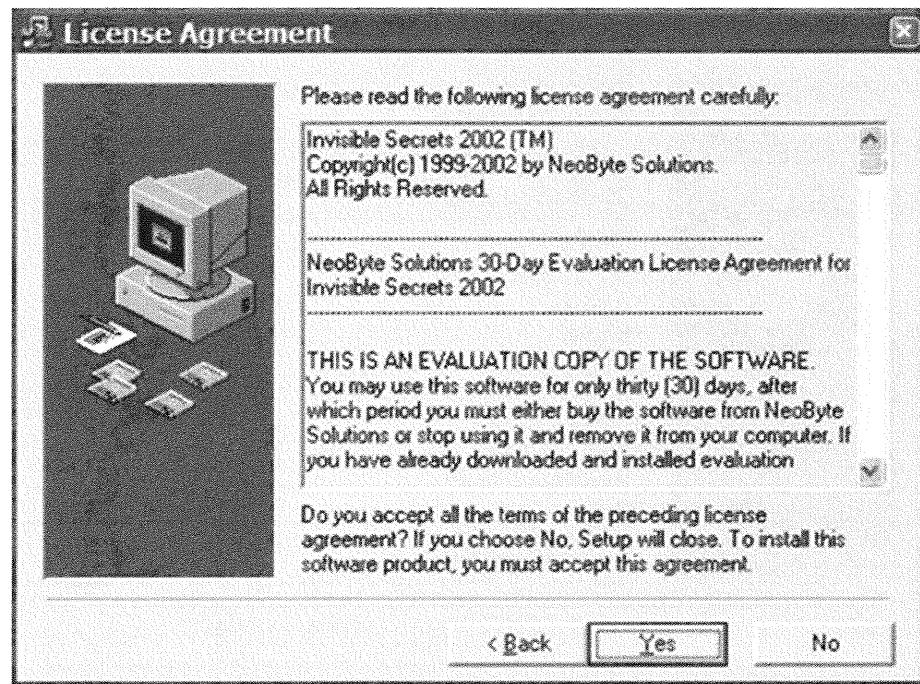
Running Invisible Secrets

To install the program, perform the following steps:

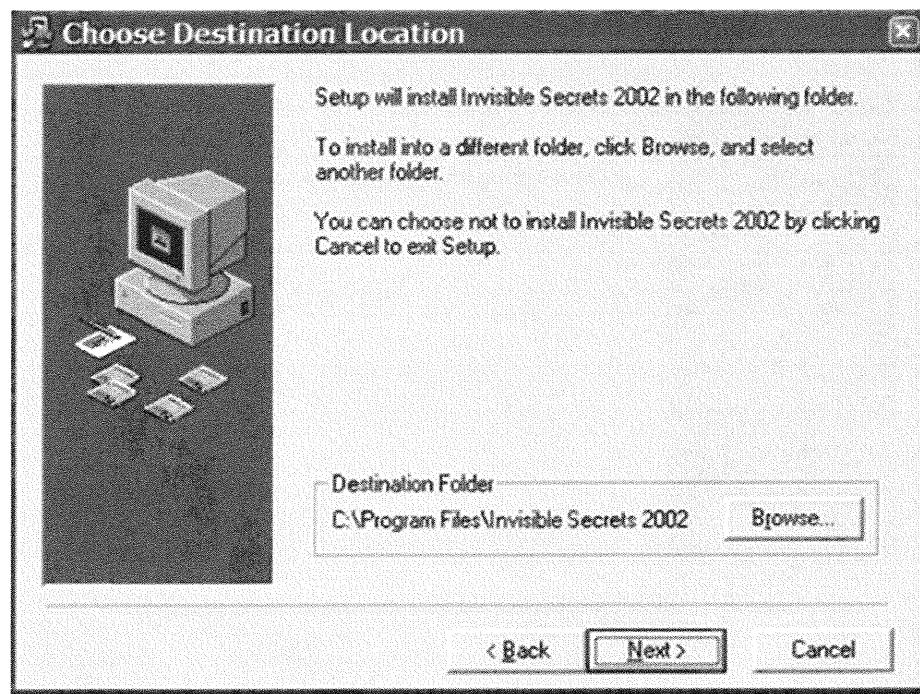
1. On the CD-ROM, locate *invsecr.exe*. Double-click it to start the installation process. In the Welcome window, click *Next* to continue.



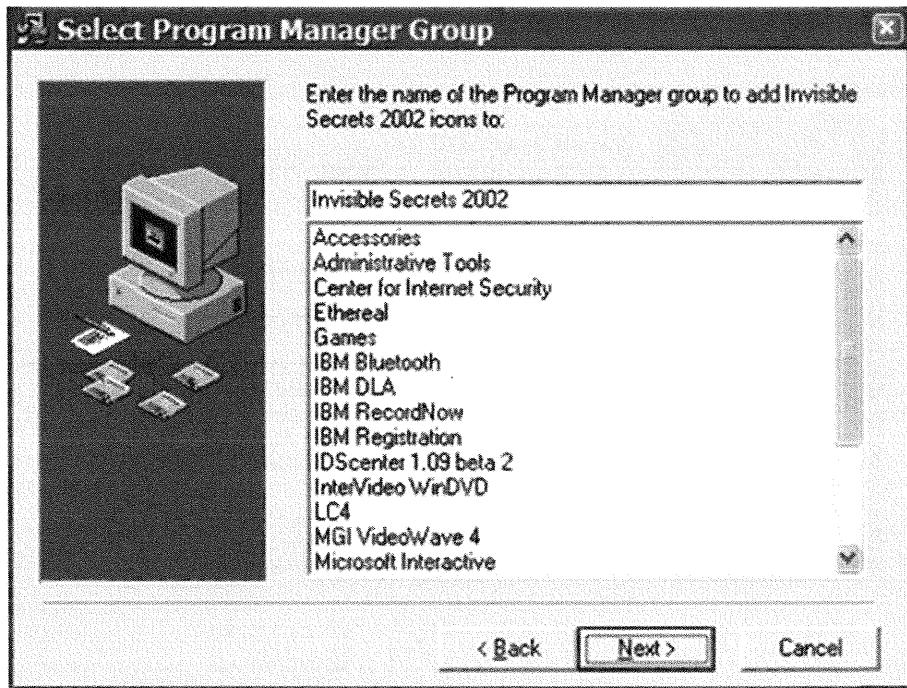
2. The License Agreement window displays. If you agree to the 30-day evaluation license agreement, click Yes; otherwise, you cannot continue with this chapter.



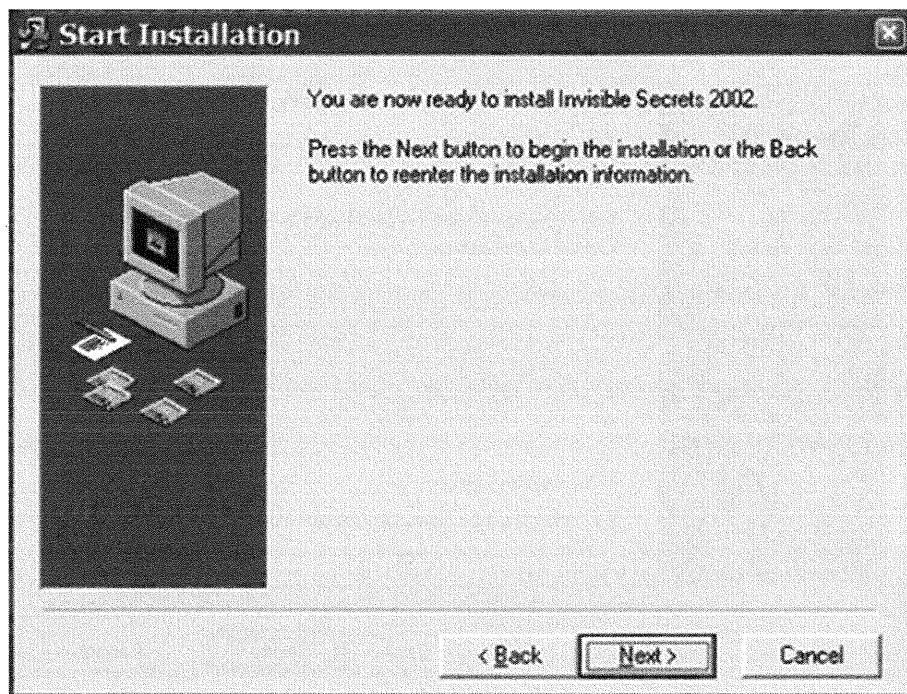
3. In the Choose Destination Location window, accept the default installation folder by clicking *Next*.



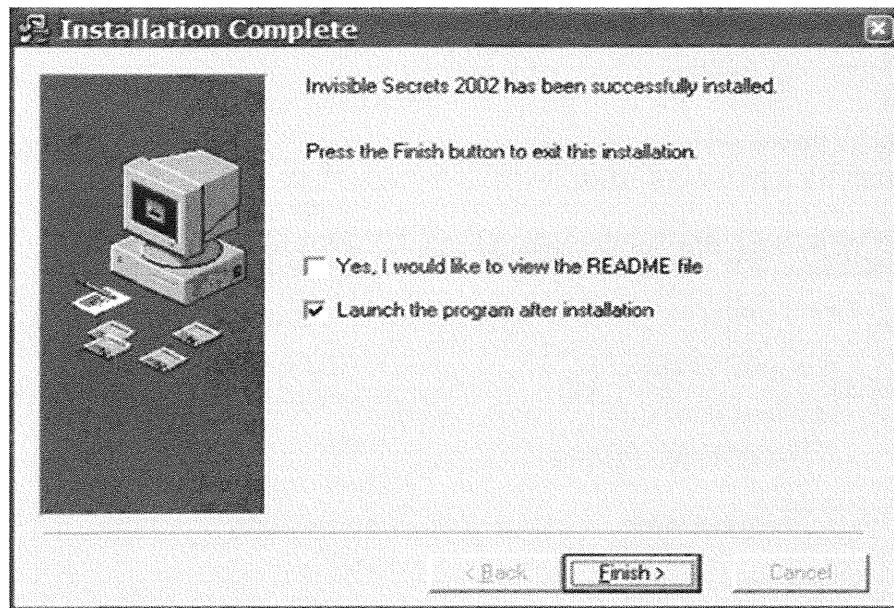
4. In the Select Program Manager Group window, accept the default Program Manager group by clicking *Next*.



5. In the Start Installation window, click *Next* to start the installation of Invisible Secrets 2002.



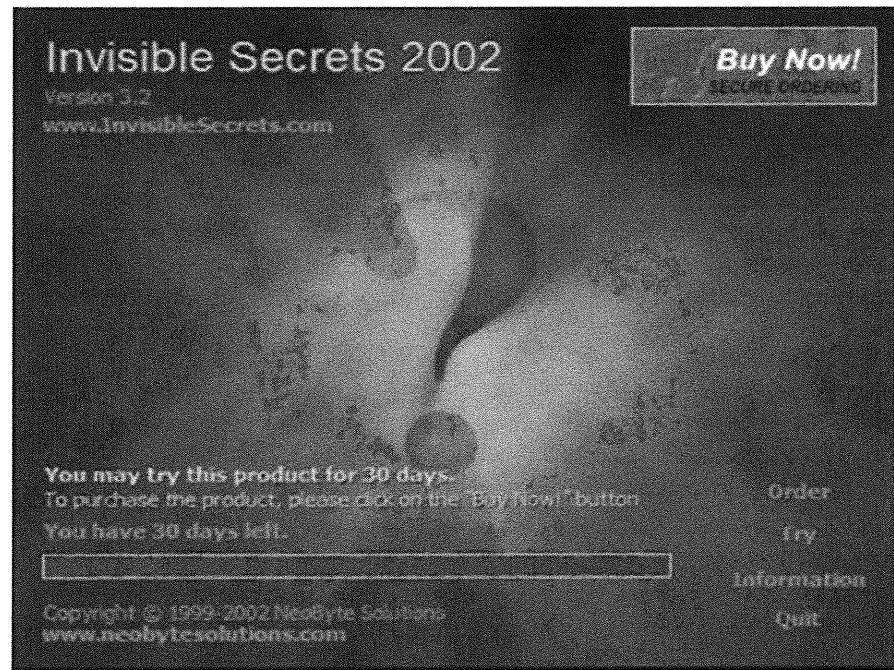
6. The Installation Complete window displays. After the installation is finished, uncheck *Yes, I would like to view the README file* and click *Finish*.



Running Invisible Secrets 2002

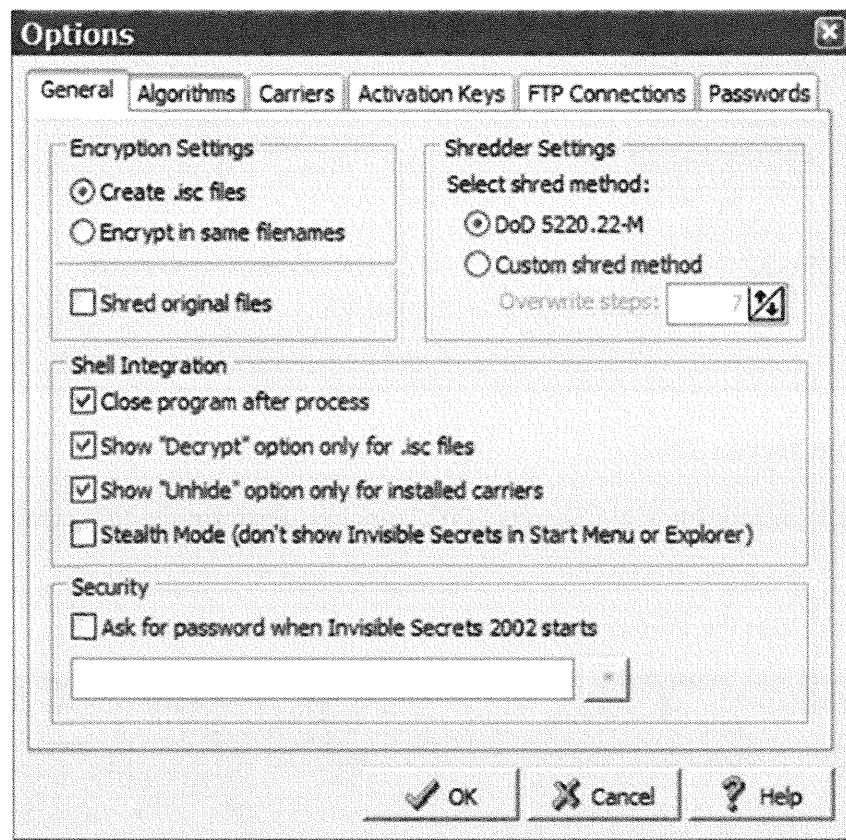
After you finish the install, Invisible Secrets 2002 launches. Perform the following steps to run it:

1. Click *Try* to start the 30-day countdown.

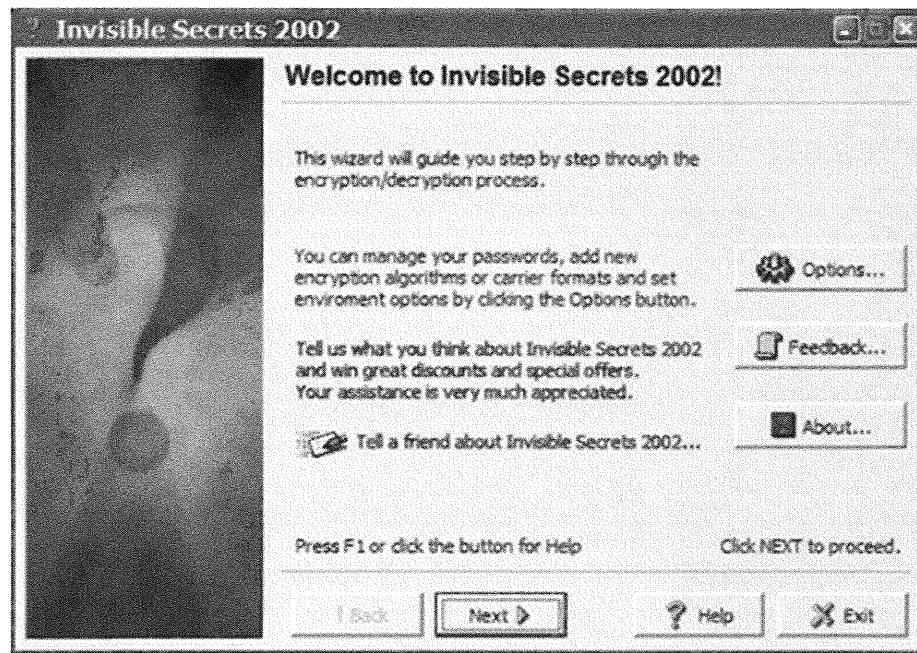


2. The settings you access with the Options button let you perform a number of tasks, including the following:

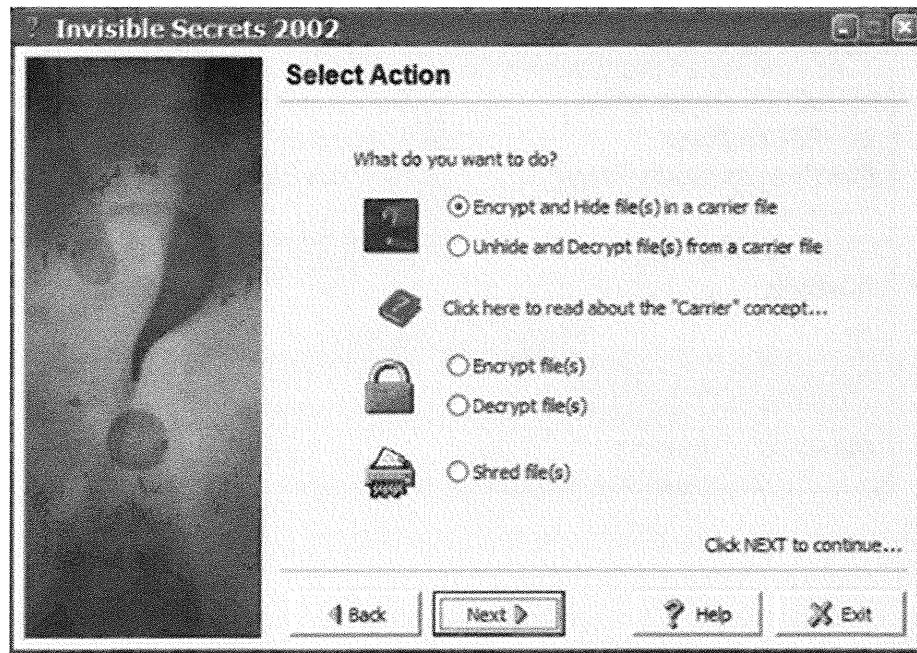
- Add new carrier file types
- Add new algorithms
- Set the number of times to overwrite a file when shredding
- Manage passwords
- Prompt for a password when Invisible Secrets starts



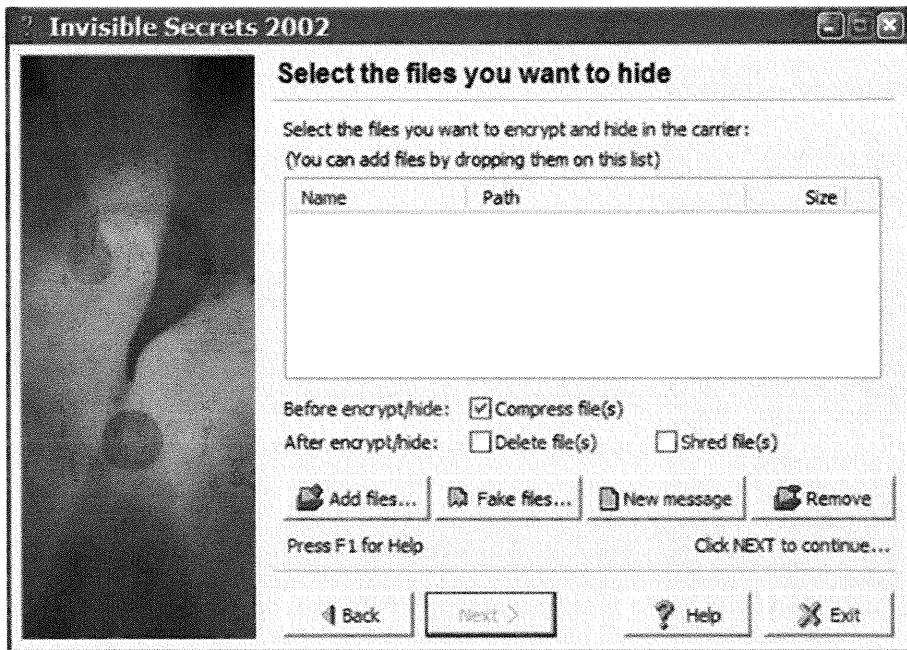
3. In the Welcome window, you use the default settings for the purposes of this book. Click *Next* to start the fun.



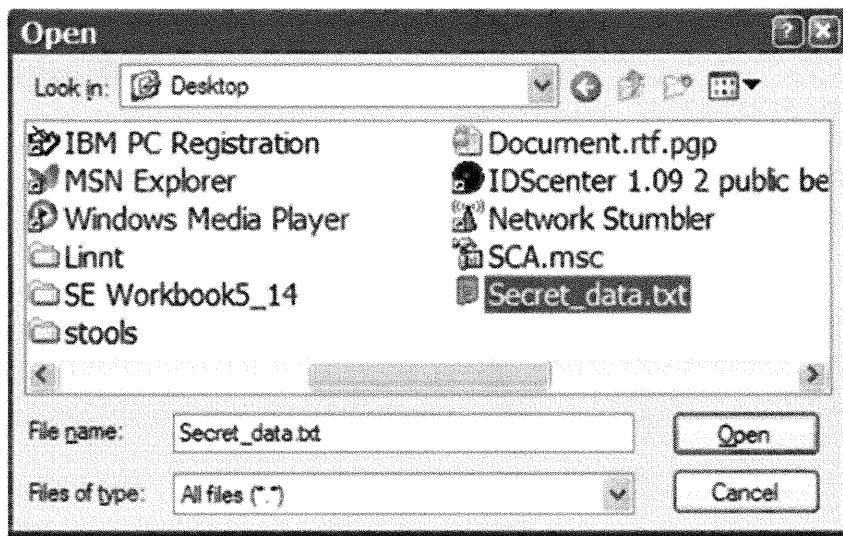
4. The Select Action window displays. As shown in the following screen, you can select the action that you want to perform. Use the default setting *Encrypt and Hide file(s) in a carrier file*. Click *Next*.



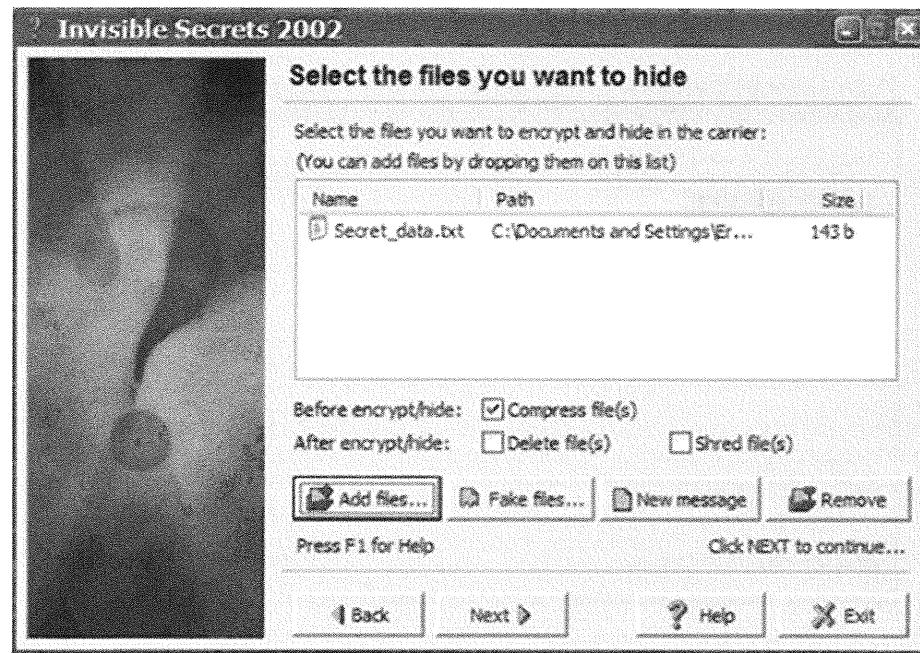
5. In the Select the files you want to hide window, you are prompted to select the files that you want to hide. Click *Add files*.



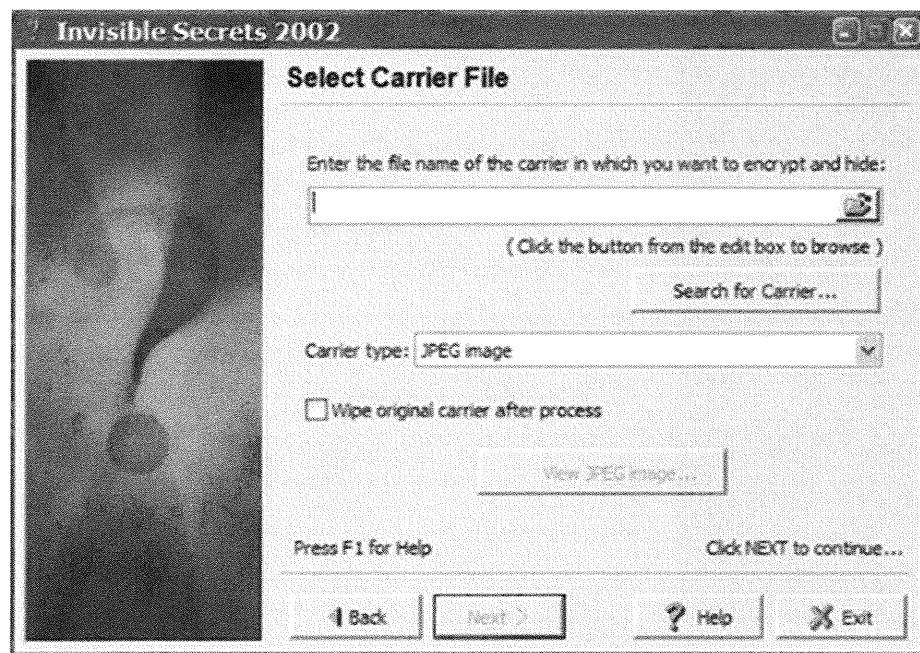
6. Select a file to hide. I am going to use the secret-data.txt file that I created in the S-Tools exercise, but any text file will do. Then, click *Open*.



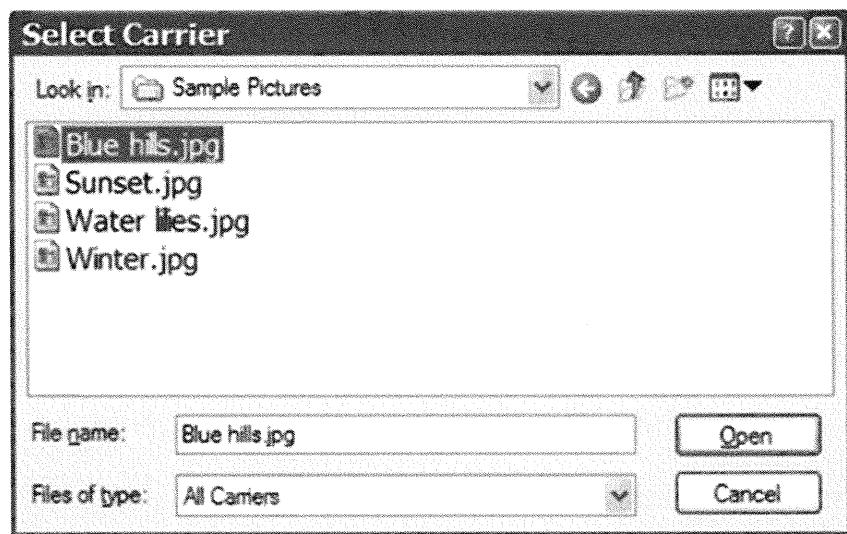
7. After the Select the files you want to hide window displays again, click *Next* to continue.



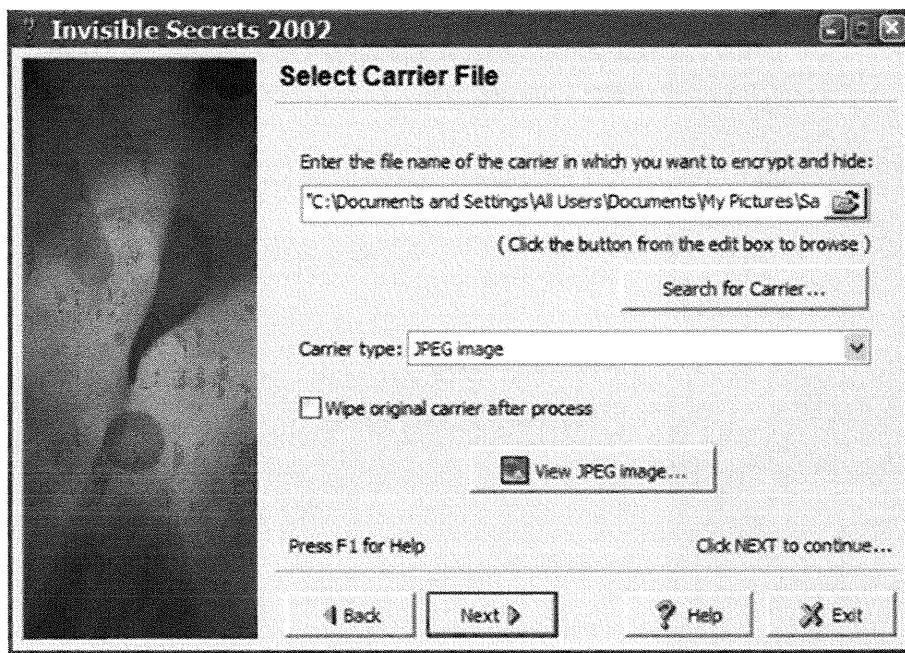
8. In the Select Carrier File window, you are prompted to select the carrier file that you will use to hide data in.



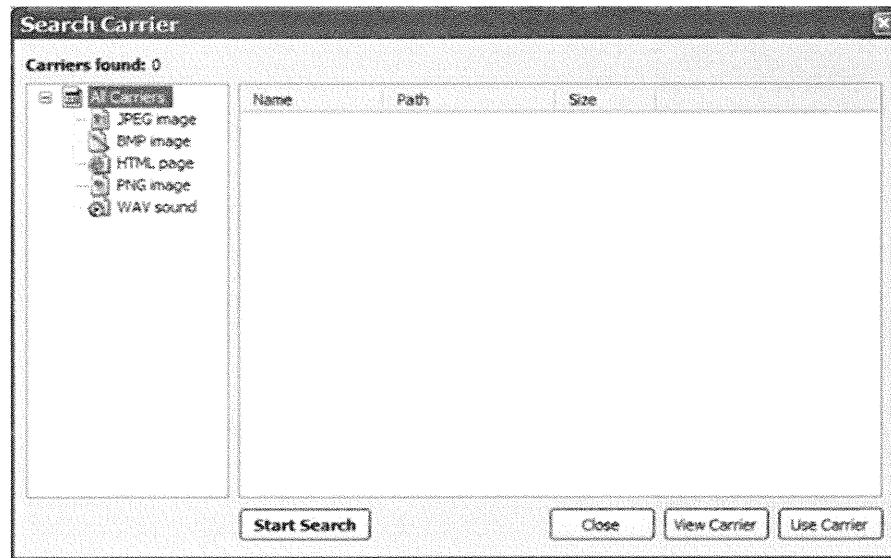
9. Click the open folder image to the right of the field, as shown in the next screen. Select a jpg file from your system and click *Open*. You can search your hard drive for sample .jpg files that are included with the install.



10. The file Blue Hills.jpg is selected as the carrier file. (You can use any .jpg that you would like.) You are given the option to wipe the original carrier after the encryption and hide process finishes. Do NOT select this option.



11. If you don't know where a good carrier file is located, click *Search for Carrier*. Invisible Secrets provides you with an interface to search for a carrier by file type. After you select a file, you can view it by clicking *View Carrier*, and if you decide to use it, click *Use Carrier*.



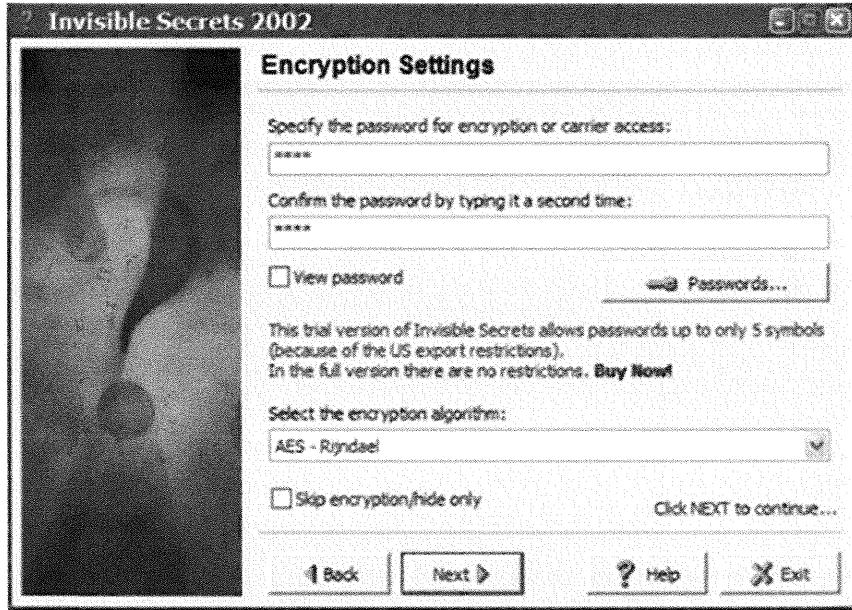
12. In the Search Carrier window, click *Close*, and then in the Select Carrier File window, click *Next*.

The next window prompts you for the password and the algorithm to use when encrypting the file. The algorithms that are natively supported include:

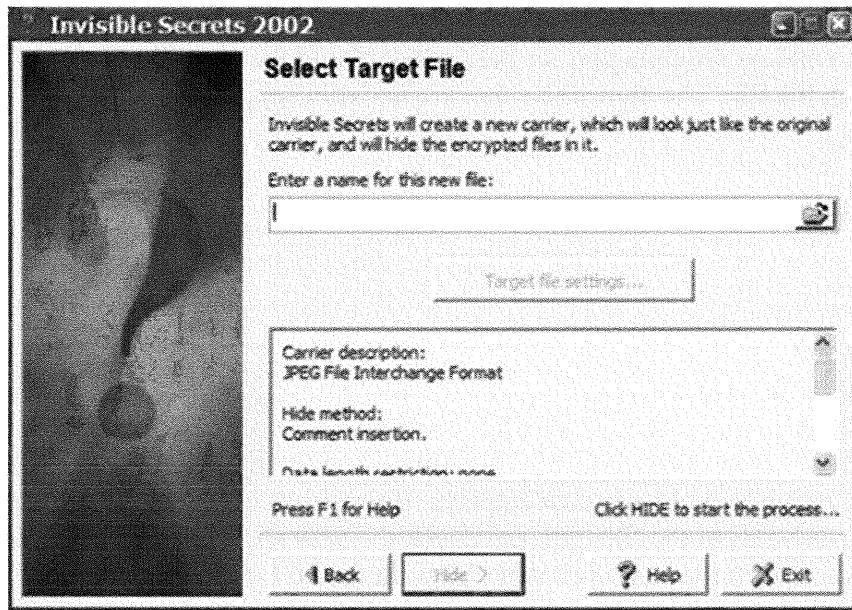
- AES
- Twofish
- RC4
- CAST
- GOST
- Diamond 2
- Sapphire II
- Blowfish

13. The Encryption Settings window displays. If you want to hide a file but do not want to assign a password to the file, you can check the *Skip encryption/hide* box at this time. Enter and confirm the password that you want to use for the newly created file. After you are done, click *Next*.

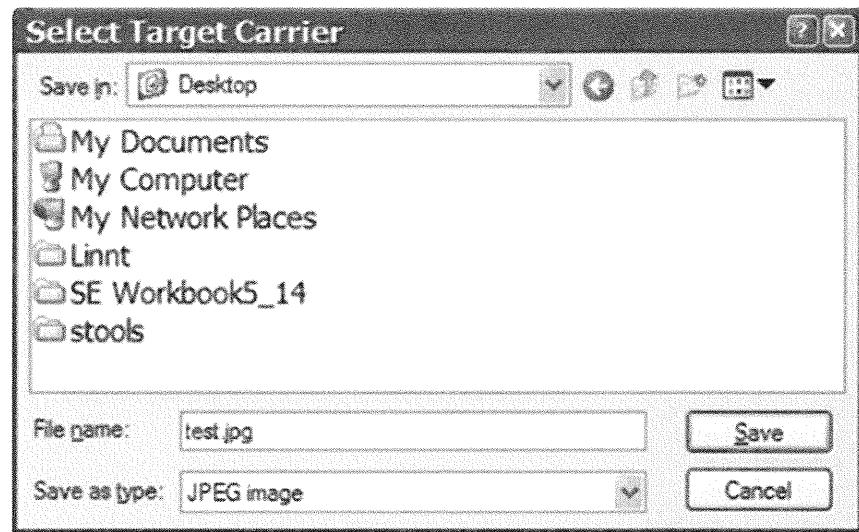
Note: The demo version of Invisible Secrets limits you to five symbols due to U.S. export restrictions.



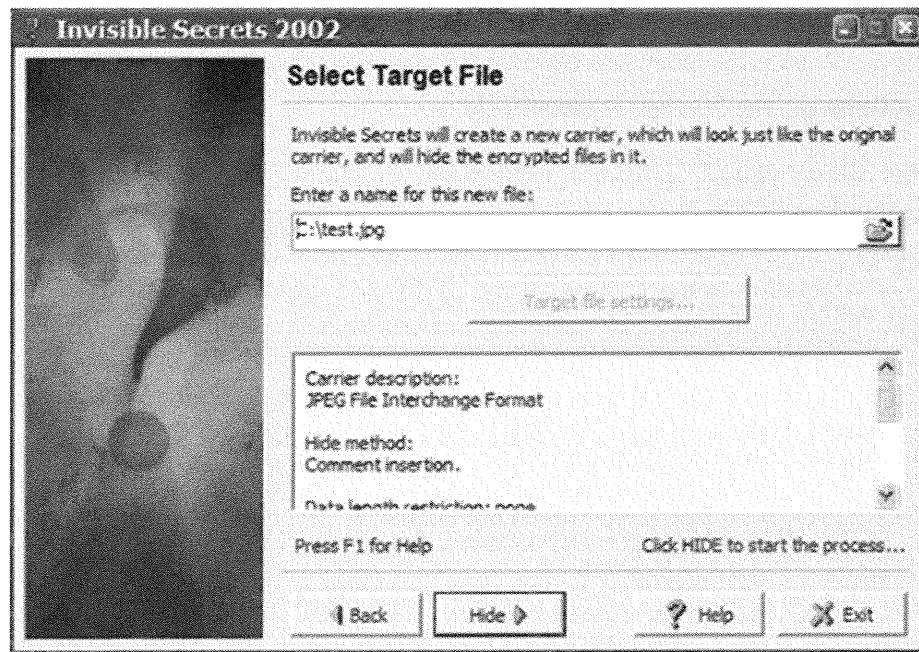
14. In the Select Target File window, enter the name that you want to use for this new file. To do this, click the *open folder* button located to the right of the field.



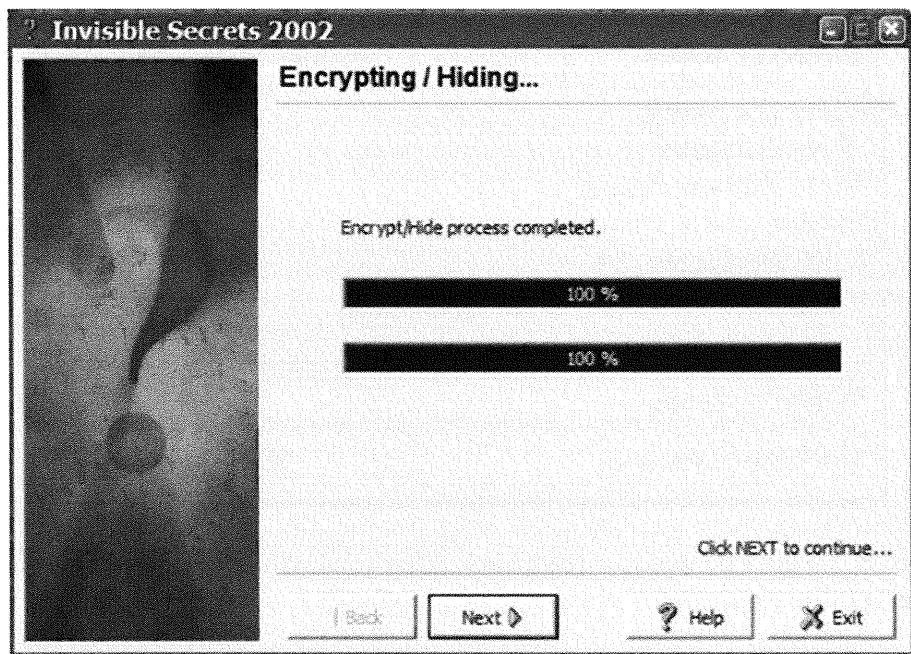
15. In the Select Target Carrier window, enter **test.jpg** as the name of the target carrier, save it to **C:**, and then click *Save*.



16. The Select Target File window reappears. After reviewing the location and filename of your target file, click *Hide*.



17. The Encrypting / Hiding window displays. The file is now encrypted whereas the Secret_Data.txt file is hidden within your new target carrier. After this process is done, click *Next*.

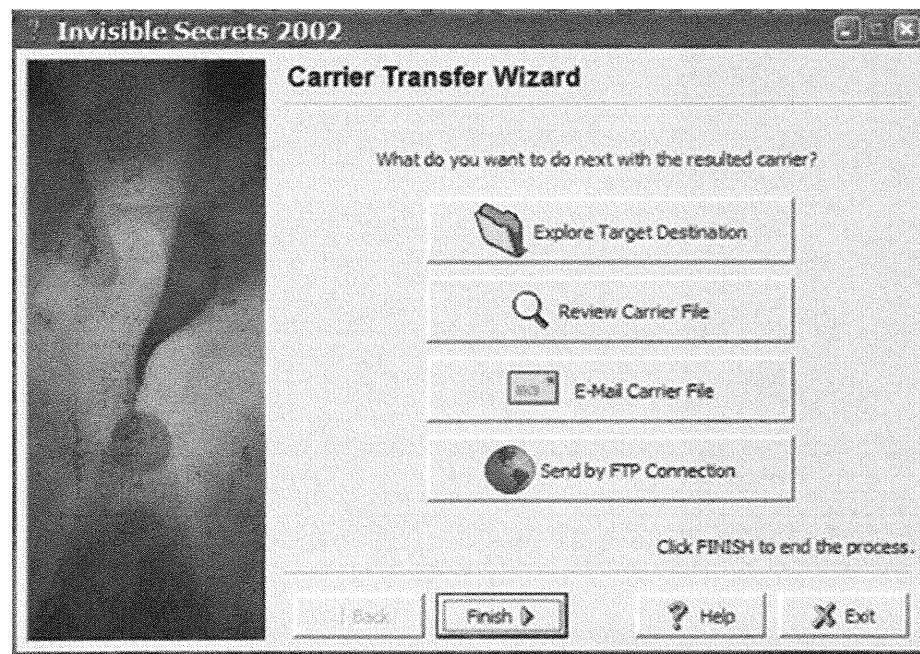


Performing Actions on the Newly Created File

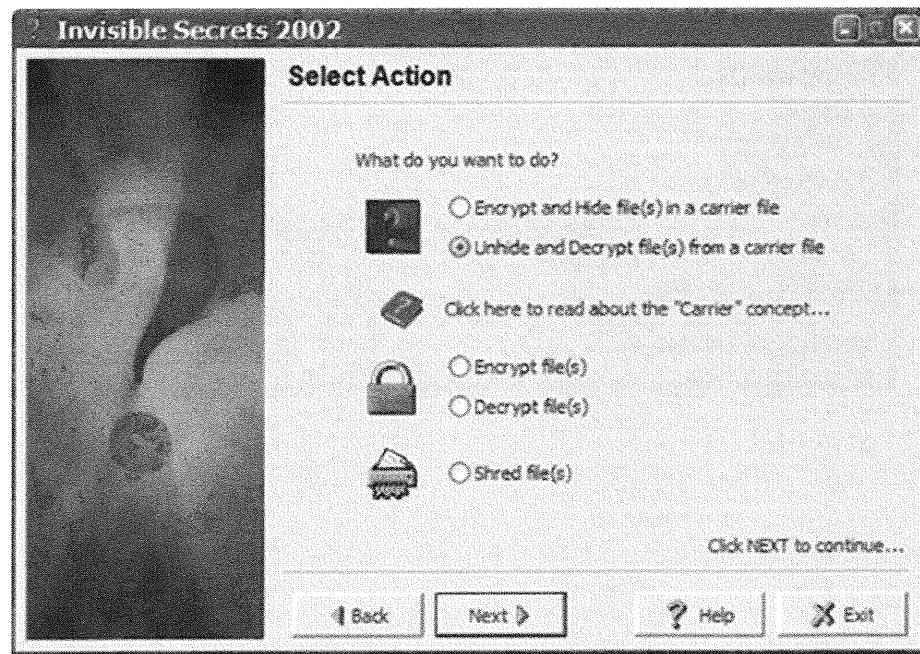
Invisible Secrets allows you to perform a number of steps on the newly created file, including sending it via e-mail and FTP.

Follow these steps to perform actions on the newly created file:

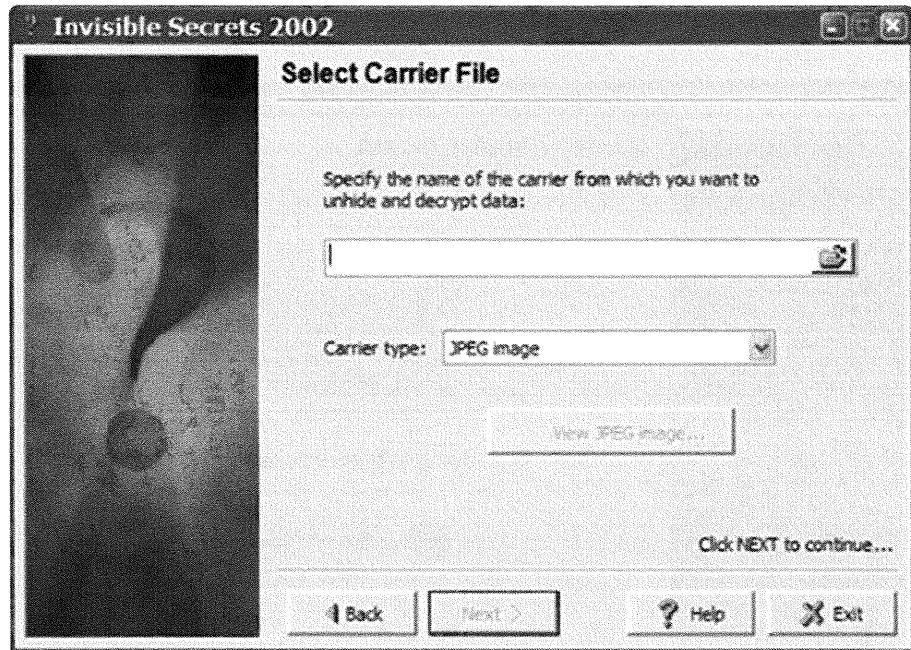
1. In the Carrier Transfer Wizard window, click *Exit* to close Invisible Secrets. You can examine the newly created file in C:\.



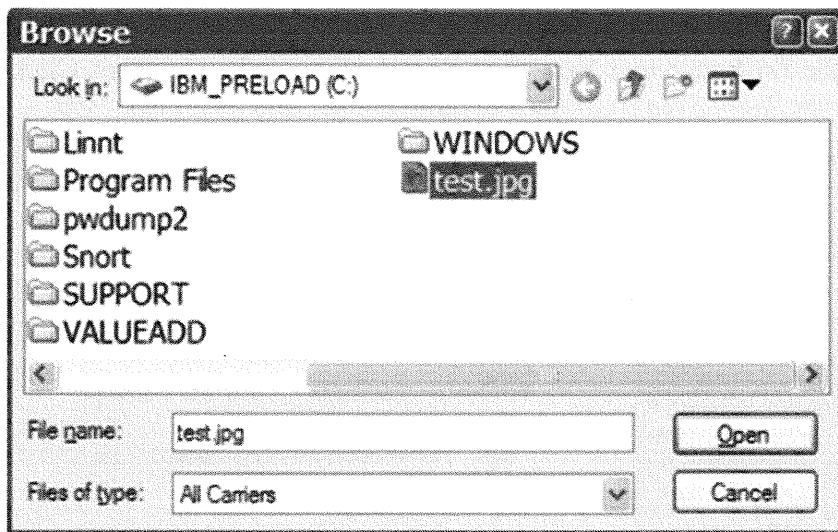
2. To reveal the information contained within c:\test.jpg, start *Invisible Secrets*. After clicking the *Try* button and *Next*, the Select Action window appears. Click the *Unhide and Decrypt file(s) from a carrier file* radio button, and then click *Next*.



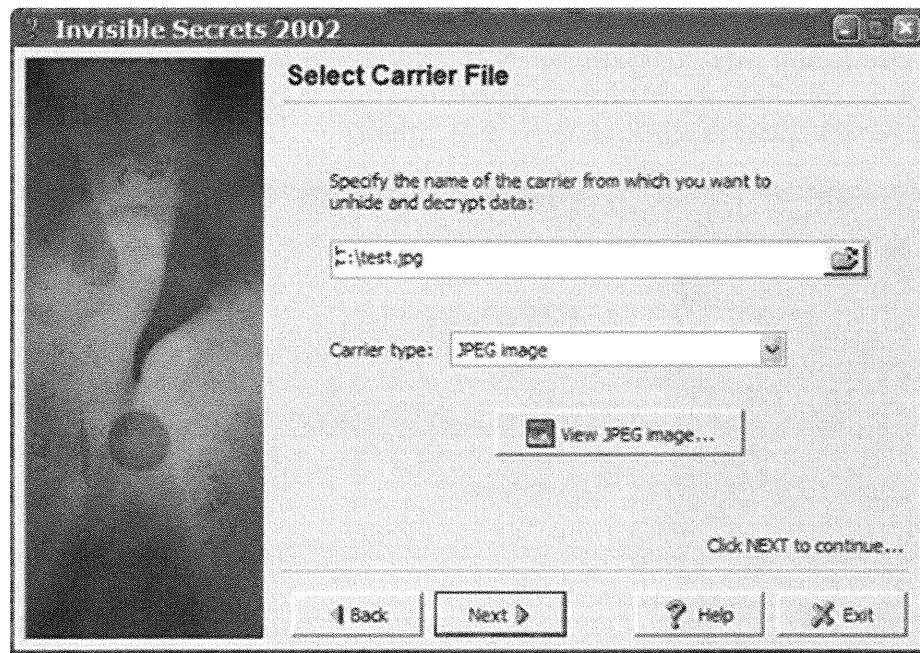
3. The Select Carrier File window displays. To choose the carrier file, click the *open folder* button to the right of the field.



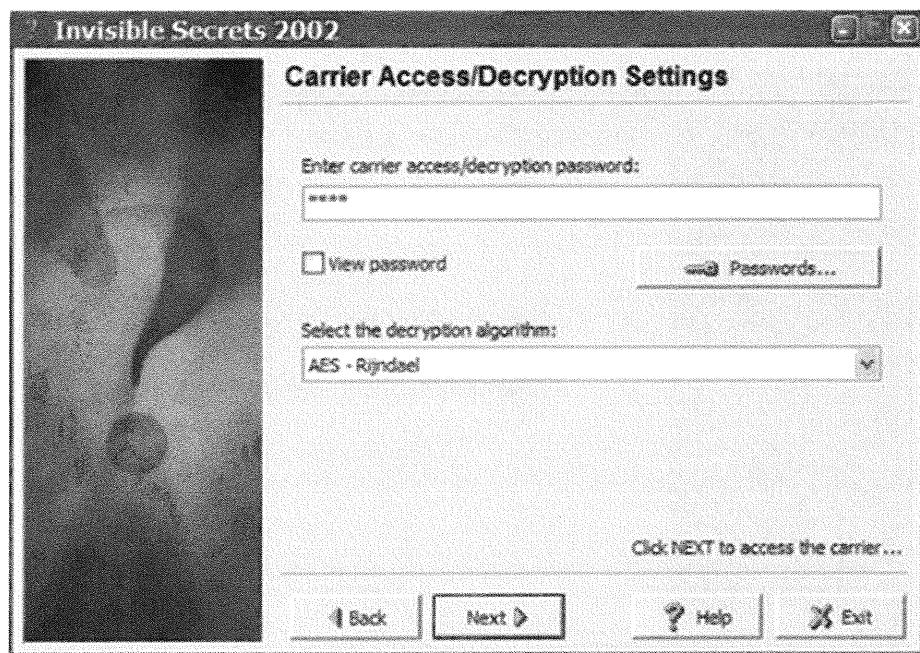
4. Click *c:\test.jpg* and click *Open*.



5. After the Select Carrier File window reappears, click *Next*.

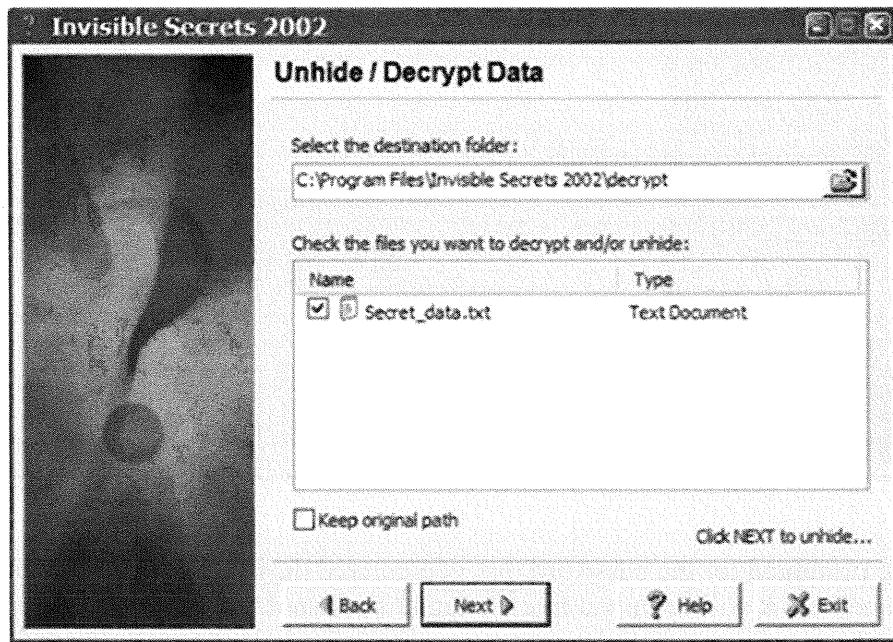


6. You are prompted for the password that you entered during the encrypt/hide phase. Remember that it will be only a five-character password due to the demo restrictions.
7. In the Carrier Access/Decryption Settings window, enter the passphrase you selected and click *Next*.

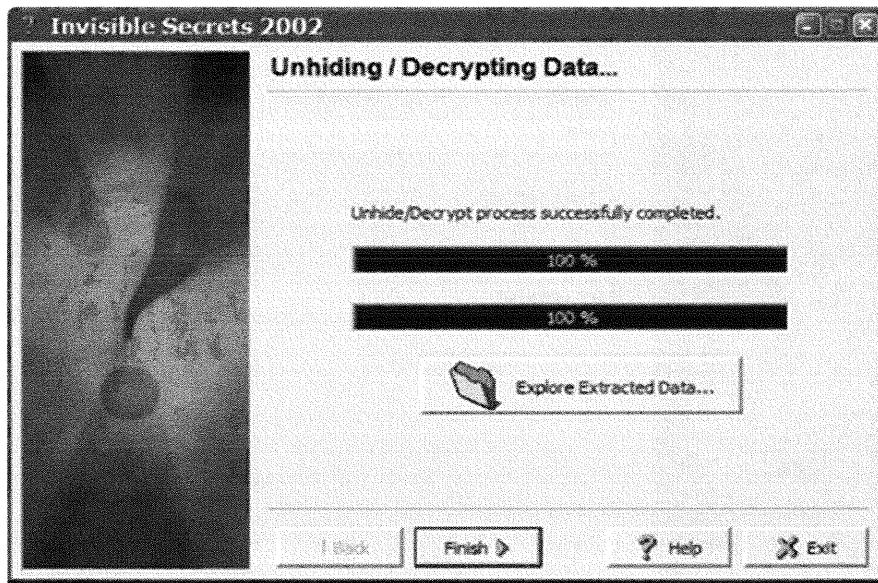


8. The Unhide/Decrypt Data window shows your original text file. Click *Next* to extract the file.

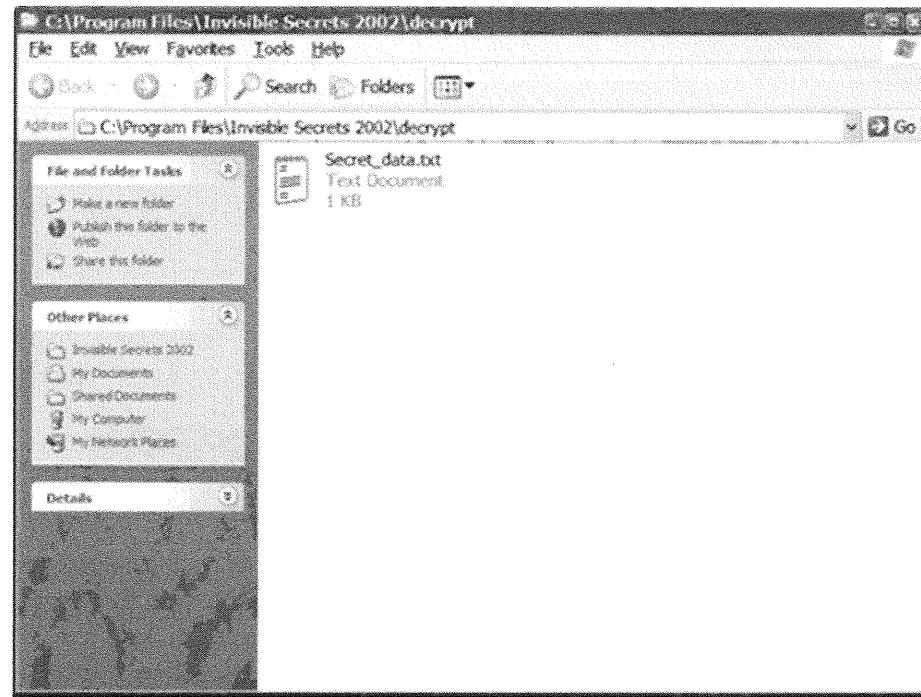
Note: The default location to decrypt files is the C:\Program Files subdirectory. If the data contained within the file is in fact sensitive, you might want to change the location to a more suitable place, such as an encrypted volume.



9. The Unhiding / Decrypting window displays. After the process is finished, you can explore the data by clicking the *Explore Extracted Data* button, as shown in the following screen.



10. Invisible Secrets starts Windows Explorer and starts at the location that you selected.



11. Invisible Secrets also adds a right-click option in Windows Explorer to hide and encrypt files. Simply right-click a file that you want to hide or encrypt, and click *Invisible Secrets* and then the option you want to perform. If you click *Hide*, the Invisible Secrets Wizard automatically starts at the prompt for the carrier file.

Wireshark

Wireshark is a sniffer/protocol analyzer that runs on a variety of platforms.

SANS Security Essentials – © 2016 Secure Anchor Consulting LLC

Wireshark

To know what is happening on a network, you need access to a network protocol analyzer such as Wireshark. Part of Wireshark's popularity is its capability to run on so many platforms. Wireshark is a powerful program with a lot of analysis capability built in. Let's examine this in more detail.

Wireshark Details

- Name: Wireshark
- Operating system: Windows, Unix
- License: Open source, GPL
- Category: Sniffer
- Description: Wireshark is a powerful sniffer with thousands of protocol dissectors. Wireshark includes features to assess VoIP traffic.
- URL: <http://www.wireshark.org>

SANS Security Essentials – © 2016 Secure Anchor Consulting LLC

Wireshark Details

The following topics and action items are covered in this chapter:

- Installing Wireshark
- Installing WinPcap
- Running Wireshark
- Analyzing traffic with Wireshark
- Analyzing VOIP traffic with Wireshark

In this chapter, you install Wireshark on Microsoft Windows 8. Wireshark is also installed with Kali Linux and can be run side by side with Tcpcdump.

Wireshark Background

- What is happening on your network is critical from a troubleshooting and security standpoint
- Analyzing IP, ICMP, TCP, and UDP headers helps but interpreting the various higher level protocols is also critical
- Protocol analyzers gives you a unique view into your network

SANS Security Essentials – © 2016 Secure Anchor Consulting LLC

Wireshark Background

Sniffers like Tcpdump only perform analysis of the layer 3 and layer 4 protocols; they perform minimal analysis of the application-level protocols. Because you can see the exact application data with Wireshark, you get a more detailed view of what is happening on your network.

Wireshark's Purpose

- Analyzes traffic to troubleshoot problems
- Aids in making analysis more straight-forward:
 - Does a lot of interpretation up front
 - Easy to use GUI
 - Powerful filtering capability

SANS Security Awareness - © 2010 Secure Anchor Consulting LLC

Wireshark's Purpose

This section intentionally left blank.

Wireshark Installation

- To install Wireshark, in the Wireshark directory, double-click the *wireshark-win64-1.4.2.exe* file
- If you are using a 32-bit platform, you should use *wireshark-win32-1.4.2.exe*

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

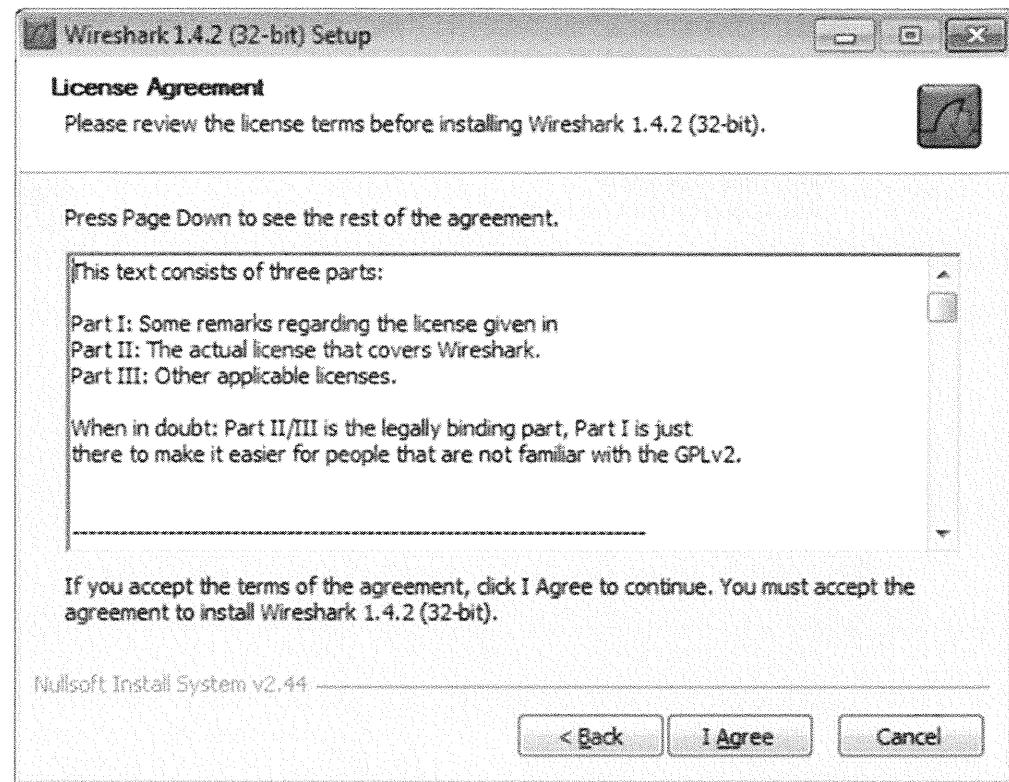
Wireshark Installation

To install and run Wireshark, perform the following steps:

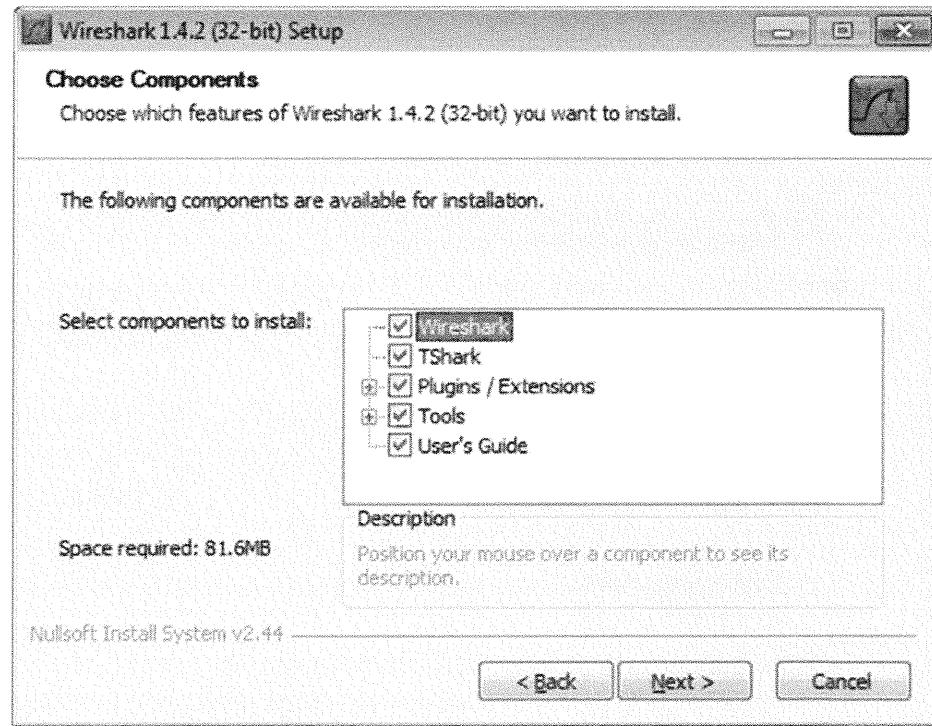
1. From the Wireshark directory on the CD, double-click the *Wireshark-setup-1.4.2.exe* file to begin the installation. In the Wireshark installation window, click *Next*.



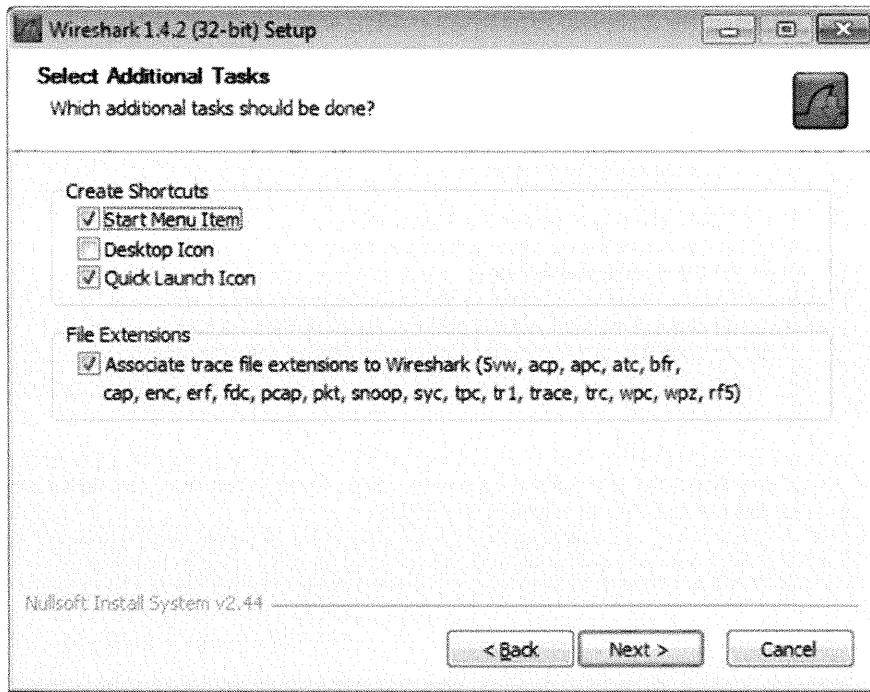
2. After the license agreement information displays, carefully read it to ensure you agree with the terms, and then click *I Agree*.



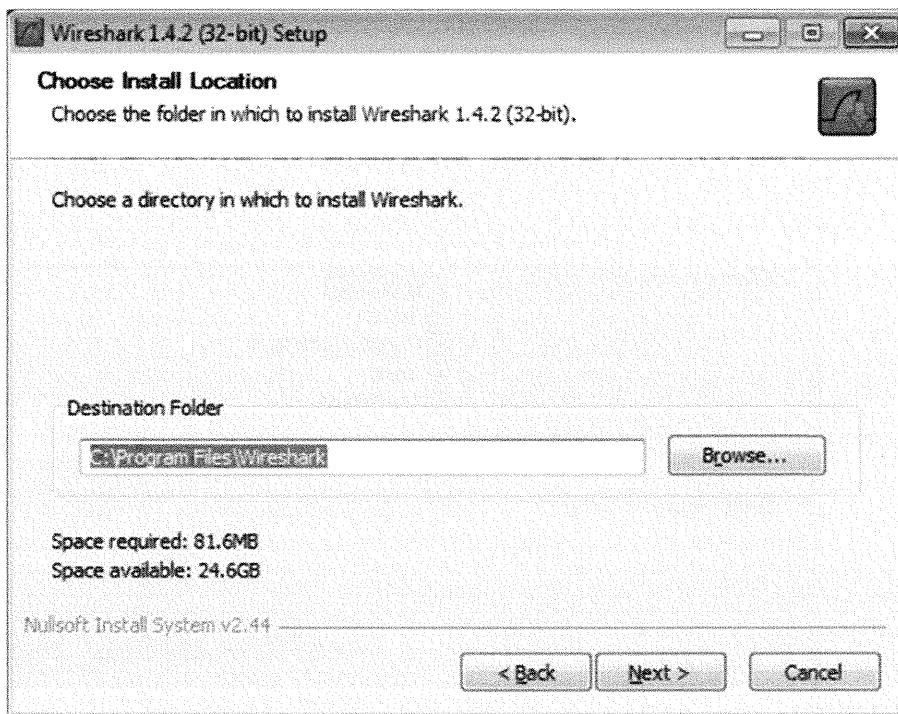
3. Accept the default installation components and click *Next*.



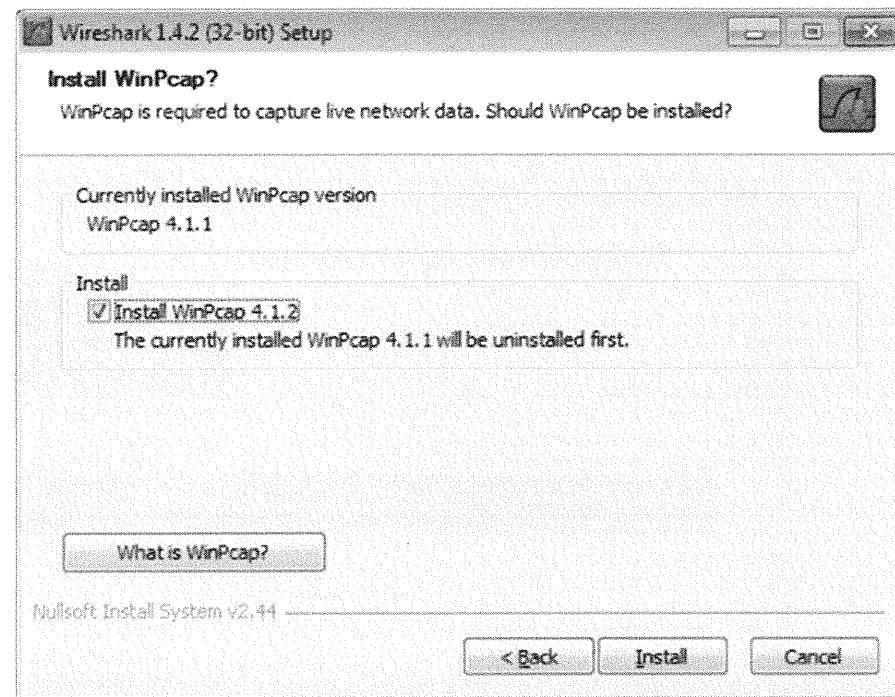
4. Accept the default additional setup tasks including the association of packet capture file extensions. Change the desired shortcuts, if desired, and click *Next*.



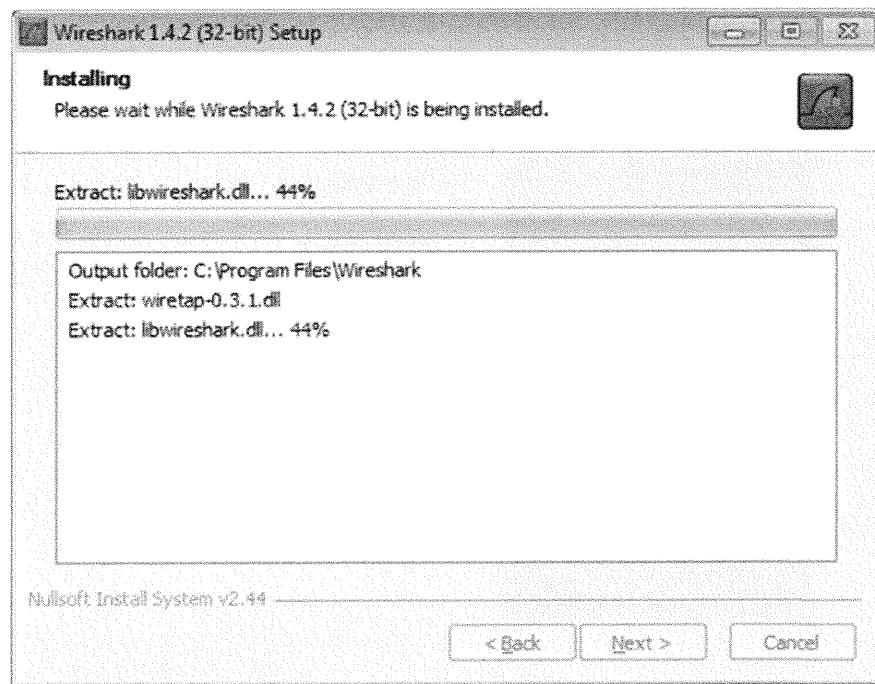
5. After the Choose Install Location window displays, accept the default 1 location and click *Next*.



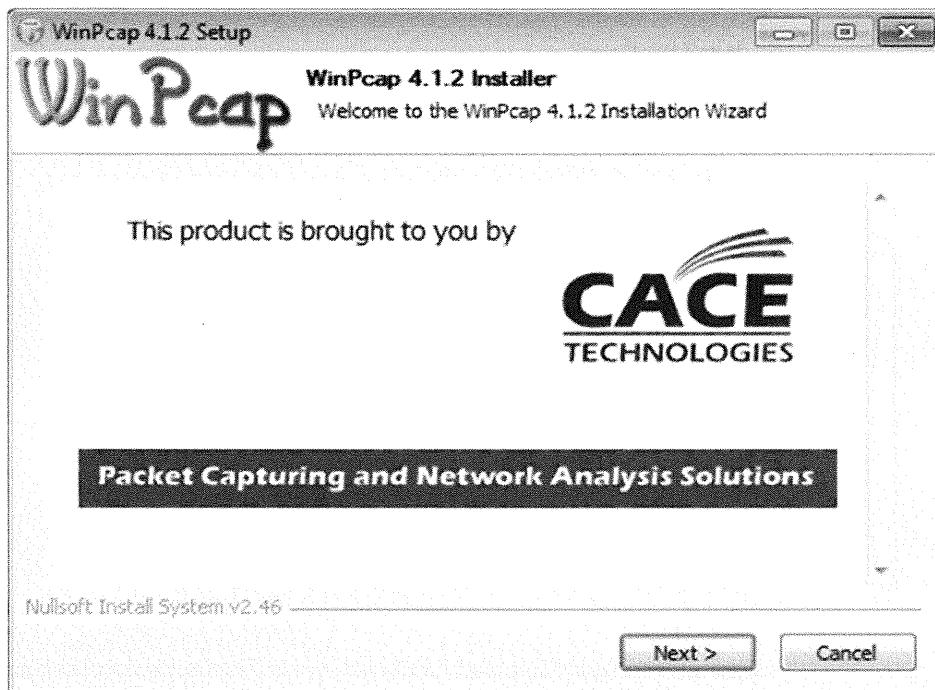
6. The Install WinPcap window displays. Wireshark requires a recent version of WinPcap to operate on Windows systems. Click *Install*.



7. After clicking *Install*, Wireshark begins copying files to the installation directory. Be patient as this process completes, because it might take several minutes.



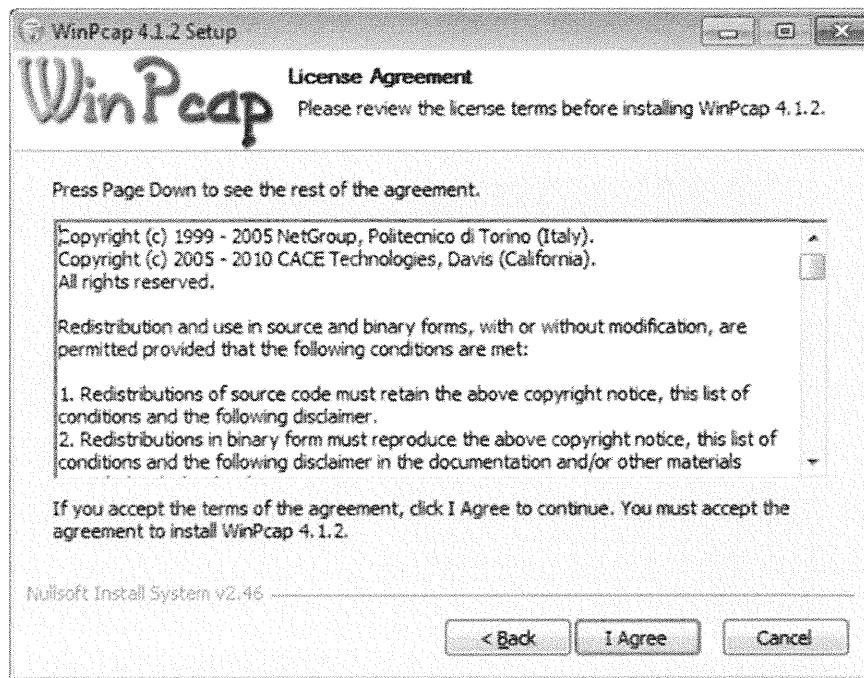
8. If you opted to install WinPcap as part of the installation process, Wireshark initiates the WinPcap installer process as shown in the following WinPcap Setup dialog box. Click *Next* to continue the WinPcap installation process.



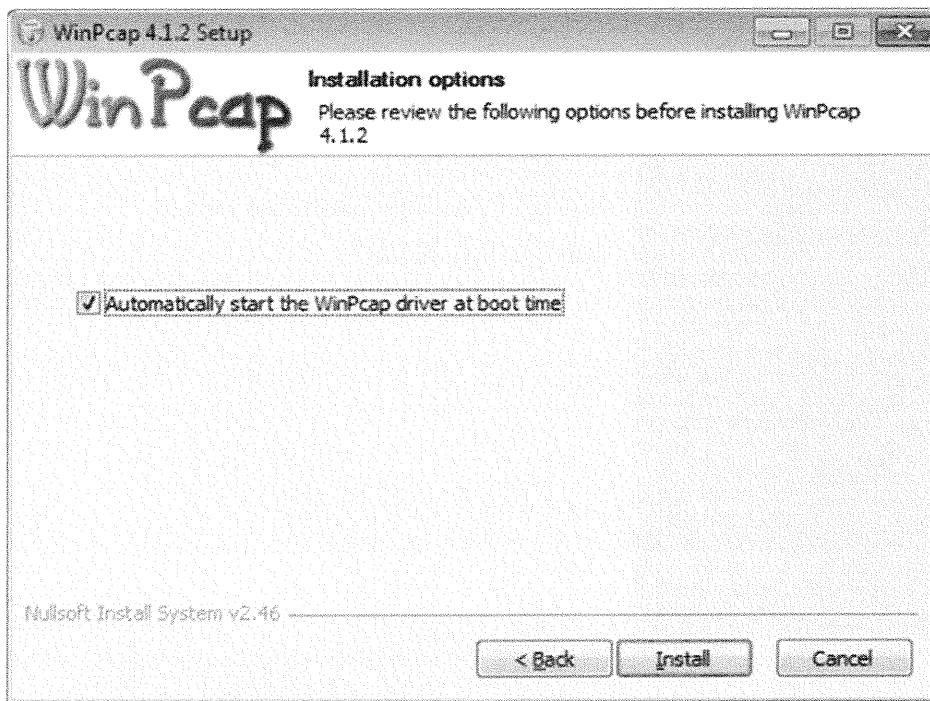
9. After the Welcome screen for WinPcap displays, click *Next* to continue.



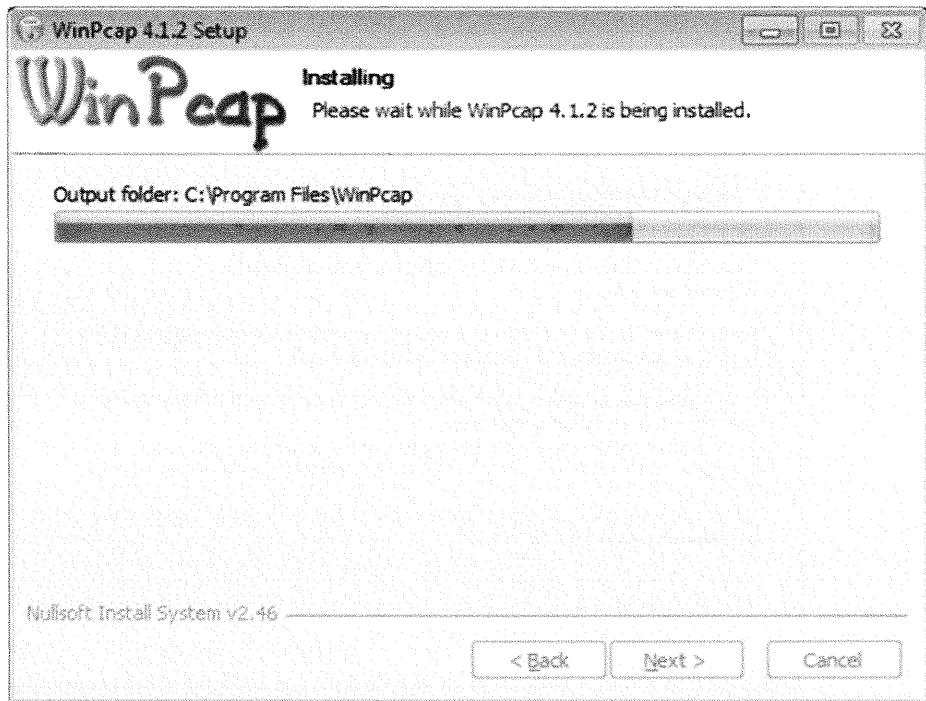
10. After the license agreement for WinPcap displays, carefully read it to ensure you agree with the terms, and then click *Next*.



11. Select whether you want WinPcap to start at system boot. Click *Install*.



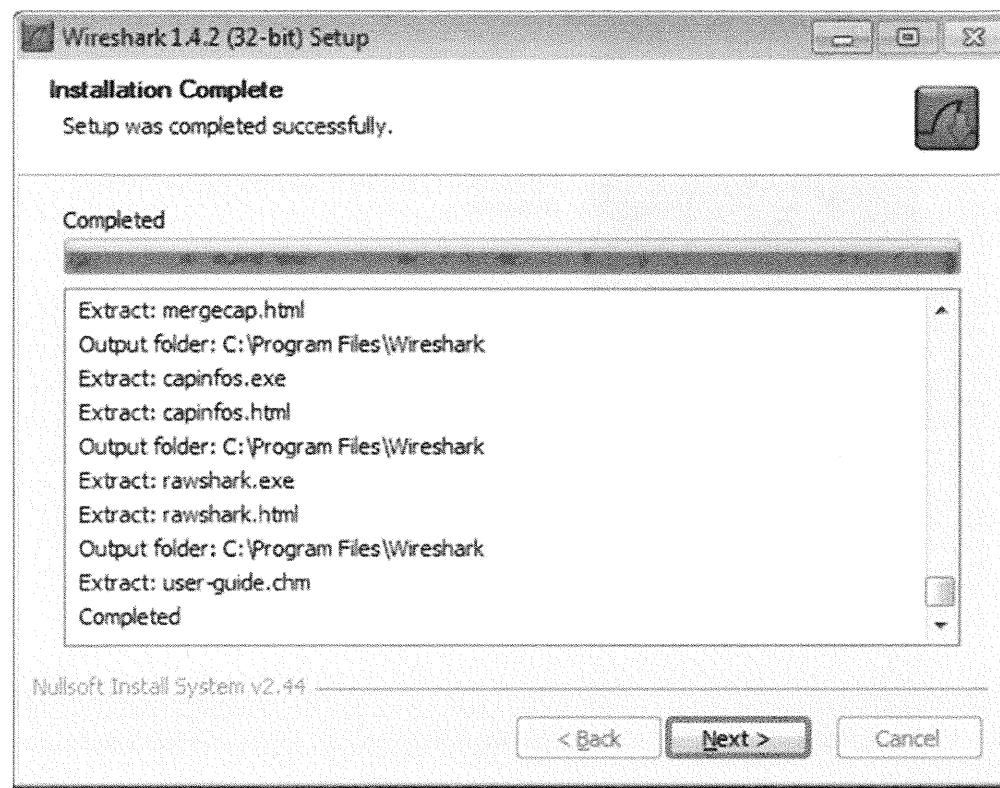
12. The installation wizard installs the WinPcap components and registers the necessary driver functions. Be patient while this process completes.



13. After the Completing the WinPcap 4.1 Setup Wizard screen displays, click *Finish*.



14. After Wireshark completes copying files, the Installation Complete dialog box displays. Click *Next*.



15. Complete the Wireshark installation by clicking *Finish*.



Running Wireshark (1)

- Wireshark can capture live traffic or read from capture files:
 - Libpcap files and other commercial sniffer captures
 - Allows Wireshark to interoperate with commercial tools
- Live sniffing requires special access, or local system and admin privileges

SANS Security Essentials - © 2016 Secure Analytics Consulting LLC

Running Wireshark (1)

Wireshark can examine packet traces from a live packet capture on a physical network, or by reading stored packet captures. Supported stored packet capture types include libpcap files as well as packet captures from a variety of commercial sniffers including NAI Sniffer, EtherPeek, AiroPeek NX, Solaris snoop, and many others.

On a local system, administrator privileges are needed to capture packets from a live interface. No special privileges are needed to use Wireshark with a stored packet capture file.

Running Wireshark (2)

- Start Wireshark from the GUI:
 - Three main windows:
 - Packet capture
 - Tree view of packet
 - Data from packet



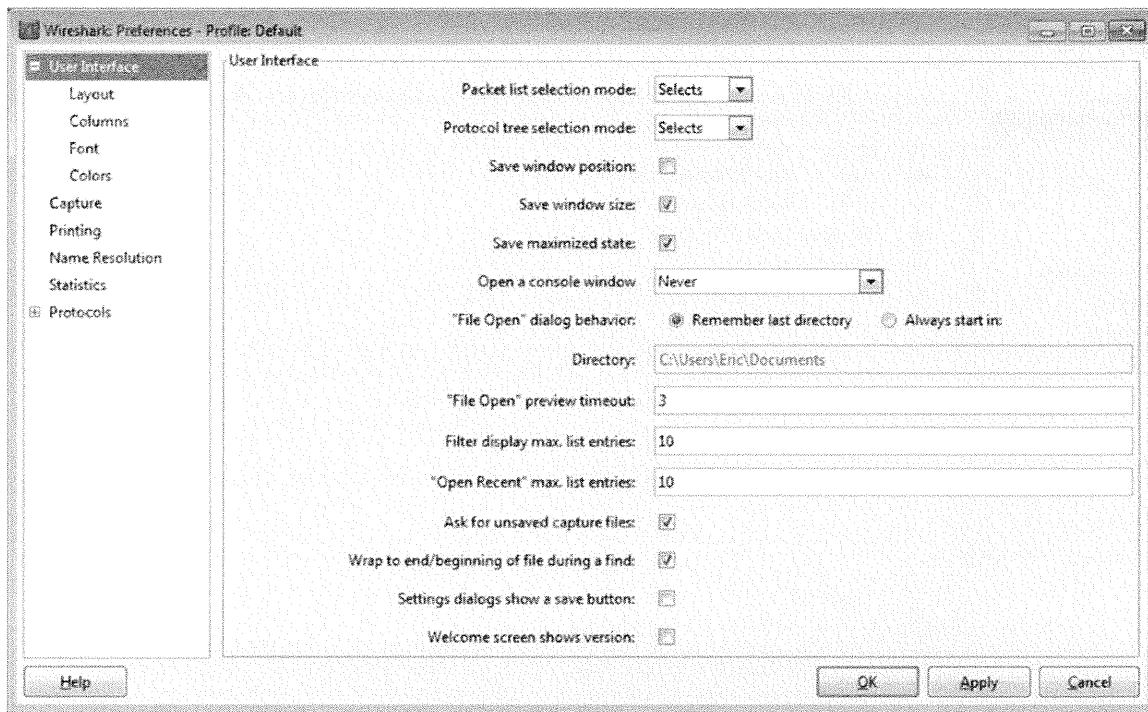
Running Wireshark (2)

Now that Wireshark and WinPcap are installed, you can start Wireshark.

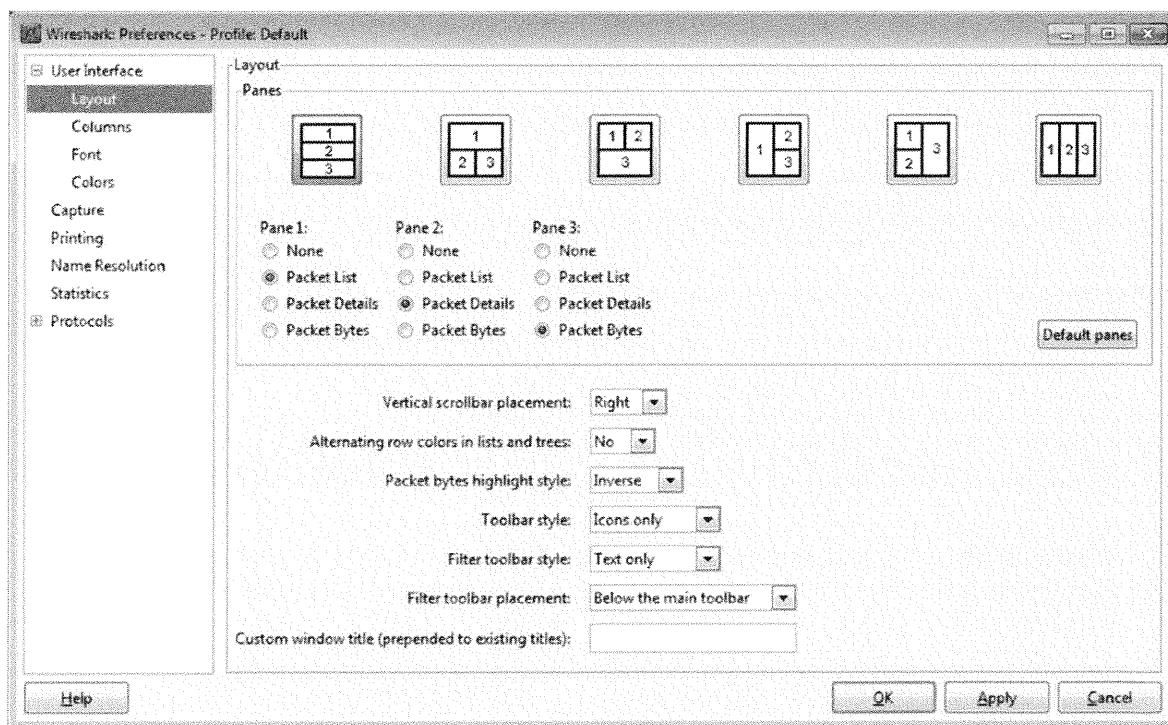
1. To do so, click *Start, All Programs, Wireshark*, and *Wireshark*. The Wireshark Network Analyzer window displays.



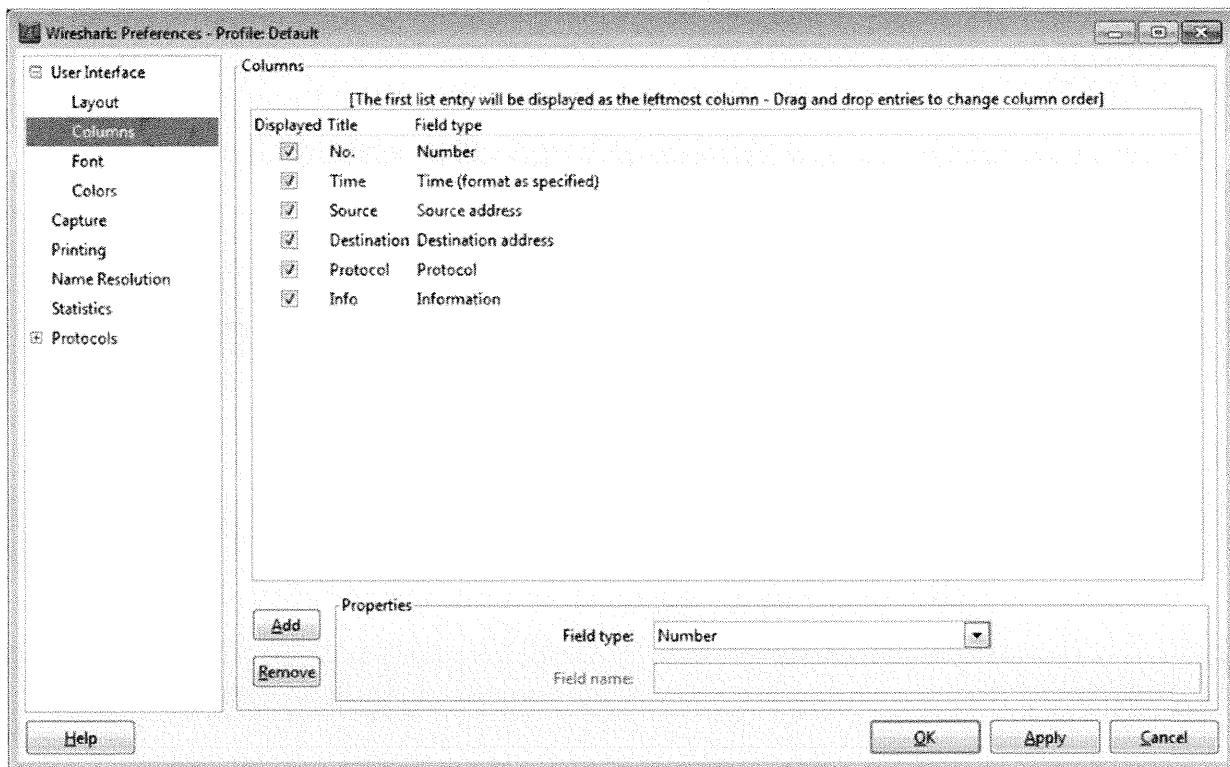
- To open the Preferences window, click *Edit, Preferences*. From this window, you can select the preferences to be used during packet captures. The first screen that opens is User Interface, which is where you control what the User Interface looks like.



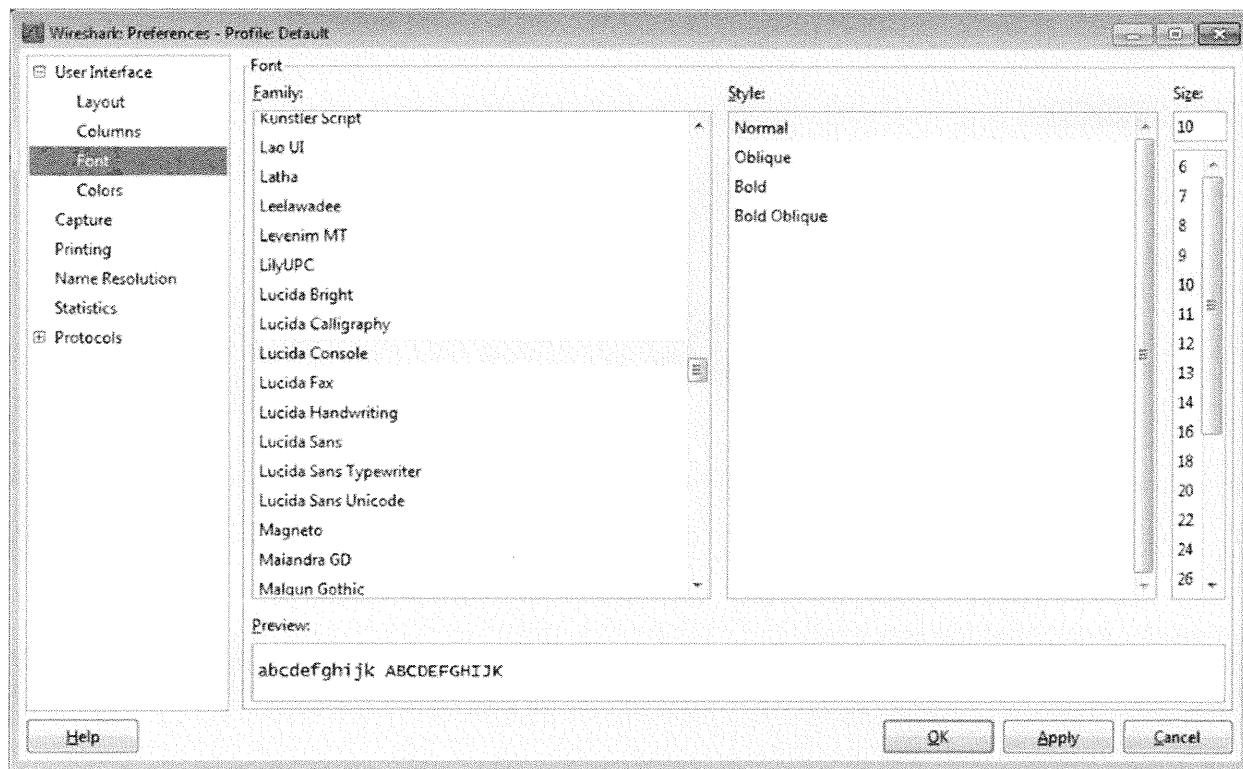
- Under User Interface, click *Layout*. Layout controls how the three main areas of Wireshark display on the screen.



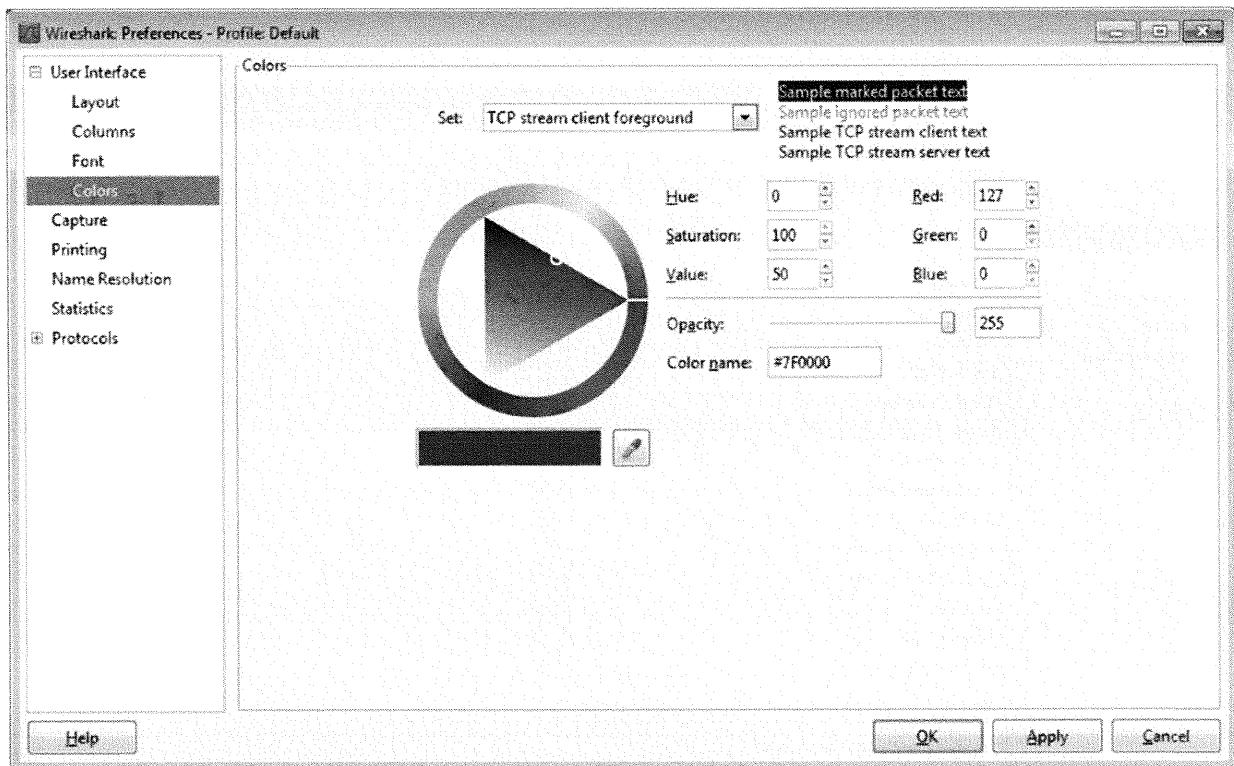
4. Under User Interface, click *Columns*. Columns controls what information is presented on the screen and allows you to customize the information presented. Experiment with adding new columns to the view.



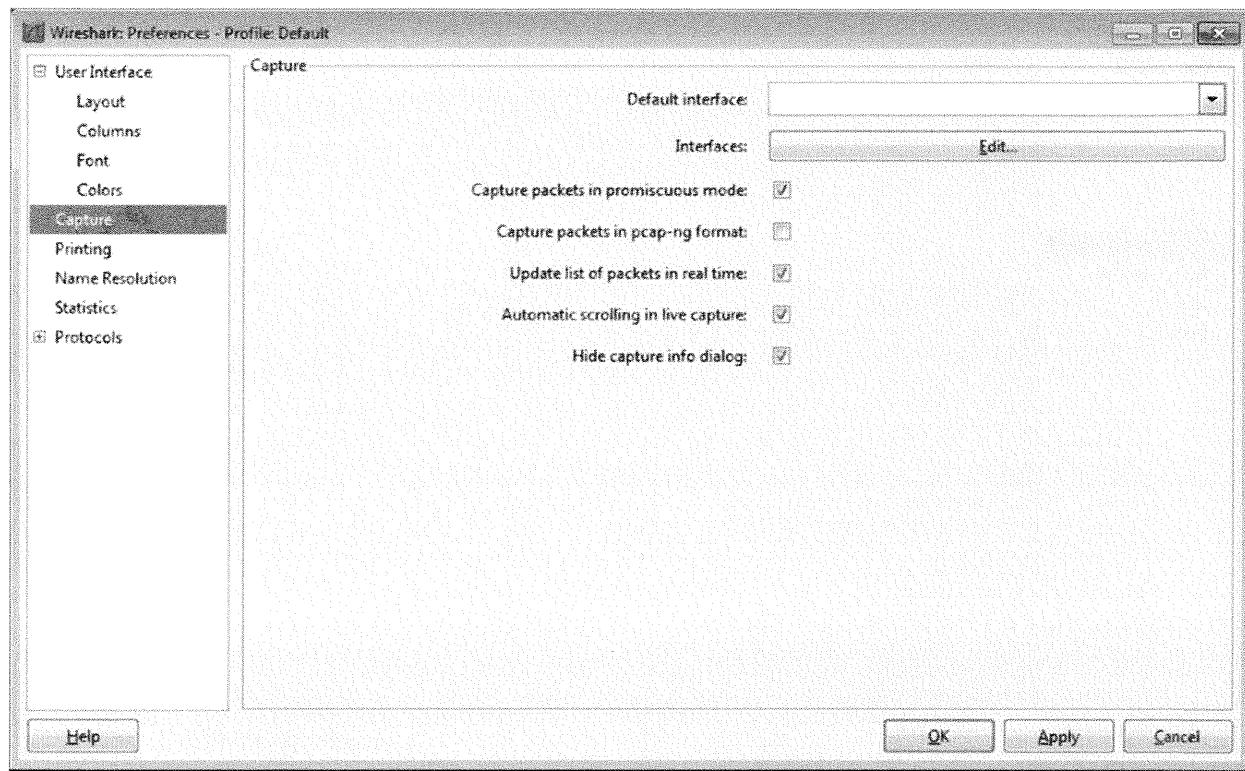
5. Under User Interface, click *Font*. Use this screen to customize the size and fonts used to display information.



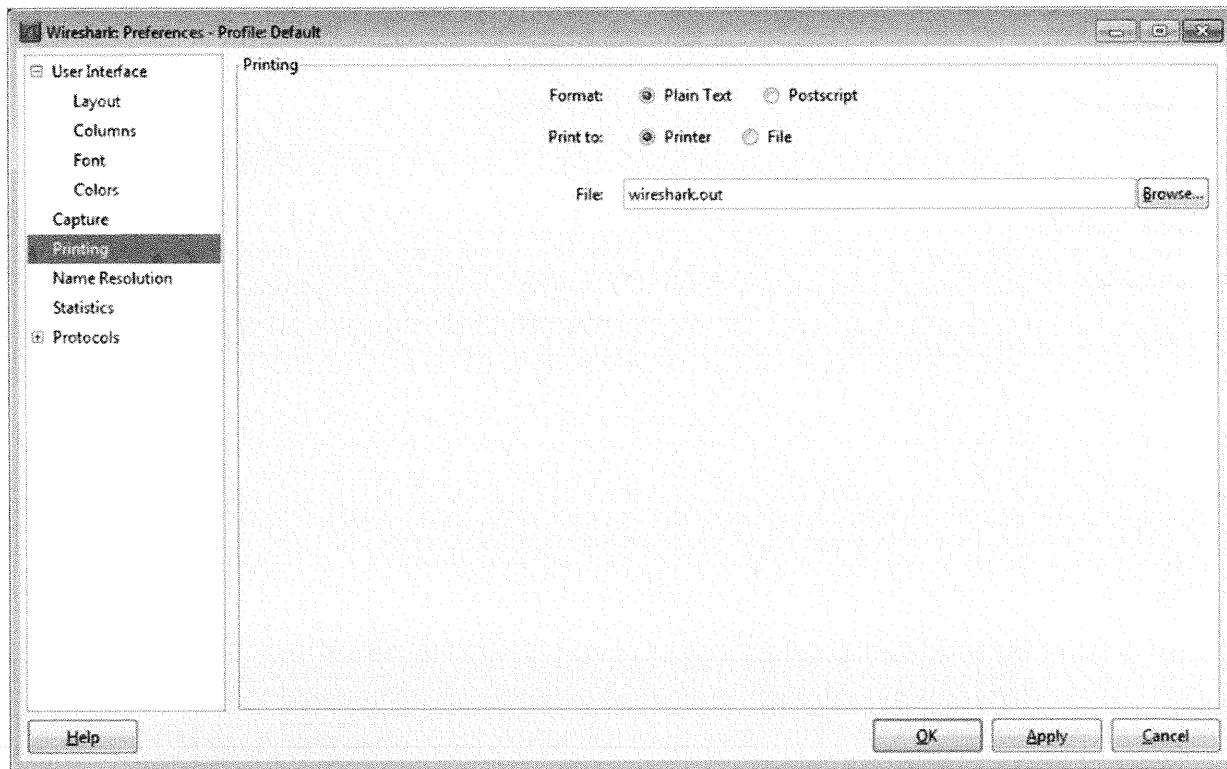
6. Under User Interface, click *Colors*, to customize the colors on the GUI.



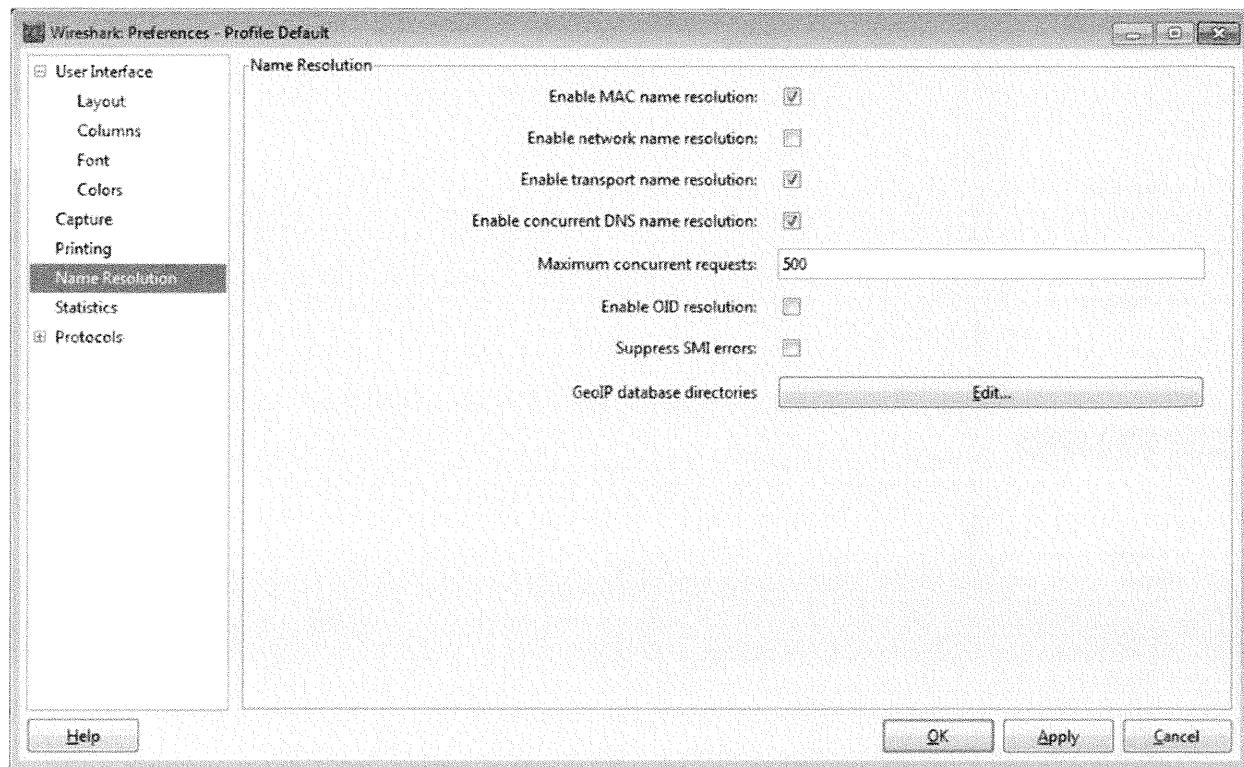
7. Next, click *Capture*, which controls how packets are captured and how they are updated during the capture.



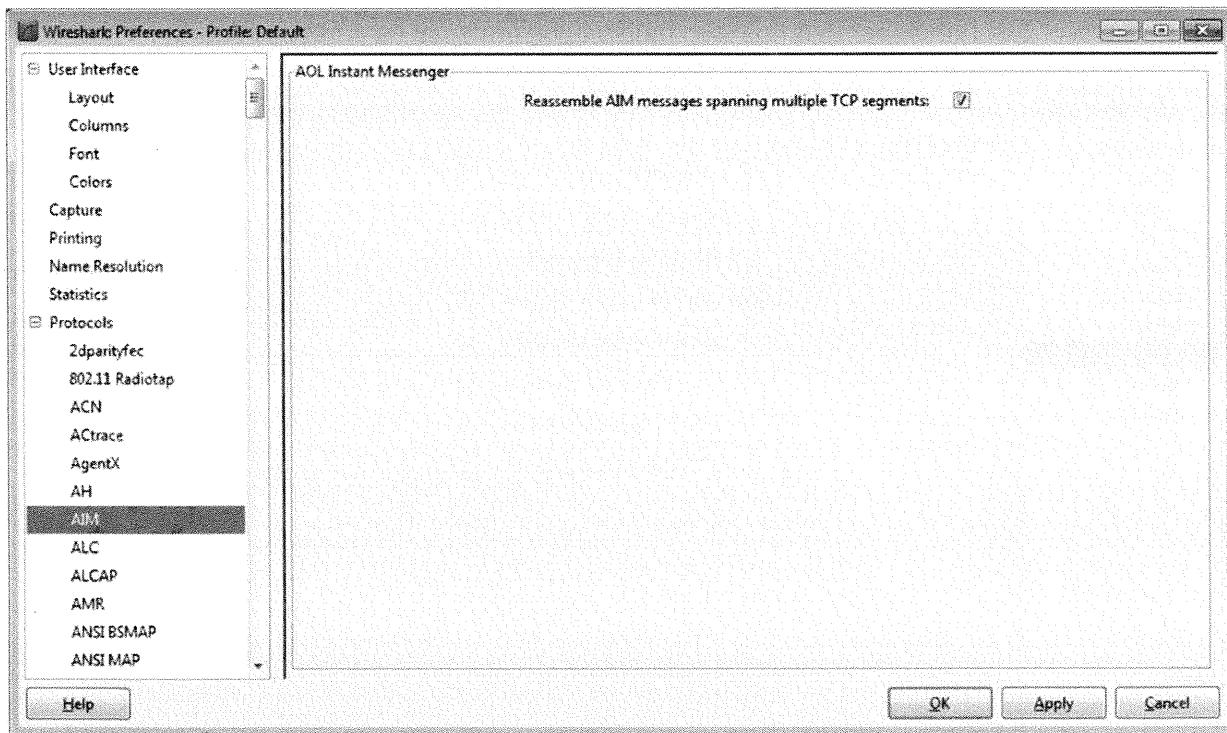
8. The next item you examine is Printing. You can click *Plain Text* or *Postscript* for the format and print either to a Printer or to a File. If you are printing to a file, you can specify the name of the file in this window, shown in the following screen.



9. Next, click *Name Resolution*, which controls how reverse look-up performs. If you prefer, instead of seeing IP addresses, they will be resolved to domain names. Depending on the purpose of the capture and how well you know your network, you might find it easier to have Wireshark attempt to resolve MAC addresses and IP addresses during a capture. Just remember that this step hinders performance on a busy network.

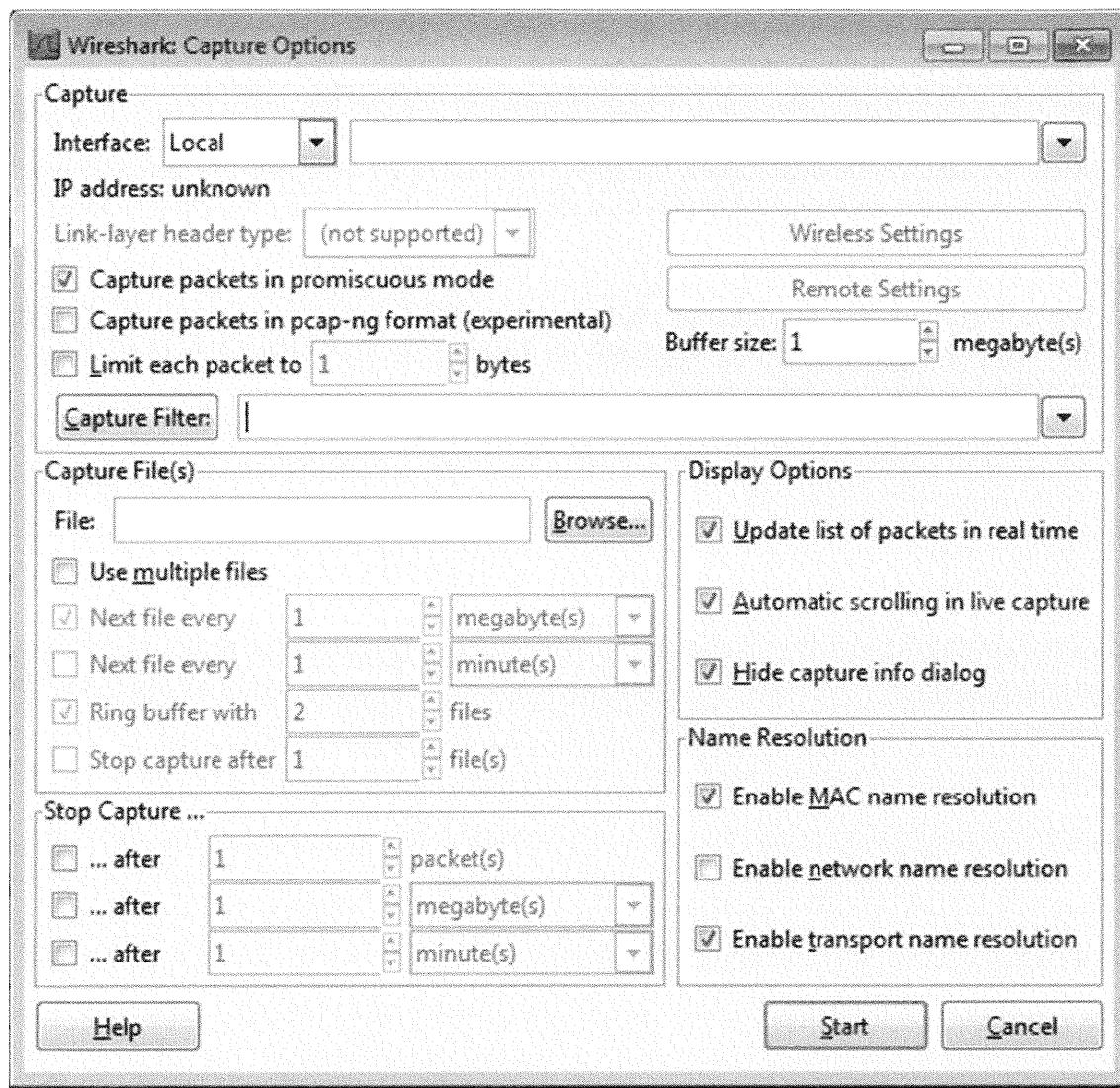


10. To view the preference for the different protocols, click the + symbol to the left of the Protocols item. This displays a list of protocols. By clicking a specific protocol, you can view the options. For example, click *AIM*.



11. After you are done selecting preferences, click *OK* to close the screen.

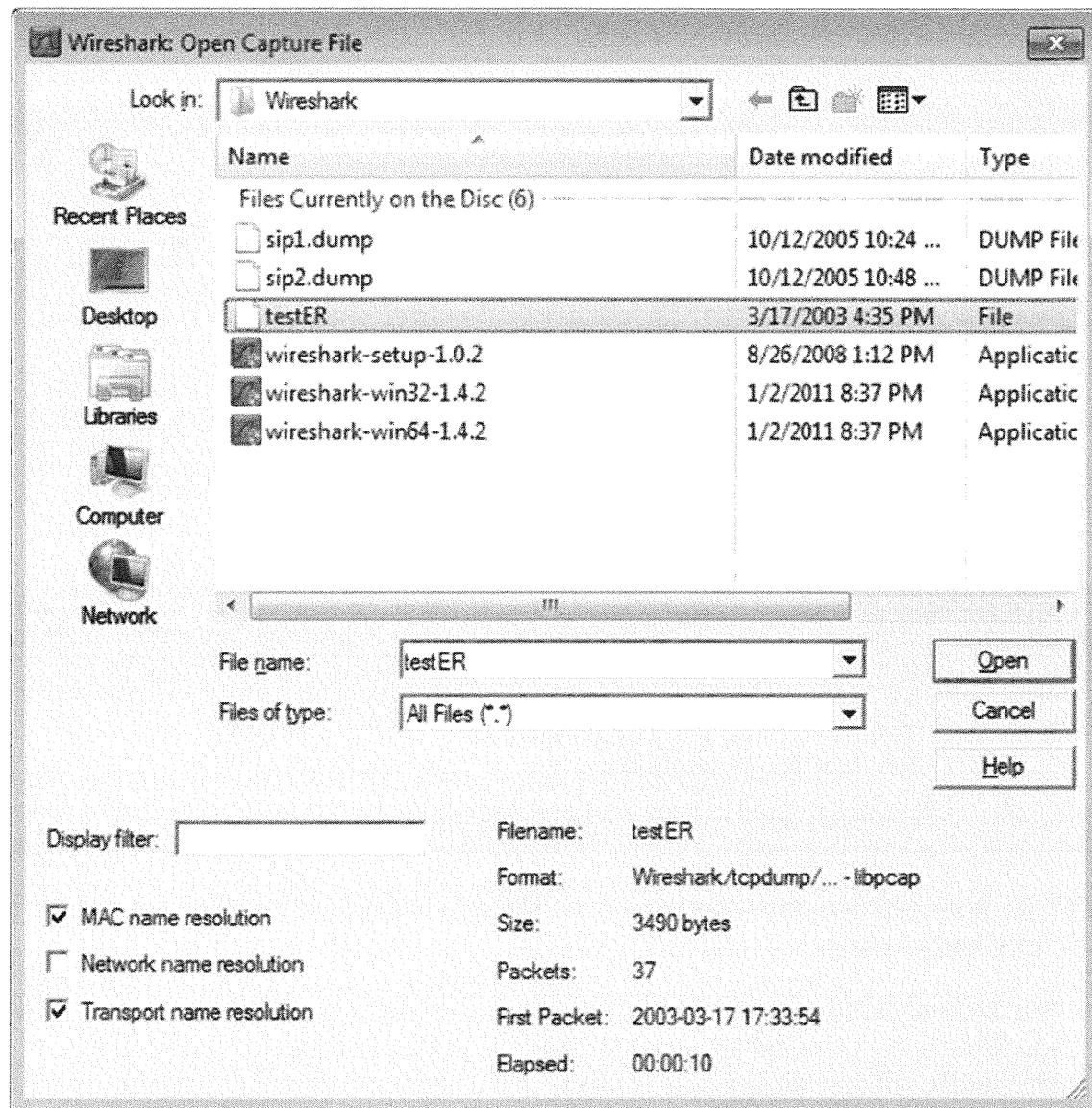
Because you might have several network adapters installed on your system, you can specify which adapter is used by default in the Capture menu, which can be seen by clicking the *Capture* menu and clicking *Options*.



You can also automatically put the chosen adapter into promiscuous mode (promiscuous mode allows the interface to receive all packets that it sees whether they are addressed to the interface or not) so you do not have to select it before each capture. The next option determines whether Wireshark updates the list of packets in real time or waits until the capture is complete. Enabling or disabling this option often depends on the reason for the capture. When troubleshooting, it is often useful to know immediately when the event takes place. You might use the final option with the Update list of packets in real time option. Automatic scrolling in live capture means that Wireshark keeps the most current information in the packet window. This is useful if you are watching traffic as it traverses the network, but it can also be a real pain if you are trying to gather more information about a certain packet during the capture.

1. To start a capture, click *Capture, Start*. You see a window capturing the files based on the preferences that you selected previously.
2. During the capture, Wireshark presents a window showing the total number of captured frames as well as the total number of frames specific to certain protocols. When finished capturing packets, click *Stop*. Unless you are connected to a hub, you will not receive any packets. For this exercise, that is okay because we do an offline analysis.
3. After the capture window closes, the captured packets are available for further inspection. You can see the various packet types sorted by when they occurred. In this example, you do not see any packets.
4. Note: Wireshark also provides you with an interface to analyze packets from a Tcpdump file. You can open an existing Tcpdump file by clicking *File, Open*. Navigate to the existing file, click it, and click *OK*.

For this exercise, in your Wireshark directory, open the file from your CD called *testER*. Click *File*, *Open*. From the window, navigate to *Wireshark directory*, click the *testER* file, as shown in the following screen, and then click *OK*.



The main screen displays with the data loaded as shown in the following screen.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	10.1.50.134	64.12.26.36	AIM	Keep Alive
2	2.113356	64.12.26.36	10.1.50.134	TCP	aol > darcorp-lm [ACK] Seq=1
3	3.5.618514	10.1.50.134	192.168.3.25	TCP	iapp > smtp [SYN] Seq=0 Win=
4	5.649444	192.168.3.25	10.1.50.134	TCP	smtp > iapp [SYN, ACK] Seq=0
5	5.649454	10.1.50.134	192.168.3.25	TCP	iapp > smtp [ACK] Seq=1 Ack=
6	5.655267	192.168.3.25	10.1.50.134	SMTP	S: 220 mail.iwc.sytexinc.com
7	5.818559	10.1.50.134	192.168.3.25	TCP	iapp > smtp [ACK] Seq=1 Ack=
8	6.947343	10.1.50.134	192.168.3.25	SMTP	C: h
9	6.948274	192.168.3.25	10.1.50.134	TCP	smtp > iapp [ACK] Seq=90 Ack
10	7.032029	10.1.50.134	192.168.3.25	TCP	[TCP segment of a reassemble
11	7.032851	192.168.3.25	10.1.50.134	TCP	smtp > iapp [ACK] Seq=90 Ack
12	7.107549	Syskonne_98:d3:3c	AmbitMic_ca:86:b7	ARP	who has 10.1.50.134? Tell 1
13	7.107567	AmbitMic_ca:86:b7	Syskonne_98:d3:3c	ARP	10.1.50.134 is at 00:d0:59:c
14	7.200140	10.1.50.134	192.168.3.25	TCP	[TCP segment of a reassemble
15	7.201027	192.168.3.25	10.1.50.134	TCP	smtp > iapp [ACK] Seq=90 Ack

Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
 Ethernet II, Src: AmbitMic_ca:86:b7 (00:d0:59:ca:86:b7), Dst: Syskonne_98:d3:3c (00:00:5a:9
 Internet Protocol, Src: 10.1.50.134 (10.1.50.134), Dst: 64.12.26.36 (64.12.26.36)
 Transmission Control Protocol, Src Port: darcorp-lm (1679), Dst Port: aol (5190), Seq: 1, A
 AOL Instant Messenger

```

0000  00 00 5a 98 d3 3c 00 d0  59 ca 86 b7 08 00 45 00  ..Z..<.. Y.....E.
0010  00 2e c4 3f 40 00 80 06  9f d3 0a 01 32 86 40 0c  ...?@... ....2.@.
0020  1a 24 06 8f 14 46 b7 e8  10 e2 cc de 5e 37 50 18  .$.F.. ....^7P.
0030  43 06 3d f4 00 00 2a 05  5f 5a 00 00  C.=....*. _Z..

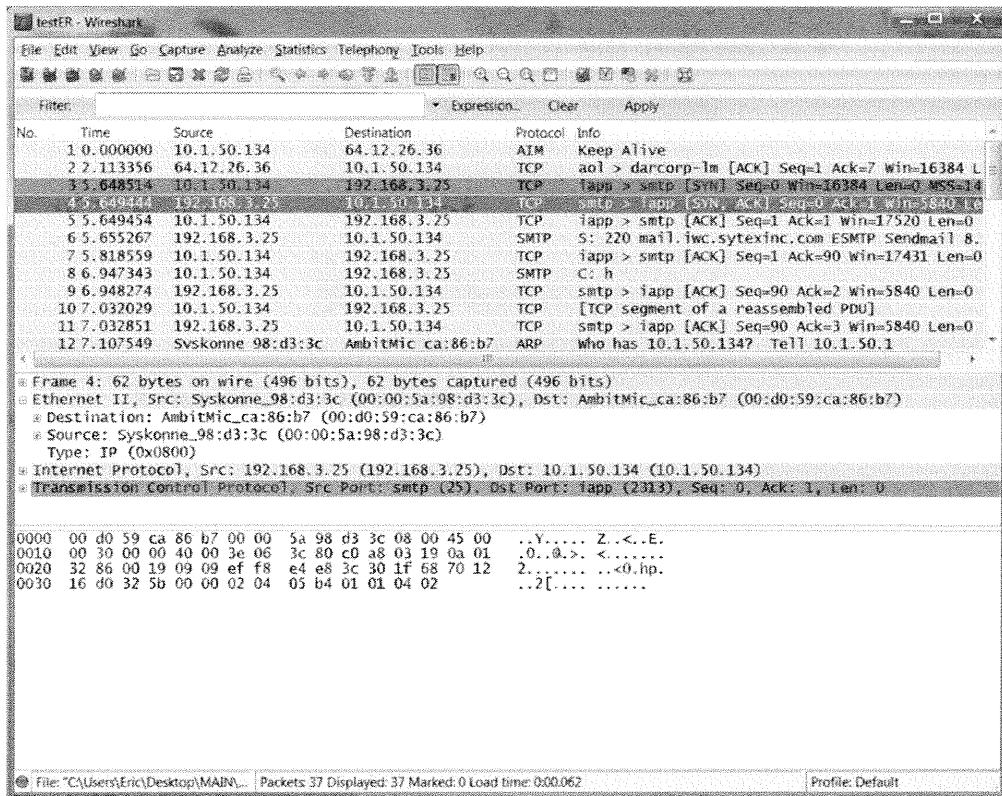
```

File: "D:\Wireshark\testER" 3490 Bytes 00:00:10 | Packets: 37 Displayed: 37 Marked: 0 Load ti... | Profile: Default

The GUI for Wireshark has three main windows. The data included in each window gets more specific from top to bottom:

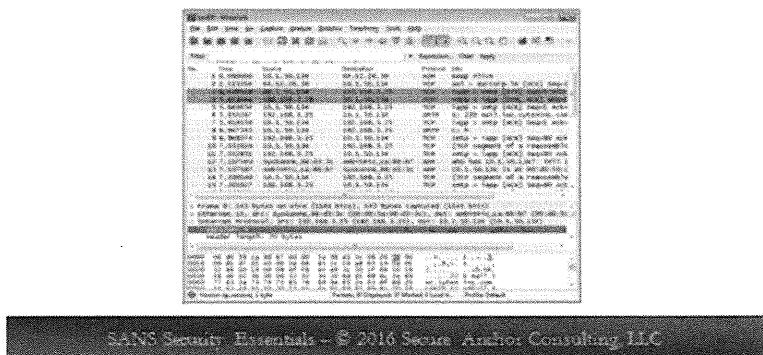
- The top window lists the packets collected during the capture. Selecting a packet in this window displays detailed information about the packet in the lower two windows.
- The middle window illustrates a tree view of the packet. To ease the analysis of captured data, the information shown here is layered, which is similar to the OSI model. The first tree details information about the Ethernet frame. For example, an HTTP packet has the Frame, Ethernet, IP, TCP, and HTTP trees.
- The last window displays data from the packet. The information you select in the middle window is highlighted in this last window. You can sort the information by clicking the header of each column.

If you select a packet, in this case line 4, which is a connection to a mail server, you are presented with details. As shown in the following screen, in the middle window, you can expand the *Ethernet II* section to view the MAC addresses and the gateway for the test network.



Running Wireshark (3)

- Analyzing the data



Running Wireshark (3)

Analyzing the Data

Now let's look at how you can use Wireshark to examine an actual packet. This section explains how to capture packet information and how to filter packet information.

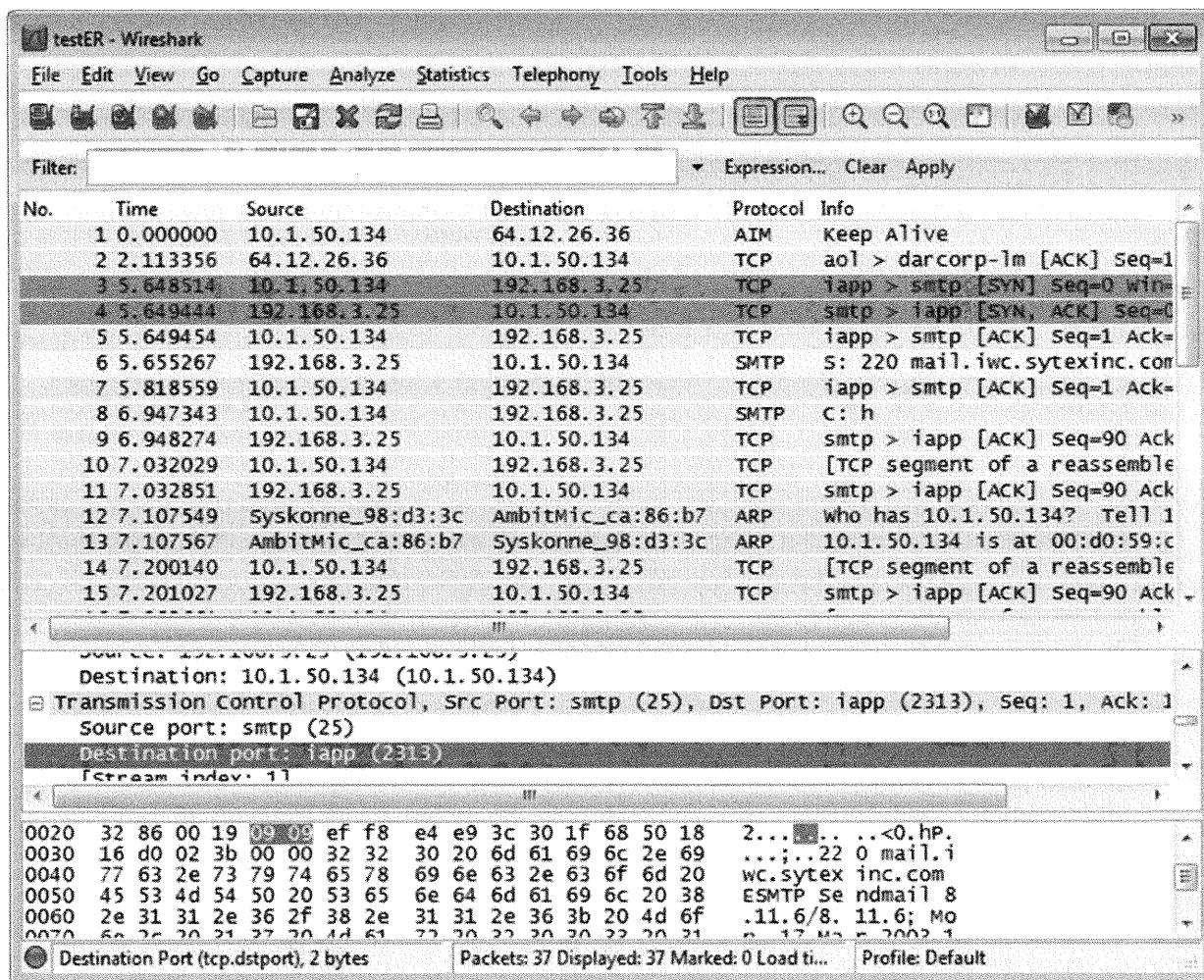
Capturing Packet Information

Expanding the Internet Protocol tree reveals all the IP-level details for the packet. Source and destination addresses, time to live, and protocol are just a few of the items listed within the Internet Protocol tree for this packet. What is nice about Wireshark is that if you select an item from the tree, it shows you where the item is located in the actual packet. In the middle window, highlight the version and you will see in the bottom window that the part of the header that corresponds to the version number is highlighted.

The screenshot shows the Wireshark interface with the following details:

- Packet List:** Shows 37 captured packets. The selected packet (Frame 6) is highlighted in blue. The details for this packet are shown below.
- Selected Packet Details:**
 - Frame 6:** 143 bytes on wire (1144 bits), 143 bytes captured (1144 bits)
 - Ethernet II:** Src: Syskonne_98:d3:3c (00:00:5a:98:d3:3c), Dst: AmbitMic_ca:86:b7 (00:d0:55:b7)
 - Internet Protocol:** Src: 192.168.3.25 (192.168.3.25), Dst: 10.1.50.134 (10.1.50.134)
- Selected Field:** Version: 4
- Selected Sub-field:** Header length: 20 bytes
- Hex View:** Displays the raw byte sequence of the selected packet's header. The first few bytes are: 00 d0 59 ca 86 b7 00 00 5a 98 d3 3c 08 00 ...
- Selected Sub-field:** Version (ip.version), 1 byte
- Bottom Status Bar:** Packets: 37 Displayed: 37 Marked: 0 Load ti... Profile: Default

Looking a step deeper into the packet, you can expand the Transmission Control Protocol tree, as shown in the following screen. Included in this tree are the source and destination ports, sequence numbers, checksums, and various other items included within a TCP packet. You might have to scroll down in the middle window to be able to expand the Transmission Control Protocol tree. Again, if you highlight a line in the middle window, the corresponding piece in the header will be highlighted in the lower window. From the middle window, click *Destination Port* to see how this works.

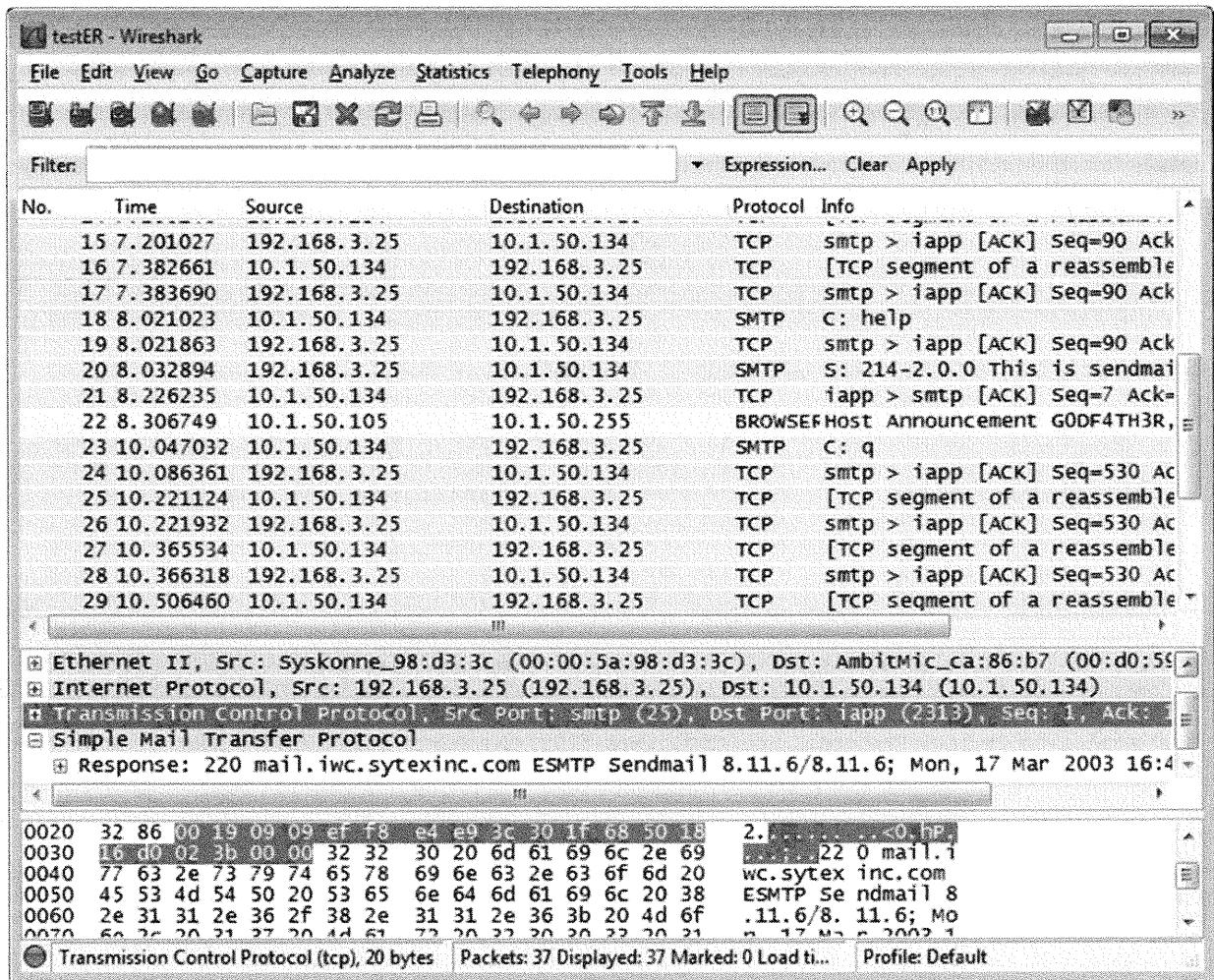


The final layer of information included within this packet is the application data. In this case, you are viewing a mail connection, which is Simple Mail Transfer Protocol (SMTP) header information. In the top window, click line 6. Move your mouse to the middle window, click the right mouse button and from the pop-up menu, click *Collapse All*.

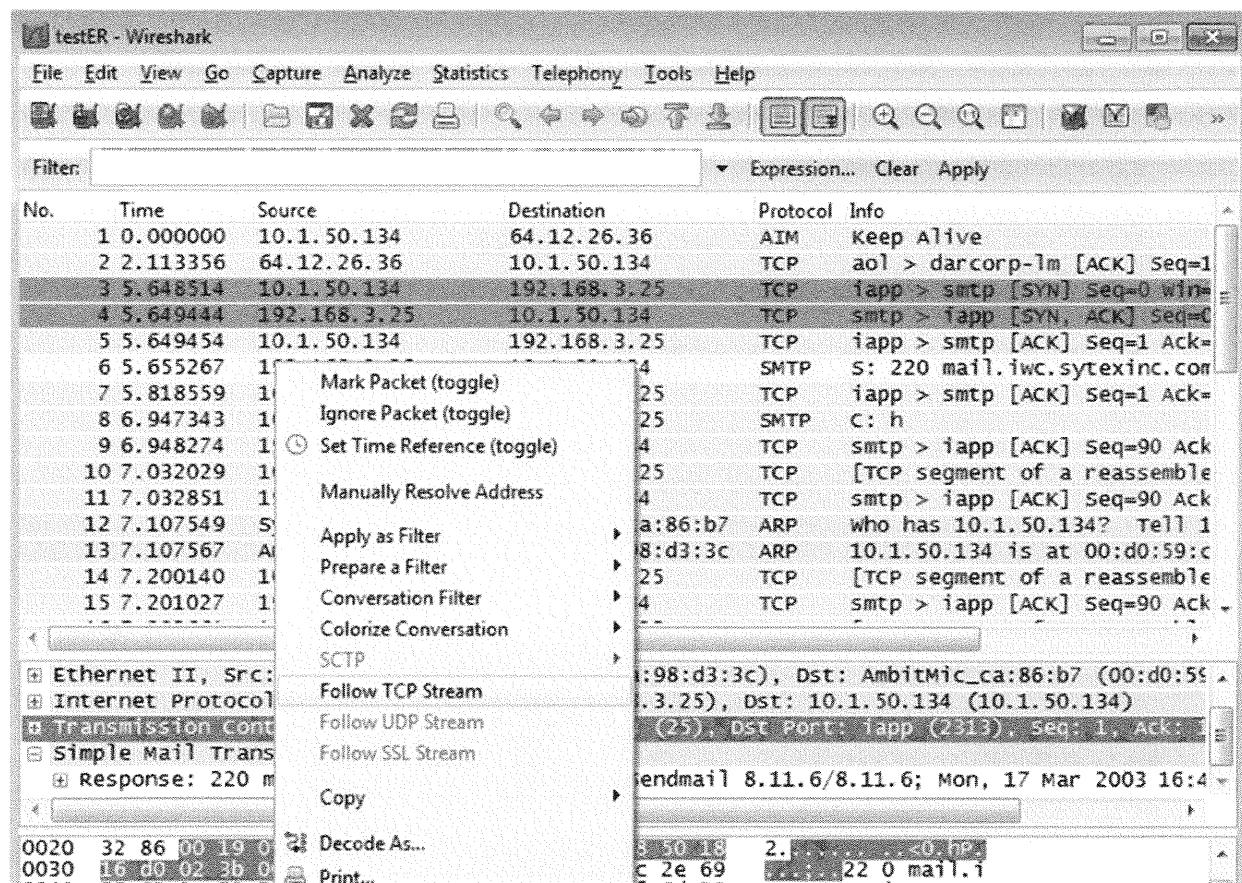
The screenshot shows the Wireshark interface with the following details:

- Top Window (List View):** Shows a list of network packets. The selected packet (Frame 6) is highlighted in blue.
- Middle Window (Details View):** Displays the selected packet's details. The selected row is expanded, showing the following information:
 - Frame 6:** 143 bytes on wire (1144 bits), 143 bytes captured (1144 bits)
 - Ethernet II, Src: Syskonne_98:d3:3c (00:00:5a:98:d3:3c), Dst: AmbitMic_ca:86:b7 (00:d0:59:c)**
 - Internet Protocol, Src: 192.168.3.25 (192.168.3.25), Dst: 10.1.50.134 (10.1.50.134)**
 - Transmission Control Protocol, Src Port: smtp (25), Dst Port: iapp (2313), Seq: 1, Ack: 1,**
 - Simple Mail Transfer Protocol**
- Bottom Window (Hex/Bin View):** Shows the raw hex and ASCII data of the selected packet.

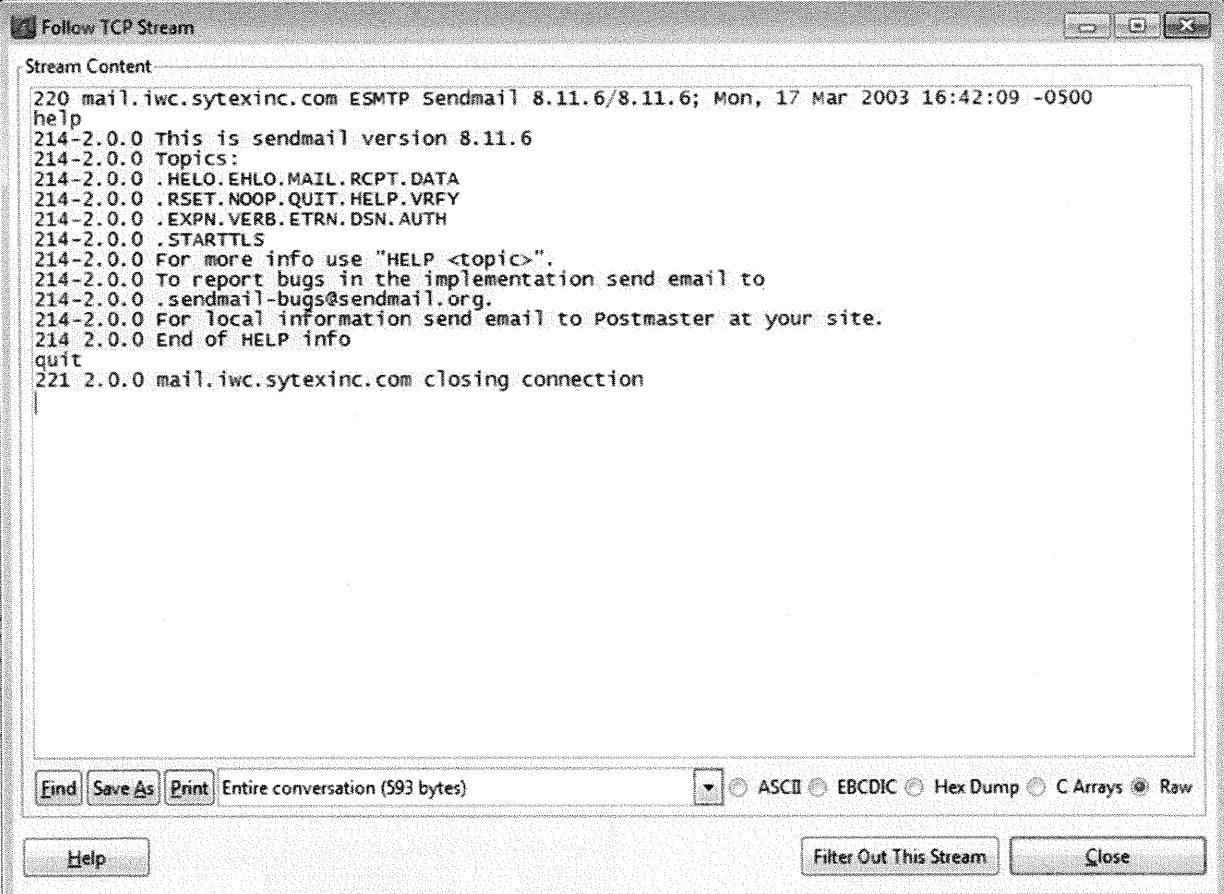
Now all the headers are collapsed. From the middle window, click the + symbol next to the Simple Mail Transfer Protocol line. Then, click the message line below and you can see the corresponding information in the packet.



During the analysis of data in a capture file, you often want to follow a specific communication stream. To accomplish this, right-click the packet in question, and then click *Follow TCP Stream*.



The contents of the TCP stream now display in a new window. All communications listed in red are from the client, and all communications listed in blue are from the server.



The screenshot shows a window titled "Follow TCP Stream". The main area is labeled "Stream Content" and contains the following text:

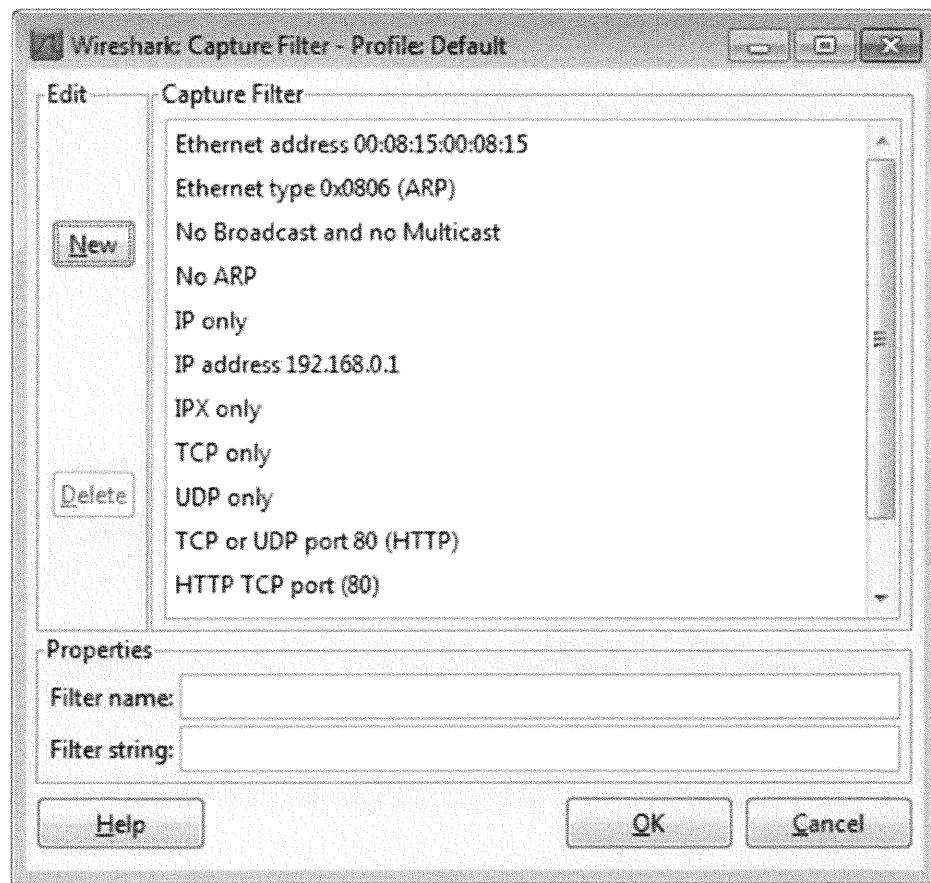
```
220 mail.iwc.sytexinc.com ESMTP Sendmail 8.11.6/8.11.6; Mon, 17 Mar 2003 16:42:09 -0500
help
214-2.0.0 This is sendmail version 8.11.6
214-2.0.0 Topics:
214-2.0.0 .HELO.EHLO.MAIL.RCPT.DATA
214-2.0.0 .RSET.NOOP.QUIT.HELP.VRFY
214-2.0.0 .EXPN.VERB.ETRN.DSN.AUTH
214-2.0.0 .STARTTLS
214-2.0.0 For more info use "HELP <topic>".
214-2.0.0 To report bugs in the implementation send email to
214-2.0.0 .sendmail-bugs@sendmail.org.
214-2.0.0 For local information send email to Postmaster at your site.
214 2.0.0 End of HELP info
quit
221 2.0.0 mail.iwc.sytexinc.com closing connection
```

Below the text area, there are several buttons: "Find", "Save As", "Print", and "Entire conversation (593 bytes)". To the right of these are dropdown menus for "Encoding" (set to "ASCII") and "Format" (set to "Raw"). At the bottom of the window are three buttons: "Help", "Filter Out This Stream", and "Close".

Here, you can see a simple interaction where someone connected to a mail server, issued the help command and quit the session.

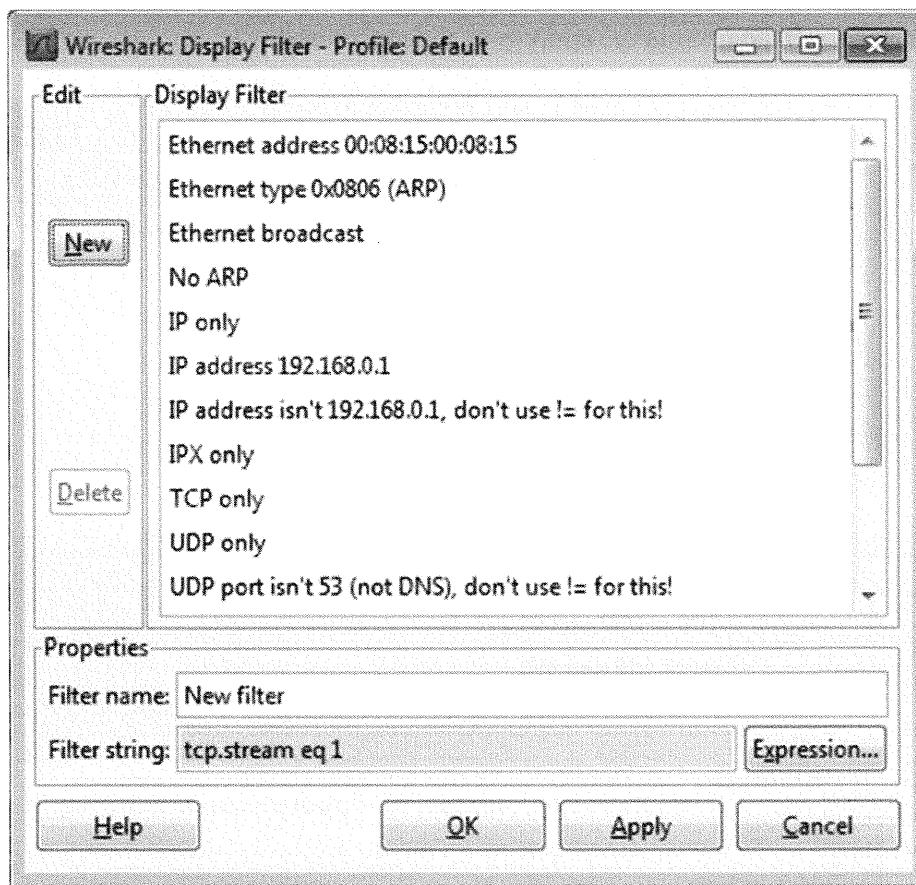
Filtering Packet Information

You have a few options for filtering the information you see or the packets you collect during a capture. The Capture Filters option on the Caption menu allows you to use the Libpcap filter language to determine which data gets captured. For example, you can create a filter that captures only SMTP traffic to or from a certain host by writing the filter, TCP port 25, and host IP address, where IP address is the address of the mail server.

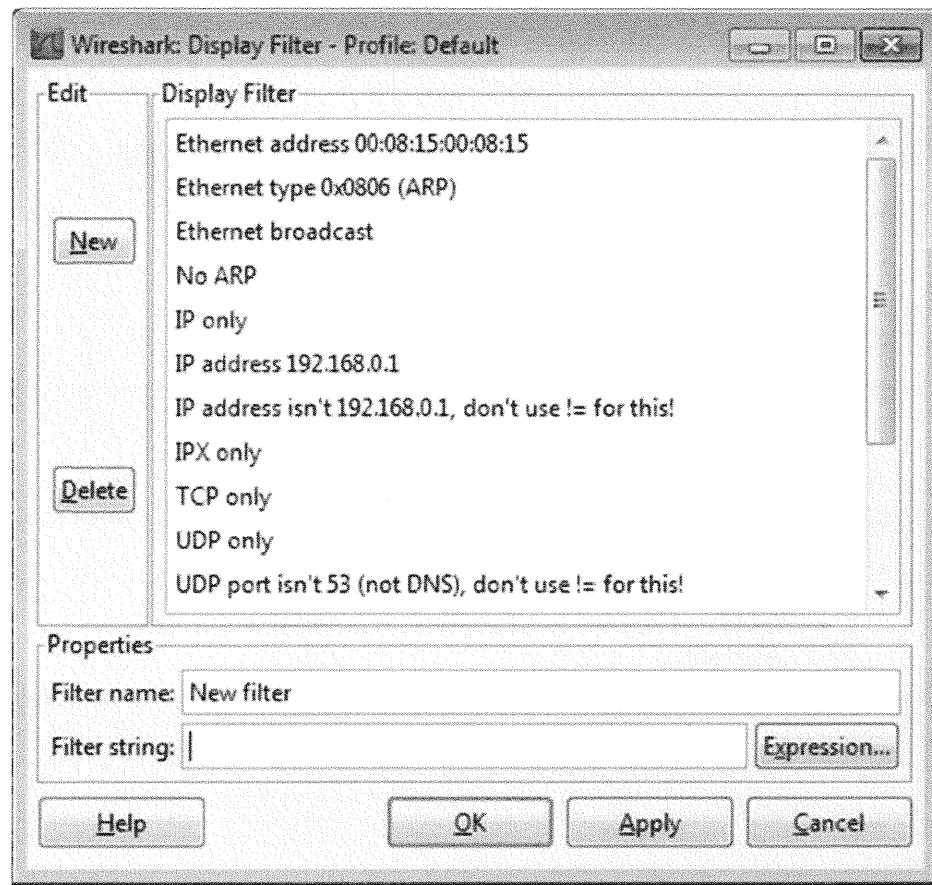


Another option is to capture all packets, and then create a filter that displays any data that matches the filter. To create a display filter, perform the following steps:

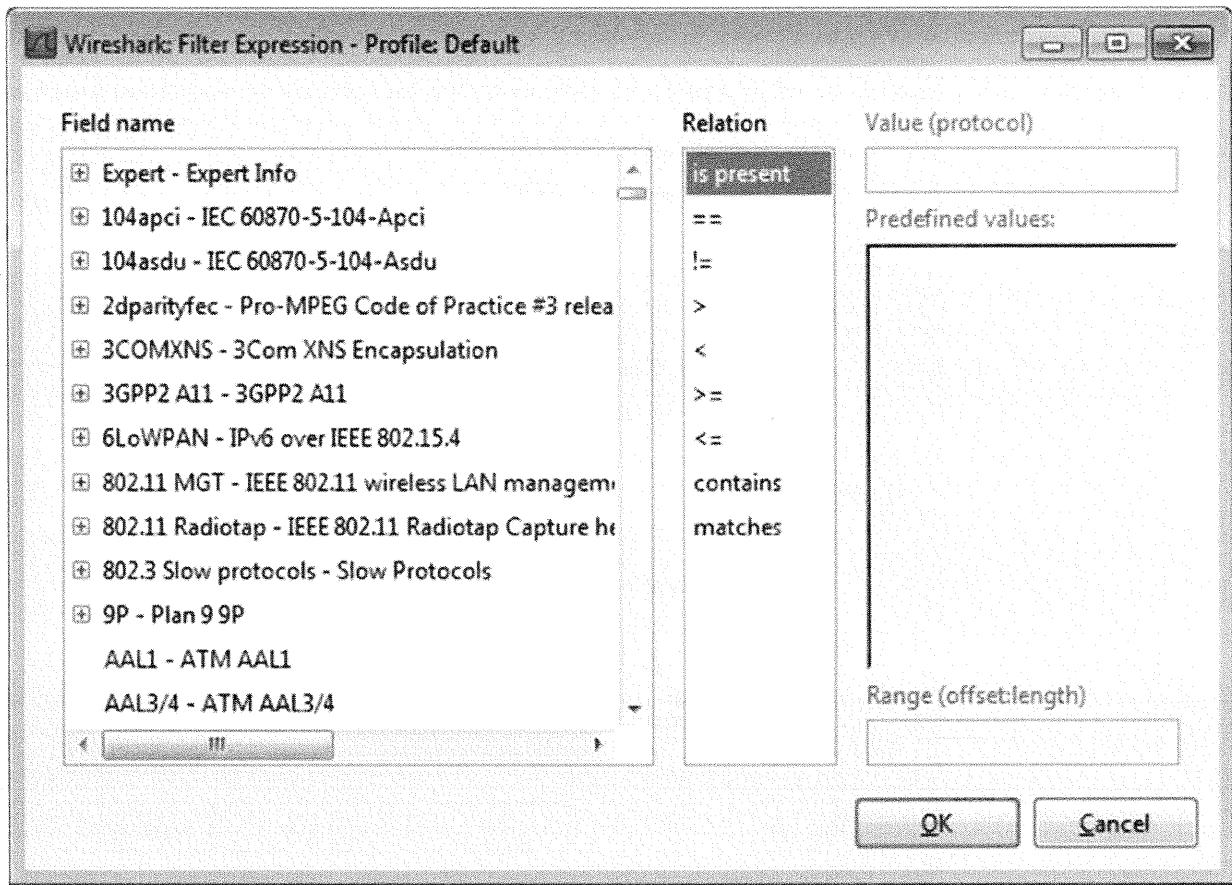
1. After capturing packets on a network, click *Analyze*, *Display Filters*. The Wireshark: Display Filter window, shown in the following screen, allows you to create new filter expressions.



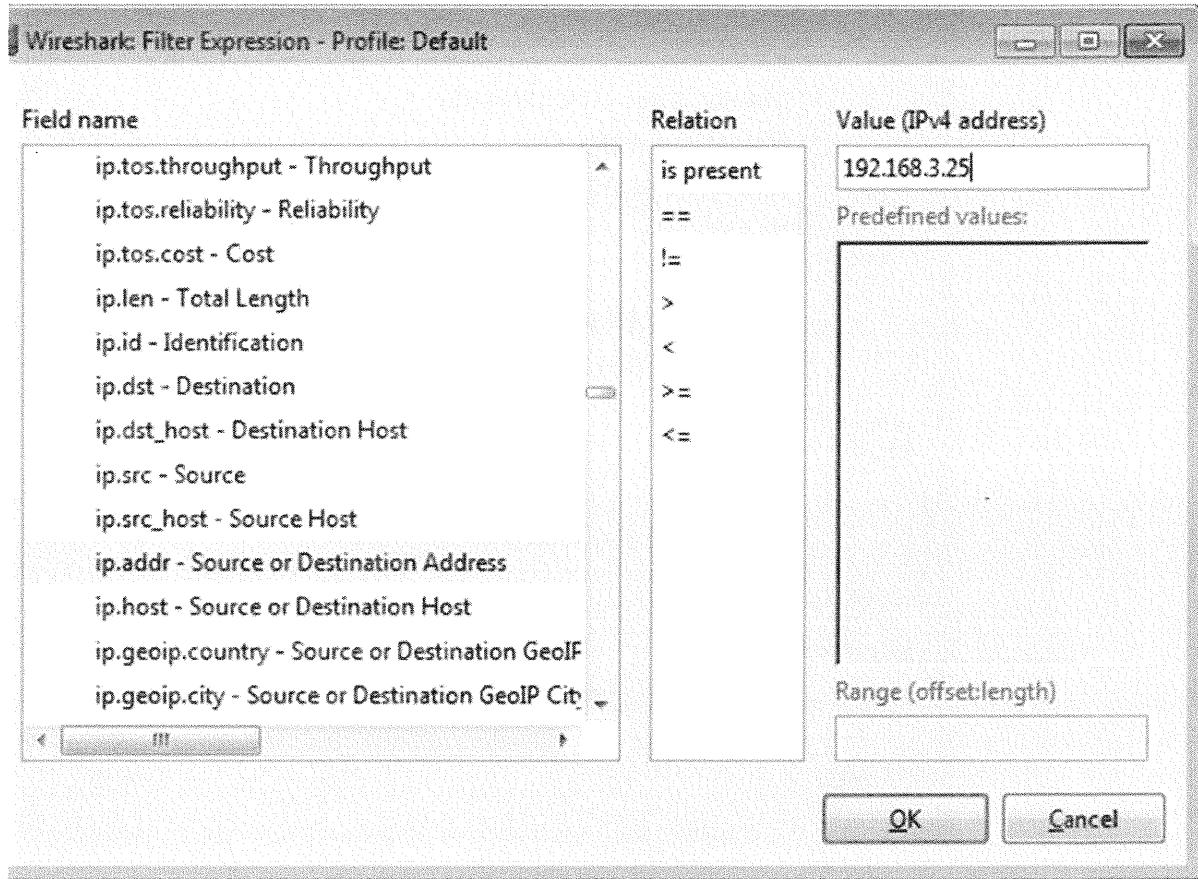
2. To create a new filter, click *New* and then delete all the information currently in the Filter string field.



3. To create a new expression, at the end of the Filter string line, click *Expression*. The Filter Expression window provides a relatively easy way to create new expressions.

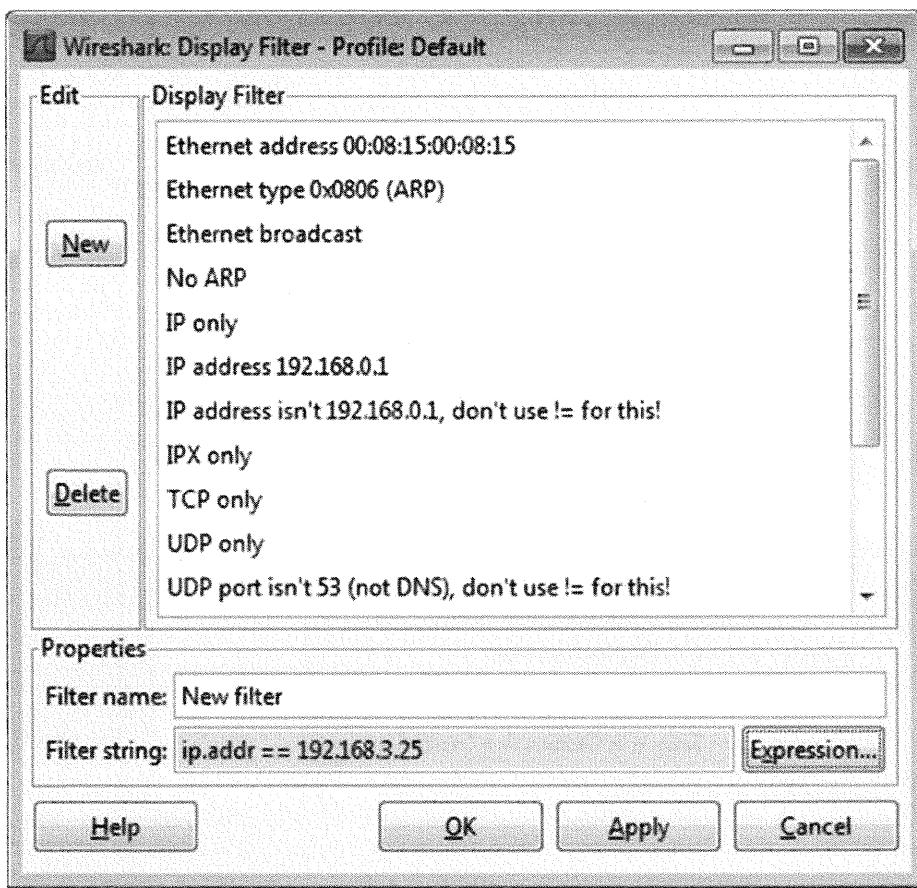


4. To create a new expression, from the Field name window, click *IP*. You can choose to filter on any of the displayed options. Within IP, choose *Source or Destination Address*. Next, in the Relation menu, click $=$. Finally, enter the IP address of the host that you want to view in the Value (IPv4 address) field. This particular expression shows only those packets that have a source or destination address of 192.168.3.25. After creating the expression, click *OK*.

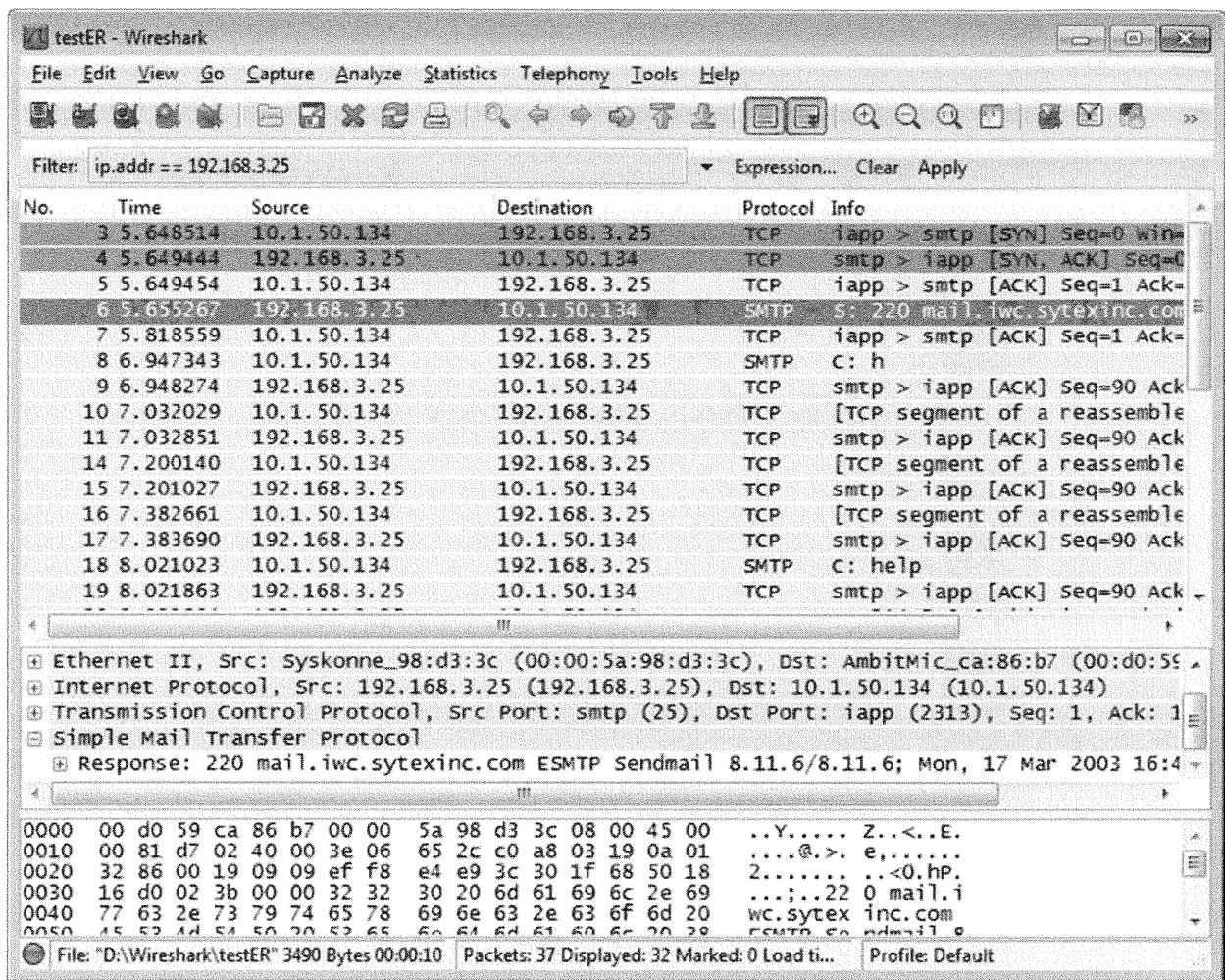


5. You see the new expression that the Add Expression Wizard created, as shown in the Filter string field in the following screen.

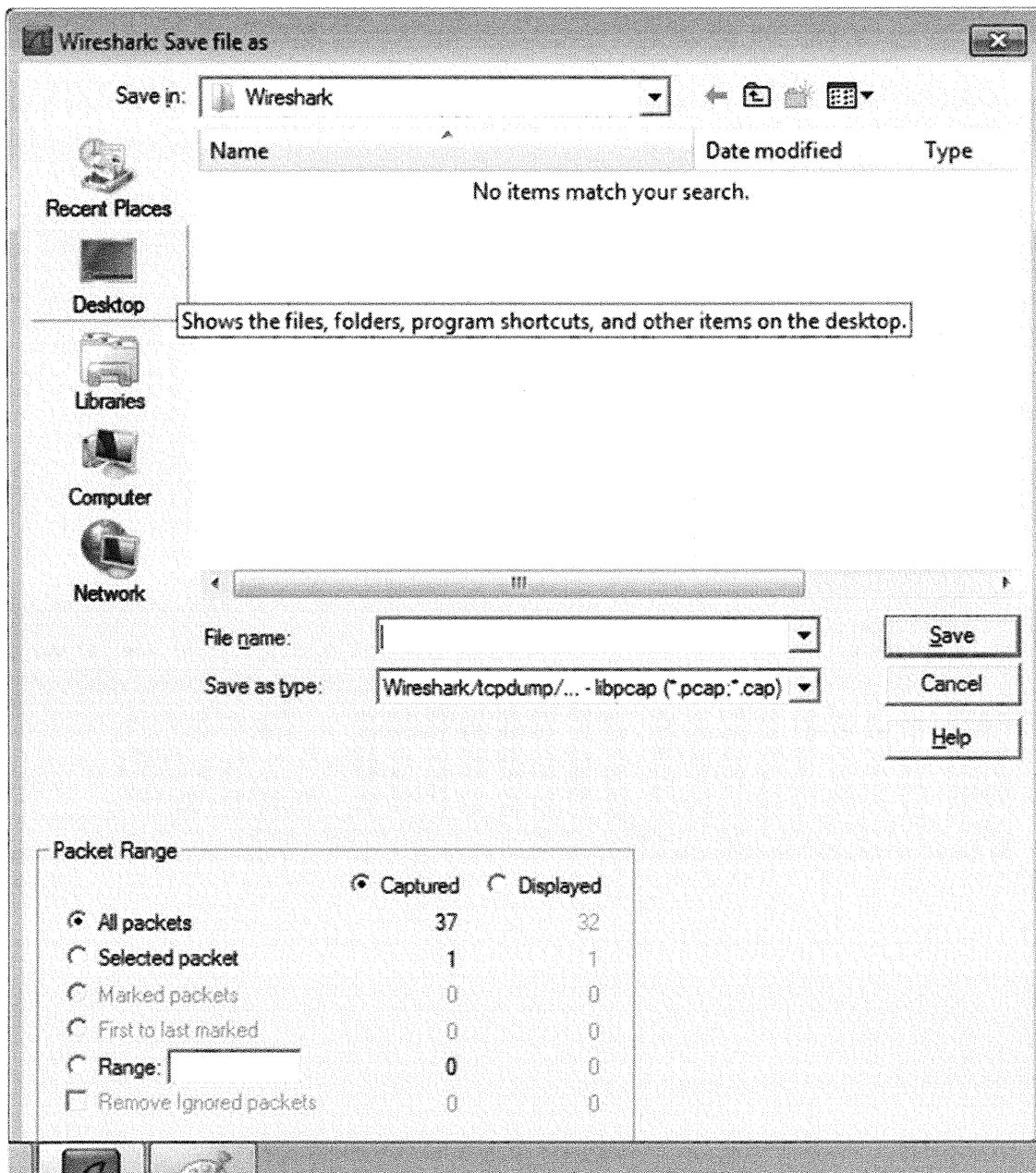
6. Next, name the newly created filter by typing a name in the Filter name field.



7. After you create the filter, you can apply it by clicking *Apply* and then clicking *OK*.



8. Save a dump file by clicking *File, Save As*. The File type drop-down menu provides a list of formats in which you can save the file. Select the type that you want to use, enter the name and location where you want to save the file, and click *Save*.



As you can see from this demonstration, taking the time to learn Wireshark and how to use Wireshark Display Filters will save you large amounts of time when you are troubleshooting, investigating incidents, or just trying to understand how things work.

VoIP Assessment with Wireshark

VoIP networks can be challenging to deploy and manage, requiring different protocols for call setup and for streaming audio, as well as companion protocols to assist with QoS, device discovery and lookup, file transfers, and more.

Wireshark is a powerful tool that can help provide visibility into the operation of VoIP networks by providing a decoded view of packets sent between hosts as well as intelligent protocol assessment mechanisms that can identify statistics and errors in transmission. Wireshark can also record and save the contents of an audio conversation between two participants to an audio file. With the appropriate permission, this feature is a wonderful mechanism to identify drops or packet loss in conversations that can affect the audio quality of a voice conversation; without permission, this can be an alarming attack, where an attacker can easily snoop on phone conversations.

Now you use Wireshark to inspect VoIP-related protocols, use the statistics collection mechanisms to identify and report on VoIP traffic, identify errors in the transmission of a VoIP stream, and save a VoIP conversation to an audio file for playback.

Warning: Make sure you get the appropriate permissions to do this work prior to performing any audits.

Wireshark and VoIP

- Wireshark can assess VoIP traffic:
 - Protocol dissector support for H.323, SIP, and RTP
- Intelligent VoIP statistical analysis
- Identifies errors in VoIP streams
- Can save RTP traffic as audio files

SANS Security Essentials – © 2016 Secure Anchor Consulting LLC

Wireshark and VoIP

To support the analysis of VoIP networks, Wireshark includes extensive support for VoIP protocols, including the ability to decode H.323, SIP, and RTP traffic. In addition to the protocol decoders, Wireshark also includes the ability to perform statistical analysis on VoIP traffic to report the utilization of VoIP protocols, and can intelligently examine protocol traffic to identify errors in VoIP streams. Finally, Wireshark can take an RTP stream of traffic and decode a number of CODECs to convert the packet content into an .au audio file, which can be played back with nearly all audio players. This feature allows an administrator to record a conversation and listen for anomalies in the audio stream that might indicate dropped packets or other network problems.

Wireshark's VoIP Support

- Designed to help analyze VoIP traffic
- Useful for troubleshooting an existing deployment, or monitoring a new deployment
- Blackhat or whitehat tool, depending on permission!

SANS Security Essentials – © 2016 Secure Analytics Consulting, LLC

Wireshark's VoIP Support

Support in Wireshark for analyzing VoIP traffic was designed to help you analyze the network traffic in your environment, which is often a necessary component of network troubleshooting. Wireshark is equally valuable when applied to a new VoIP deployment where the administrators must assess the network to identify whether it will support the additional load of VoIP traffic, or to an existing deployment where the administrator is charged with ensuring VoIP is working smoothly.

Exercise: Wireshark/VoIP

1. Examine SIP traffic
2. Collect SIP statistics
3. Graph call setup and protocol utilization
4. Identify errors in RTP stream
5. Save RTP traffic to audio file

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Exercise: Wireshark/VoIP

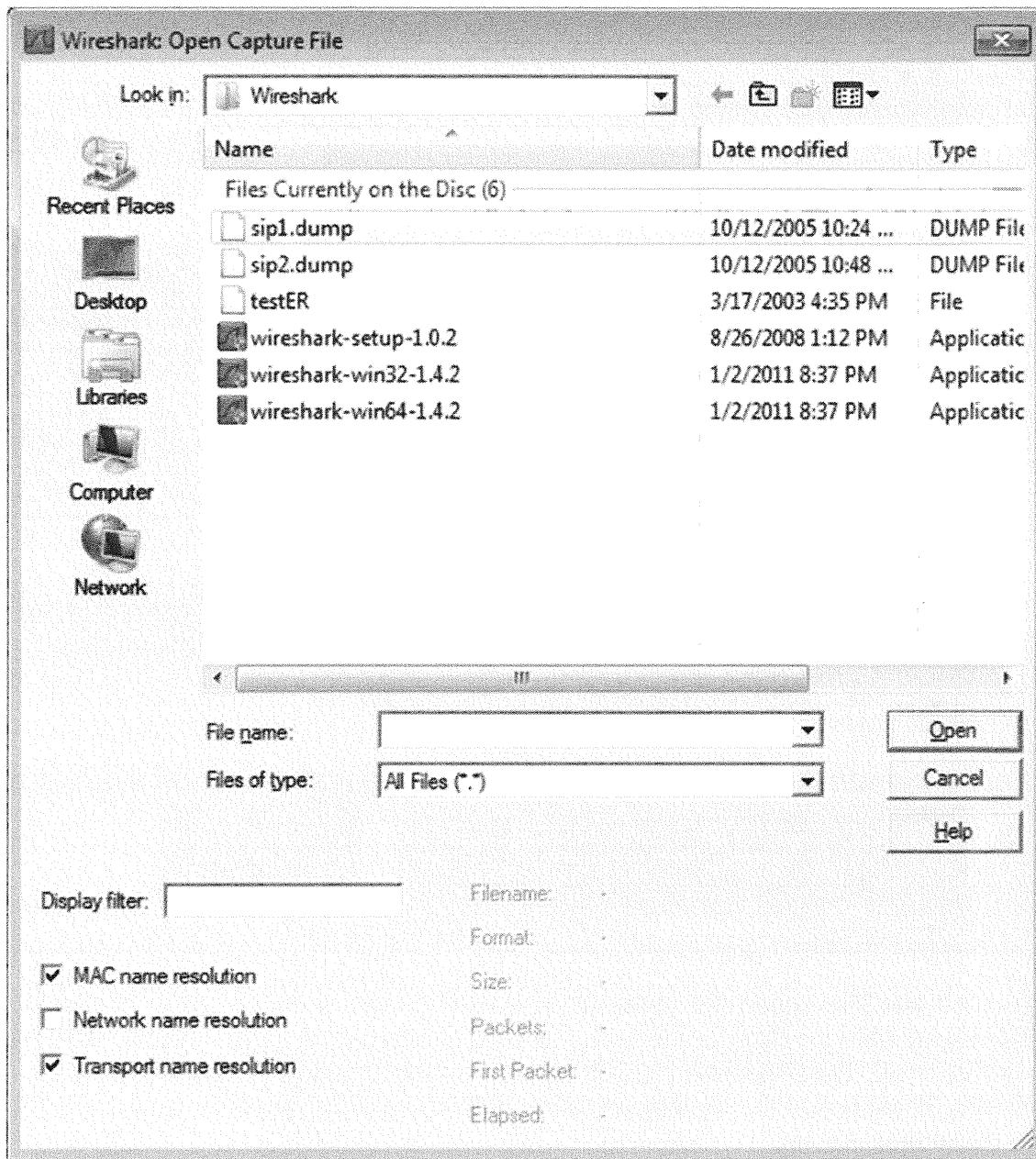
Complete the following exercises to gain hands-on experience using the powerful VoIP assessment features available with Wireshark.

Examine SIP Traffic

Complete the following steps to become familiar with the components of a SIP packet using the protocol dissector functionality in Wireshark:

1. Start *Wireshark*. Invoke the Wireshark program by clicking *Start, All programs, Wireshark, Wireshark*. An empty Wireshark window displays, as shown in the following screen.

2. Open *Supplied Capture File*. Next, click *File*, *Open* to open the Open Capture File dialog box. On the left, double-click the lab CD-ROM drive, and then navigate to the Wireshark directory. Click the *sip1.dump* file and click *Open*, as shown in the following screen.



3. Examine Frame 7. After Wireshark loads the capture file, click frame 7 by clicking the frame in the Packet List view. Next, in the Packet Dissector view, expand the *Session Initiation Protocol* section. This reveals the three sections of the SIP packet, the Request Line, the Message Header, and the Message Body.

The screenshot shows the Wireshark interface with a capture file named "sip1.dump". The packet list pane displays 15 SIP-related frames. Frame 7 is selected, showing the following details:

- User Datagram Protocol, Src Port: sip (5060), Dst Port: sip (5060)**
- Session Initiation Protocol**
- Request-Line: INVITE sip:steve@192.168.1.254:5060 SIP/2.0**
- Message Header**
- Message Body**

The bytes pane shows the raw hex and ASCII data for the selected frame, which corresponds to the selected Request-Line.

Request Line: The request line in this frame is INVITE sip:steve@192.168.1.254:5060 SIP/2.0. This indicates that the caller is attempting to use the universal resource indicator (URI) steve@... to initiate a call. The IP address 192.168.1.254 is not the IP address of the call recipient, but rather the IP address of the registration server. SIP is a signaling protocol, exchanged between two registration servers.

Message Header: Expanding the message header line reveals additional details about the caller, including the From URI, the user-agent, an administrative contact URI (matching the URI in this case), the date, allowed methods, and additional information.

The screenshot shows the Wireshark interface with the following details:

- File Menu:** File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Tools, Help.
- Toolbar:** Standard file operations (Open, Save, Print, Copy, Paste, Find, etc.) and search functions.
- Filter Bar:** Filter: Expression... Clear Apply.
- Table View:** Shows a list of 15 SIP messages. The columns are No., Time, Source, Destination, Protocol, and Info. The messages are numbered 1 to 15 and show a sequence of SIP requests and responses between two hosts.
- Message Header Panel:** Displays the expanded message header for the selected packet (packet 1). It includes fields such as Via, From, To, and Contact.
- Hex and ASCII Panels:** Shows the raw hex and ASCII representation of the selected SIP message. The ASCII panel shows the SIP INVITE message with headers like Via, From, To, and Contact.
- Bottom Status Bar:** Frame (frame), 826 bytes; Packets: 945 Displayed: 945 Marked: 0 Load t...; Profile: Default.

Message Body: Expanding the message body header and the session initialization protocol header reveals additional configuration about the call including supported CODECs and other media attributes to be negotiated in the call.

The screenshot shows a Wireshark capture window titled "sip1.dump - Wireshark". The main pane displays a list of network packets, mostly SIP messages, between two hosts: 192.168.1.251 and 192.168.1.2. The packet details and bytes panes are visible at the bottom. A context menu is open over the 12th packet, which contains session configuration details:

- + Owner/Creator, Session Id (o): root 32732 32732 IN IP4 192.168.1.2
- Session Name (s): session
- Connection Information (c): IN IP4 192.168.1.2
- Time Description, active time (t): 0 0
- Media Description, name and address (m): audio 19844 RTP/AVP 0 3 18 101
- Media Attribute (a): rtpmap:0 PCMU/8000
- Media Attribute (a): rtpmap:3 GSM/8000
- Media Attribute (a): rtpmap:18 G729/8000
- m=audio 19844 RTP/AVP 0 3 18 101

The bytes pane shows the raw hex and ASCII data for the selected SIP message.

You can use Wireshark to examine the characteristics of a SIP call, using the protocol dissectors to identify the protocol header fields and meanings. Continue examining the SIP traffic in the sip1.dump file, identifying the response to the INVITE method in frame 7, as well as the session termination with the BYE method in frame 928.

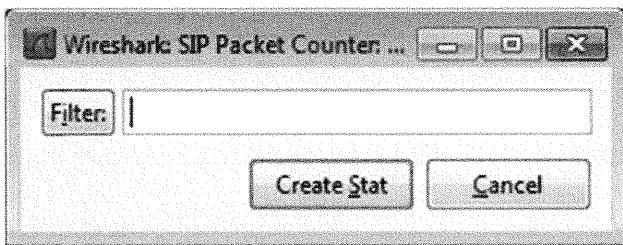
The screenshot shows the Wireshark interface with the following details:

- Panels:**
 - Packet List:** Shows 945 packets, with frame 928 highlighted. It lists columns: No., Time, Source, Destination, Protocol, and Info.
- Details:** Displays the dissected fields for the selected packet, including the SIP Request: BYE sip:1301@192.168.1.2.
- Bytes:** Shows the raw hex and ASCII representation of the selected packet's bytes.
- Status Bar:** Shows the file path "D:\Wireshark\sip1.dump", the number of packets (945), and the profile used (Default).

Collect SIP Statistics

Complete the following steps to collect statistics on a SIP exchange between two gateway servers. Continue using the capture sip1.dump file from the previous exercise.

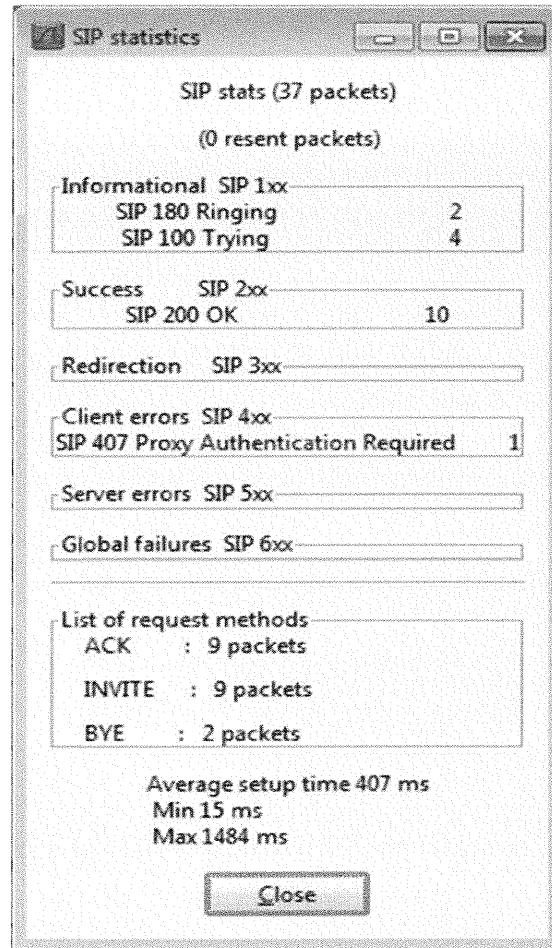
1. With the sip1.dump capture file open, click *Telephony - SIP* to open the SIP Packet Counter window, as shown in the following screen. This window allows you to enter a filter to summarize SIP statistics to a specific host or conversation, or any other valid Wireshark Display Filter syntax you specify. For the purposes of our analysis, no filtering is needed; click *Create Stat* to continue.



2. Examine SIP Statistics. After a few seconds, Wireshark populates the SIP statistics window. This informative window identifies the number of SIP packets in the capture file, identifies the number and list of SIP methods used in the call setup (INVITE, ACK, and BYE), and summarizes the response codes in the exchange.

SIP uses HTTP-like response codes to positively acknowledge each message sent in the call setup. The response codes are a three-digit number, where the first digit identifies the type of message being relayed:

- **1XX:** Messages beginning with 1 are used for provisioning purposes such as 100 (Trying).
- **2XX:** Messages beginning with 2 are used for conveying success status such as 200 OK.
- **3XX:** Messages beginning with 3 are used for redirect messages such as 301 (Moved Permanently) and 302 (Moved Temporarily).
- **4XX:** Messages beginning with 4 are used for conveying client errors such as 400 (Bad Request) or 401 (Unauthorized).
- **5XX:** Messages beginning with 5 are used for conveying server failure such as 500 (Internal Failure) or 501 (Not Implemented).
- **6XX:** Messages beginning with 6 are used for conveying global failure such as 604 (Protocol Not Supported).

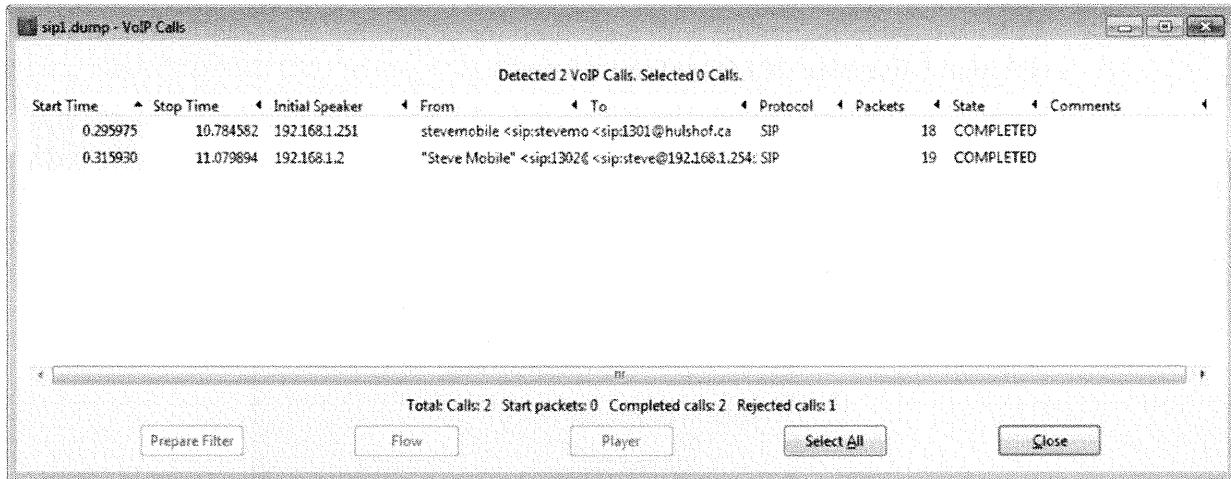


Wireshark provides a fast mechanism to assess a SIP conversation, quickly identifying errors returned by the SIP server. We can assess the purpose of these errors by the first digit of the response code to characterize the impact of the error.

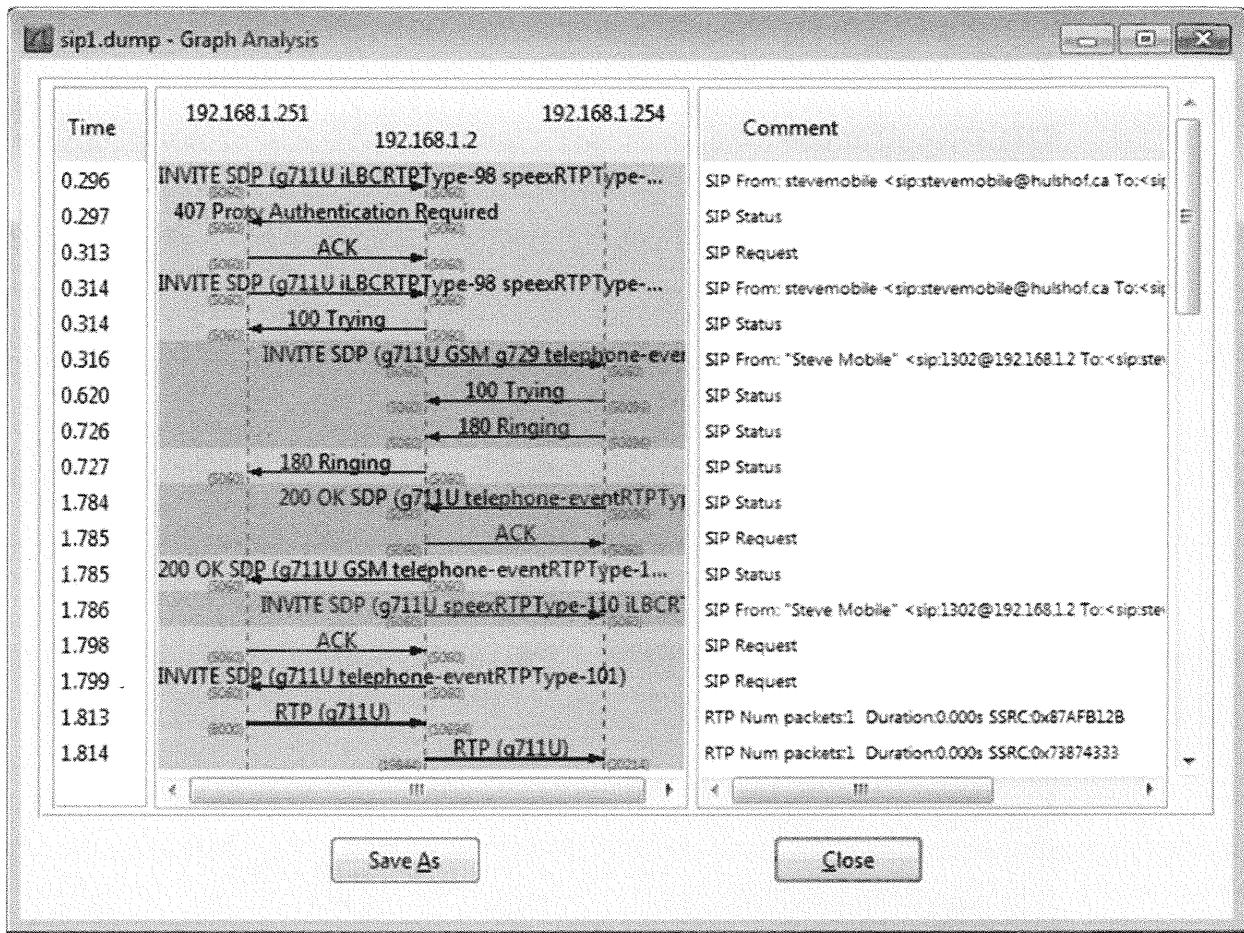
Graph Call Setup and Protocol Utilization

Complete the following steps to graph the call setup and protocol utilization of a VoIP exchange between two gateway servers. Continue using the sip1.dump capture file from the previous exercise.

1. Wireshark can parse a packet capture to identify all VoIP calls that are present, examining frames for RTP, SIP, or H.323 traffic. With the sip1.dump capture file open, click *Telephony, VoIP Calls* to open the VoIP Calls window, as shown in the following screen.



2. Click *Calls to Graph*. From the VoIP Calls window, single-click both the VoIP calls that are listed to highlight them. Next, click the *Flow* button to open the Graph Analysis screen, as shown in the following screen.



3. Examine Graph Analysis. Wireshark graphs the exchange between the selected VoIP calls, identifying the SIP and RTP exchange between the two systems, color-coding the origination of the callers differently to quickly assess the call exchange.

Examine the exchange between the two systems to identify the call setup exchange. SIP traffic is sent to the intermediate system at 192.168.1.2 (the VoIP gateway), but RTP traffic is between the two endpoints of the call.

Identify Errors in RTP Stream

Complete the following steps to assess the errors identified between two devices in the VoIP exchange. Continue using the sip1.dump capture file from the previous exercise.

1. Wireshark can identify all the RTP traffic in a capture, allowing the administrator to select additional analysis options. With the sip1.dump capture file open, click *Telephony, RTP, Show All Streams* to open the RTP Streams dialog box, as shown in the following screen.

The screenshot shows the 'RTP Streams' dialog box from Wireshark. The title bar says 'Wireshark: RTP Streams'. The main area displays a table with 8 rows of RTP stream information. The columns are: Src IP addr, Src port, Dst IP addr, Dst port, SSRC, Payload, Packets, and Lost. The last column shows percentages in parentheses. Below the table, there is a message: 'Select a forward stream with left mouse button, and then Select a reverse stream with Ctrl + left mouse button'. At the bottom are several buttons: Unselect, Find Reverse, Save As, Mark Packets, Prepare Filter, Copy, Analyze, and Close.

Src IP addr	Src port	Dst IP addr	Dst port	SSRC	Payload	Packets	Lost
192.168.1.2	19844	192.168.1.254	20214	0x73874333	g711U	1	0 (0.0%)
192.168.1.2	10694	192.168.1.251	8000	0x484F4E6F	g711U	207	0 (0.0%)
192.168.1.251	8000	192.168.1.2	10694	0x87AFB12B	g711U	204	149 (42.2%)
192.168.1.251	8000	192.168.1.254	20214	0x87AFB12B	g711U	142	7 (4.7%)
192.168.1.251	8000	192.168.1.254	10694	0x87AFB12B	g711U	7	0 (0.0%)
192.168.1.251	8000	192.168.1.2	20214	0x87AFB12B	g711U	97	0 (0.0%)
192.168.1.254	20214	192.168.1.2	19844	0x35C2C394	g711U	12	233 (95.1%)
192.168.1.254	20214	192.168.1.251	8000	0x35C2C394	g711U	233	0 (0.0%)

2. Examine Summary Statistics. The RTP Streams dialog box identifies all the RTP traffic in the capture file including the source and destination address and port combinations, the payload or CODEC for the media stream, the number of packets in the stream, and the percentage and count of lost traffic with the statistics on jitter. Scrolling to the right of the RTP Streams dialog box reveals a dialog box where Wireshark indicates whether the stream is problematic or experiencing an acceptable packet loss/jitter level.

Wireshark RTP Streams

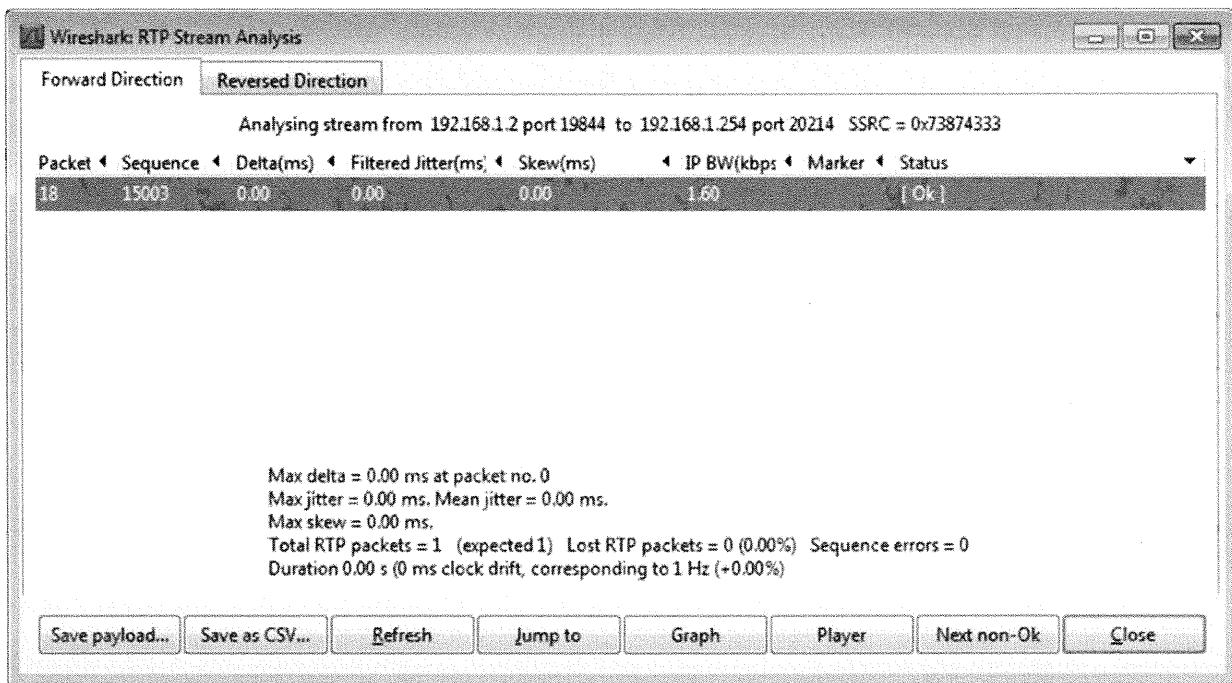
Detected 8 RTP streams. Choose one for forward and reverse direction for analysis

Payload	_packets	Lost	Max Delta (ms)	Max Jitter (ms)	Mean Jitter (ms)	Pb?
g711U	1	0 (0.0%)	0.00	0.00	0.00	X
g711U	207	0 (0.0%)	41.12	10.28	4.38	X
g711U	204	149 (42.2%)	2985.25	10.25	4.46	X
g711U	142	7 (4.7%)	166.86	8.00	4.90	X
g711U	7	0 (0.0%)	21.62	0.33	0.88	X
g711U	97	0 (0.0%)	42.33	12.68	4.78	X
g711U	12	233 (95.1%)	20.11	480.38	0.84	X
g711U	233	0 (0.0%)	27.52	33920644.08	2304167.23	X

Select a forward stream with left mouse button, and then
Select a reverse stream with Ctrl + left mouse button

Unselect Find Reverse Save As Mark Packets Prepare Filter Copy Analyze Close

3. Select a Stream for Analysis. Click the first stream in the RTP Streams dialog box, and then click the *Analyze* button. This opens the Wireshark RTP Stream Analysis dialog box, as shown in the following screen.



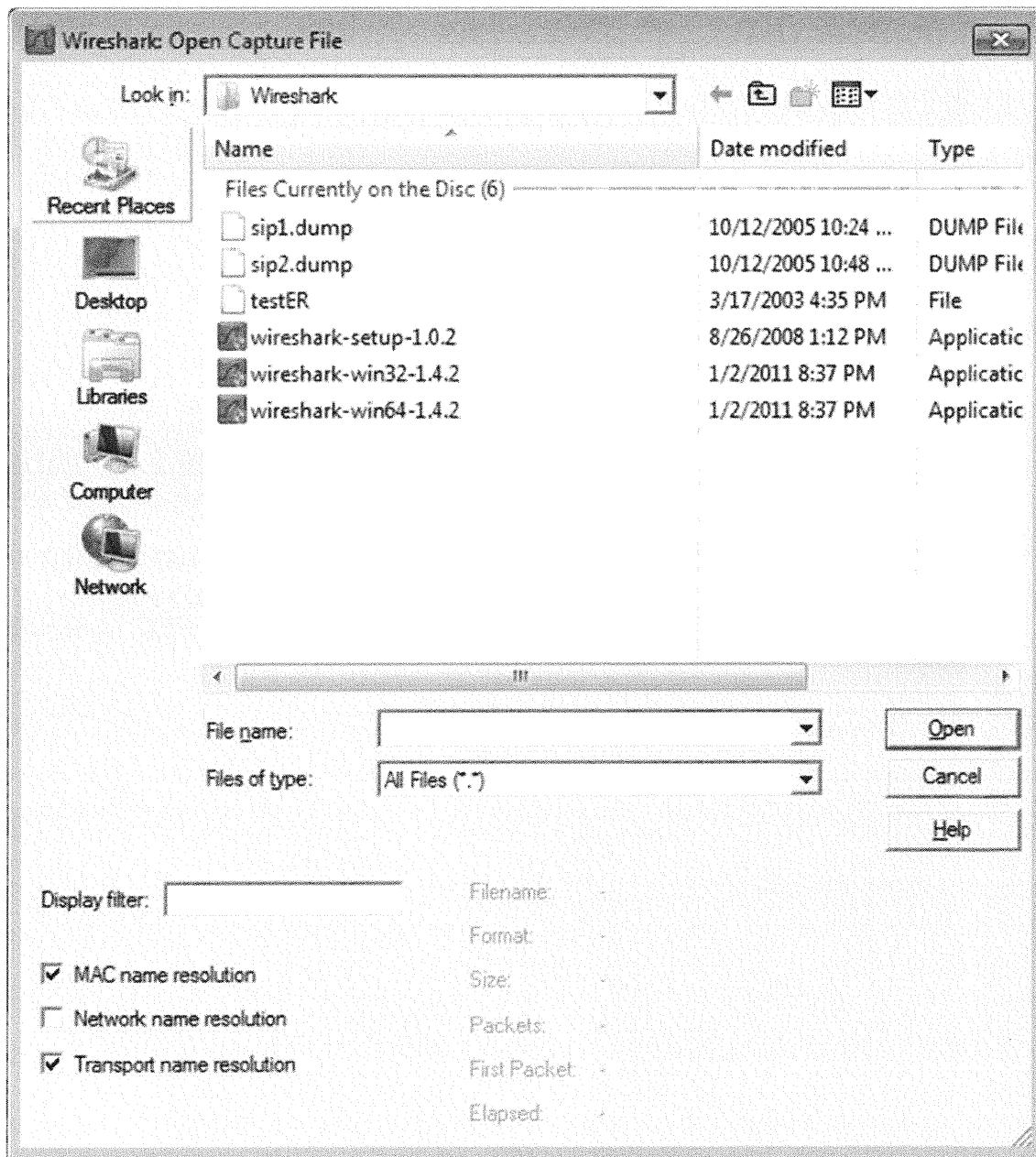
4. Examine Detailed Statistics. Examine the statistics provided to gain additional insight into the selected RTP stream. Wireshark identifies any packet framing errors by highlighting the frames in red, as shown in the previous screen.

Wireshark includes powerful analysis capabilities to give administrators additional insight into the nature of VoIP traffic on the network. Using RTP stream analysis, administrators can identify problematic data streams, and examine the detailed analysis or per-packet views to gain additional insight into the nature of the problematic traffic.

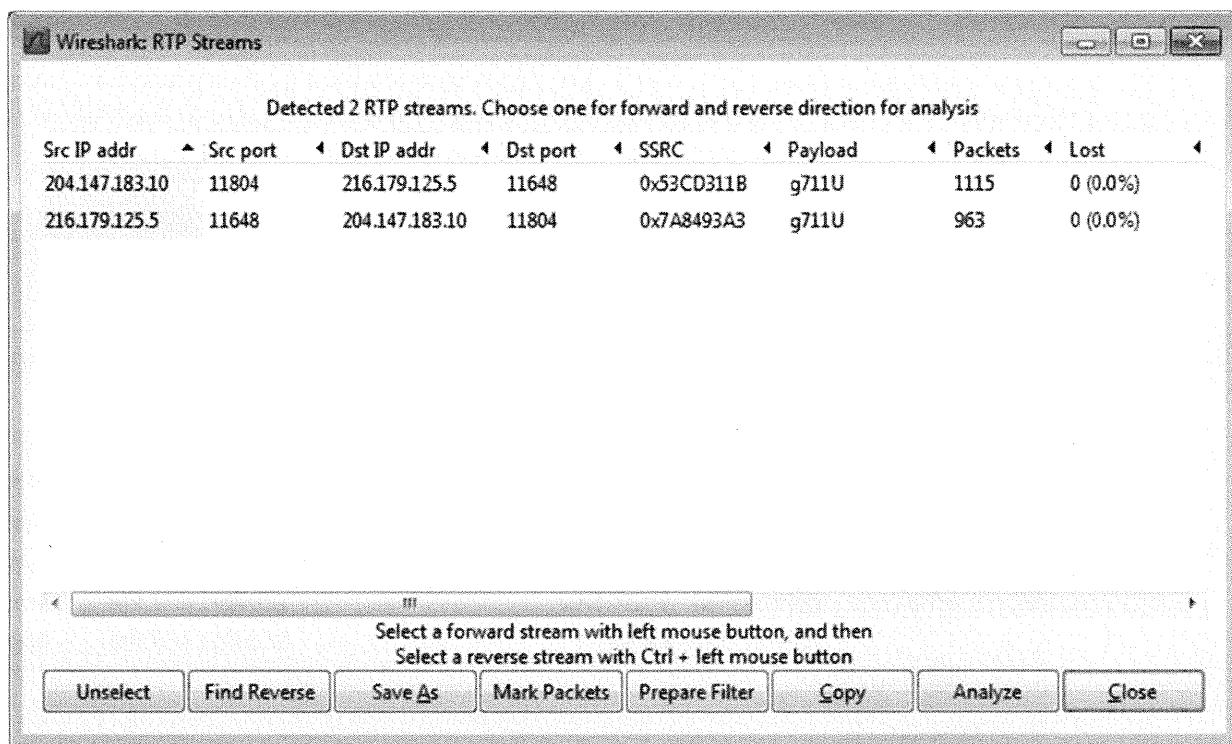
Save RTP Traffic to an Audio File

Complete the following steps to extract the audio conversation from RTP traffic using Wireshark.

1. Open Supplied Capture File. Click *File*, *Open* to open the Open Capture File dialog box. Double-click the lab CD-ROM drive on the left, and then navigate to the VoIP directory. Select the sip2.dump file and click *Open*, as shown in the following screen.

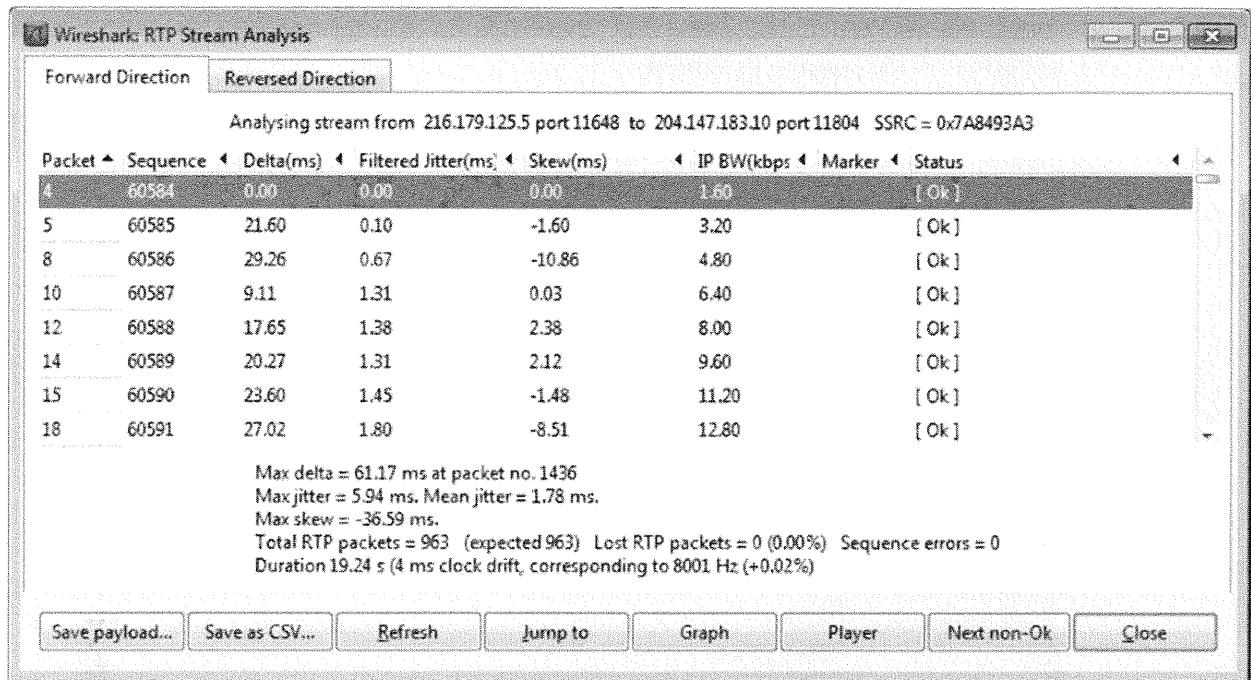


2. Identify RTP Streams. Click *Telephony, RTP, Show All Streams* to open the RTP Streams dialog box. This displays the two RTP streams present in the capture file (for a bi-directional conversation), as shown in the following screen.

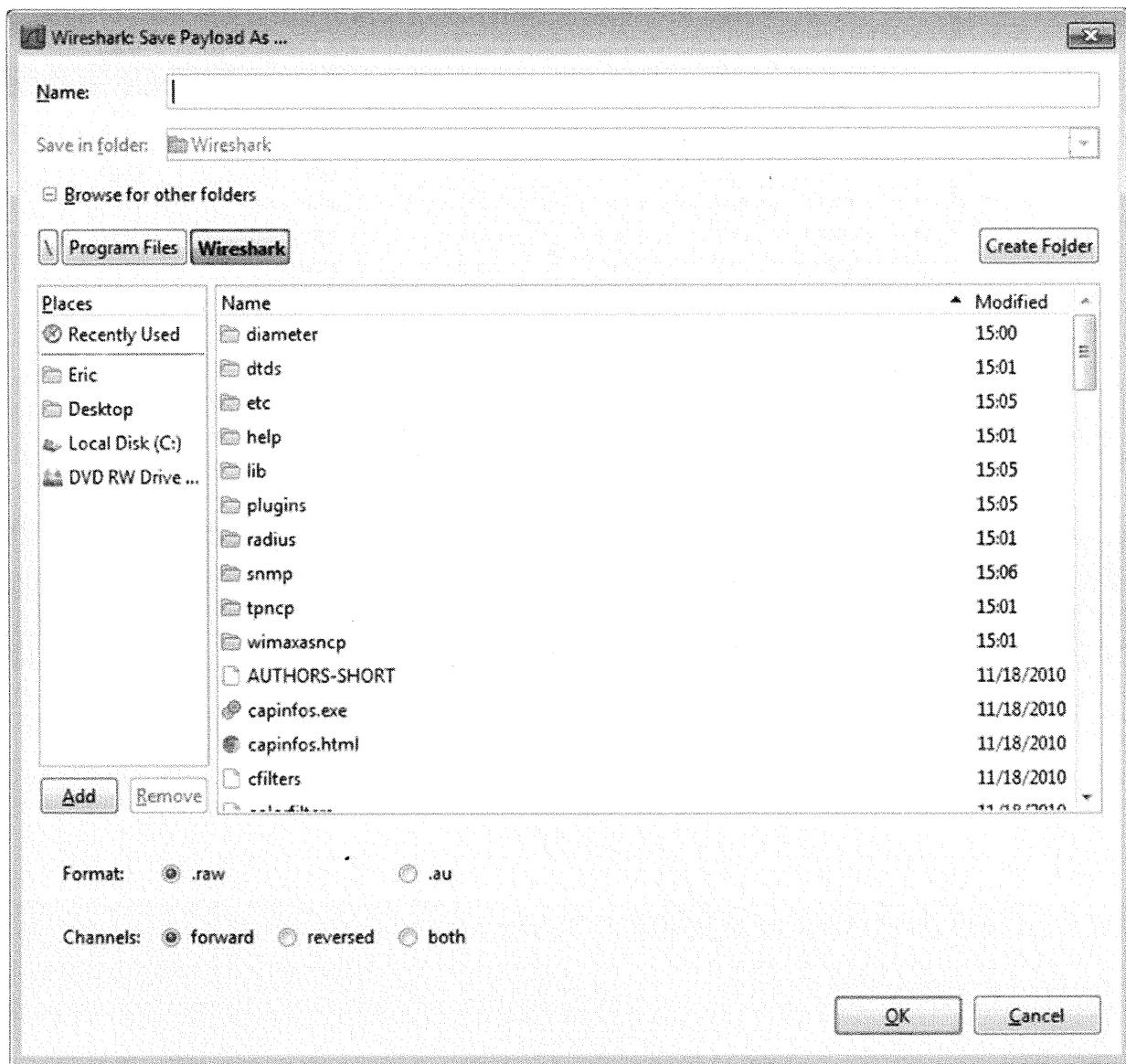


3. Select RTP Stream. Select the top RTP stream with the source IP address 216.179.125.5 by clicking it. Ensure the stream is highlighted. If you don't see the stream for 216.179.125.5, ensure you are working with the sip2.dump capture file, as described in Step 1.

4. Analyze the Selected Stream. With the top RTP stream selected, click the *Analyze* button to open the RTP Stream Analysis window, as shown in the following screen.



5. Extract Audio Payload. Next, click *Save Payload* to open the Save Payload As dialog box. Select a directory to save the file (such as your Desktop), click *.au* as the file format, click *Forward* for the channels selection, and enter a filename such as **voip-audio.au**. Be sure to include the *.au* filename extension! Click *OK* after entering the save preferences, as shown in the following screen.



6. Close Wireshark. Return to the Wireshark window and click *File, Quit* to close.
7. Play the Audio File. Using an audio player such as Windows Media Player or RealAudio, open the *voip-audio.au* file you generated in step 5. What is the content of the audio in this RTP stream? What is the decimal equivalent for the binary number 1011?

Wireshark Summary

- Wireshark adds a lot more analysis capability than Tcpdump offers alone
- Wireshark allows you to view protocol information
- Wireshark provides powerful filtering capability

SANS Security Essentials – © 2016 Secure Anchor Consulting, LLC

Wireshark Summary

This section intentionally left blank.

SECURITY 401 - SANS

Security Essentials

The End

SANS Security Essentials – © 2016 Secure Anchors Consulting LLC

This page intentionally left blank.