Linux Security:
File Encryption

This material is based on work supported by the National Science Foundation under Grant No. 0802551

National Science Foundation
WHERE DISCOVERIES BEGIN

Lesson Overview

With the advent of the Internet, social networking, and open communication, a vast amount of information is readily available on the Internet for anyone to access. Despite this trend, computer users need to ensure private or personal communications remain confidential and are viewed only by the intended party. Private information such as a social security numbers, school transcripts, medical histories, tax records, banking, and legal documents should be secure when transmitted online or stored locally.

One way to keep data confidential is to encrypt it. Militaries, governments, industries, and any organization having a desire to maintain privacy have used encryption techniques to secure information. Encryption helps to boost confidence in the security of online commerce and is necessary for secure transactions.

In this lesson, you will review encryption and examine several tools used to encrypt data. You will also learn to encrypt and decrypt data. Anyone who desires to administer computer networks and work with private data must have some familiarity with basic encryption protocols and techniques.

Objective

You should know what will be expected of you when you complete this lesson. These expectations are presented as objectives. Objectives are short statements of expectations that tell you what you must be able to do, perform, learn, or adjust after reviewing the lesson.

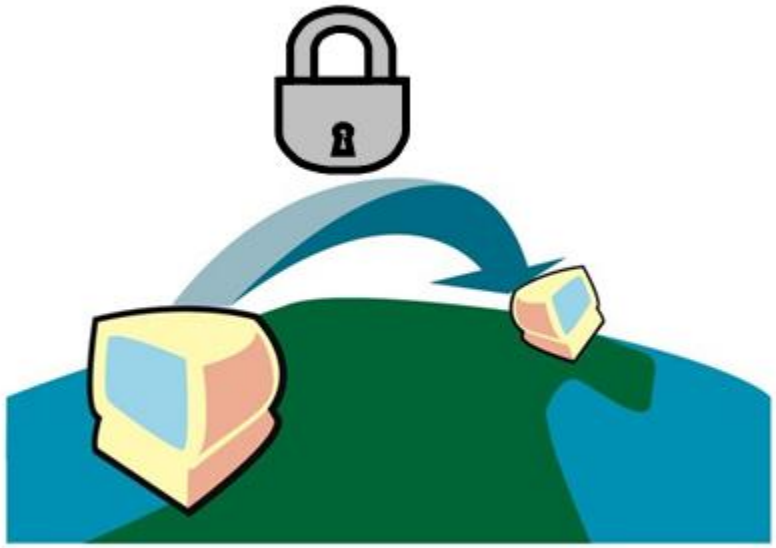
Lesson Objective:

Given the need to encrypt important data, the student will be able to assess three popular encryption tools for specific security needs and configure GPG to create secure archives and Truecrypt to lock and unlock encrypted directory trees as requested.

Lesson Outline

In this lesson, you will explore:

- File encryption
- PGP (Pretty Good Privacy)
- GPG (GNU Privacy Guard)
- Truecrypt - file tree encryption



Resources and Notes

This lesson uses Ubuntu Linux for demonstration. To complete this lesson successfully, you must have access to:

- Ubuntu Linux on bare metal or as a virtual install
- 10 Gb of hard drive space dedicated to the operating system's use
- A command shell
- Internet for research
- Word processor

Use the resources on the right to configure your system for Ubuntu if you have not yet done so.

Resources:

- Download Virtualbox
- Virtualbox for Linux Hosts
- Virtualbox manual
- Using Virtualbox with Ubuntu

Overview

Encryption goes back into history to the days of Julius Caesar. Recorded history suggests Caesar did not trust the messengers who carried his messages to military generals. So, in order to keep messages private, Caesar replaced every occurrence of an A in his messages with a D, every B with an E, and so on. Caesar's action is regarded as one of the first forms of encryption and is known as the "shift by 3" rule. Those on the receiving end of the messages had to understand the "shift by 3" rule in order to decipher or read his messages.

Today's encryption is much more complex. Had Caesar lived in today's world with computers and software, his messages would have been decrypted in less than 30 seconds because the pattern is easy to follow. A person who wanted to read Caesar's message would simply need to decrypt or discover the true meaning of a single word or letter and the rest of the message would be easily understood.

Why do we need encryption? Most people use it to maintain privacy and gain security. A vast collection of personal information is kept on computers and may be at risk of falling into the wrong hands, such as an identity thief. The financial consequences of that would be great. A company or organization can have the same problem if someone steals their data and used it for corporate espionage or exposed personal information of clients. To minimize these risks, companies should encrypt data before storing it and before transmitting it.

Required Reading

- Encryption
- Point of Encryption

What is Encryption & Decryption?

For this example, let's begin with a text file that may contain your banking records for the last year. Your accountant has asked you to email him those records to prepare your tax return. You correctly feel this information is very private and should be kept between you and your accountant.

After reviewing the data, you realize your banking history is stored in a text file, and if your email was intercepted, the interceptor would have access to your banking history. Files in text format are called "plaintext" or "cleartext" and their contents are visible to anyone opening the file.

The first step to securing the contents of the file and disguising it is called "encryption." When you encrypt the banking file, you use special software to scramble the contents into un-readable (to humans) gibberish called "ciphertext." Although the text appears to be gibberish when accessed by text editors, the contents are actually scrambled according to special rules and patterns. An associated software called a "key" knows the rules used to scramble the data and may be used to unscramble it when necessary.

Next, you email the ciphertext file to your accountant who has the necessary key to unlock this data and decrypt it back to cleartext. The process of reverting the ciphertext to plaintext is called *decryption*. The science behind this process is called *cryptography*.

Recommended Reading

- Cryptography

Encryption & Decryption (Contd)

Encryption and Decryption Process from PGP

Recommended Reading

- What is Cryptography?

Cryptography

Cryptography is the science of using mathematics to encrypt and decrypt data. When you want to store or transmit sensitive data across an insecure network (such as the Internet) and make sure it is not read by anyone other than the intended recipient, you should use cryptography. While you may not use cryptography directly, the software application you use to secure the data will use cryptography to do the job.

Cryptanalysis is the science of analyzing and breaking secure communication; it is the opposite of cryptology. Traditional cryptanalysis involves an interesting combination of analytical reasoning, application of mathematical tools, pattern finding, patience, determination, and pure luck. Most often, cryptanalysts are called *attackers* except when they are employed by a government—then they are called spies.

How well encryption holds up to an attacker is the measure of the strength of the encryption.

Recommended Reading

- Types of Data Encryption

How Does Encryption Work?

A cryptographic algorithm, or a cipher, is a mathematical function used by the cryptosystem (software tool) to encrypt and decrypt data. A cryptographic algorithm works in combination with a key – a word, number or phrase – to encrypt the plaintext.

The same plaintext encrypts to different ciphertext with different keys. In other words, if I had a key, and you had a key, and we both used our own keys to encrypt the data, we would both have different results.

The security of this encrypted data is dependent on two things: the strength of the cryptographic algorithm, and the secrecy of the key used to encrypt the data.

This cryptographic algorithm, plus all the possible keys, and all the protocols used in the process comprise a cryptosystem. PGP is one such cryptosystem.

Conventional Cryptography

Conventional cryptography is also called secret-key or symmetric-key encryption. One key is used for both the encryption and decryption of data. The data encryption standard (DES) employed by the United States Federal Government is an example of conventional cryptography.

The sender of the message has a key, and the receiver of the message has a copy of the same key. Take our earlier example of Caesar's messages. The sender (Caesar) knew that A=C, B=D, C=E, D=F, and so on through the alphabet. So, if Caesar wanted to write the word *SECRET*, he would use the key to generate a coded message that read: *VHFUHW*.

Now, when the receiving general received the coded message, he would need to use Caesar's key to decrypt the message. In other words, he would need to exchange each letter in the coded message by its correct equivalent that was three places removed in the alphabet so that V=S, H=E, F=C, U=R, H=E, W=T to get the word SECRET.

Recommended Reading

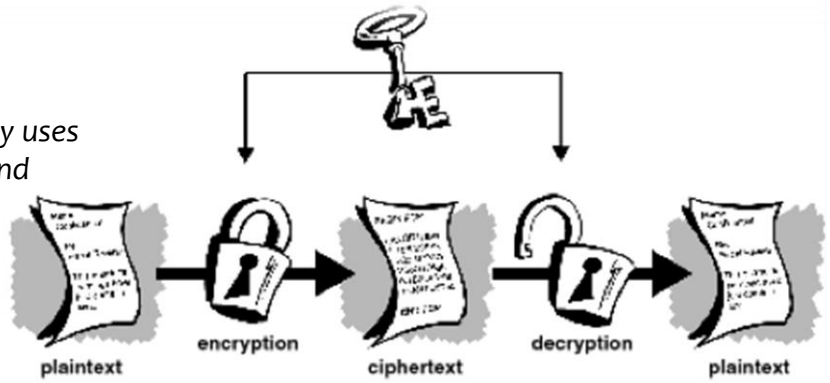
- Symmetric Key Encryption

Conventional Cryptography

Using the same key for encryption and decryption results in some problems. First, if the code needs to be changed, Caesar would have to dispatch a trusted messenger to deliver the new key to the generals in the field. Second, anyone who overheard the messenger telling the generals the new key would be able to decode the messages.

Conventional encryption (or single key encryption) has some benefits; it is extremely fast. It is also useful for encrypting data stored locally. But, as soon as the single-key method is used on data that needs to be transmitted, it becomes very expensive to secure the key distribution. How do you deliver the key to the general, or spy, or employee in the field in a secure manner?

Conventional cryptography uses the same key to encrypt and decrypt data.



Recommended Reading

- Encryption

Public Key Cryptography

Public Key Cryptography resolved the problems of key distribution. Whitfield Diffie and Martin Hellman introduced a concept in 1975 that used a pair of keys to encrypt and decrypt data. The idea is that there is a public key which is published to anyone in the world that encrypts the data. But, there is a single corresponding private key that is required to decrypt the data.

When you are ready to transmit and receive encrypted data you generate a pair of keys. One key is designated the public key and is stored on a key server or sent to all your friends and acquaintances. You keep and secure the private key.

Anyone in the world can use your public key to encrypt data to send to you, but only you have the private key required to decrypt that data.

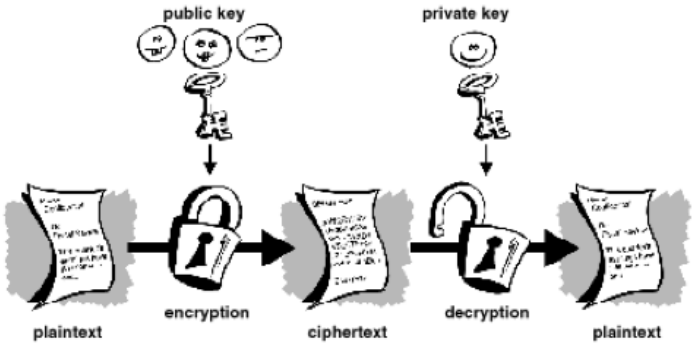
Recommended Reading

- Public Key Encryption

Public Key Cryptography

In our initial example of an accountant wanting your banking records, you would need a copy of his public key to encrypt that text file. Then you could send it over the Internet. If anyone intercepts or copies the file in transit, the text will look like garbage. When (and if) the accountant receives the file, he will use his private key to decrypt it, and he will have the banking records you transmitted to him.

The security of this transaction is based on how well the accountant protects his private key. Anyone with access to the public key can encrypt the information. Only the person having the private key can decrypt the file.



Cryptography using public and private keys

Recommended Reading

- Secure Communication

Types of Public Key Encryption

The primary advantage of public key encryption is that a person or company may send secure communication to someone with whom they do not have a pre-arranged encryption key. All you need to send secure communication is to find the public key, encrypt the data, and transmit it to the person with the private key. The need for a secure channel to exchange the public and private keys are all but eliminated. It is impossible for an attacker to generate the private key based on the public key, and there is never a circumstance where the private key needs to be transmitted over an insecure communication channel. There are a few types of public-key cryptosystems available:

- Elgamal invented by Taher Elgamal
- RSA invented by Ron Riverest, Adi Shamir, and Leonard Adleman
- Diffie-Hellman invented by Diffie and Hellman
- DSA (Digital Signature Algorithm) invented by David Kravitz

The advent of public key encryption took secure communications out of the hands of those who could afford to arrange for secure key transfer, and put it into the hands of the public. Secure communications used to be only be available to banks, large corporations, and governments, it is now available to anyone with access to a computer.

Social Effects of Public Key Encryption

It would be unwise to discuss the advantages of public key encryption without discussing the effects on society. Public key encryption promotes privacy to conduct business—legal or illegal—without the government or other organizations snooping on secure communication.

On the positive side, the availability of private key encryption allows secure communication between political organizations or groups to plan protests, rallies, and lobbying activities. Additionally, electronic communication have made it possible for these activities to be planned quickly without fear of interception.

On the negative side, criminals also have access to the same tools and a drug dealer, for instance, may use these tools to plan drug runs without the fear of wire-tapping and eavesdropping?

Everything has a positive and negative effect and society must balance these.

Recommended Reading

- Benefits and Disadvantages
- Cryptography

Encryption Tools

PGP, GPG, and True Crypt

What is PGP?

Many tools are available for encryption including three popular offerings:

- PGP
- GPG
- TrueCrypt

PGP was developed by Phil Zimmerman under the GNU Public License as copyrighted freeware. But because of patent royalty issues and the legal defense costs related to the USA export laws, he upgraded it to a proprietary program. The rights to this software has been traded around a few times. It is now owned by the PGP Corporation. The RSA algorithm patent has now expired and PGP still uses the IDEA encryption algorithm which is still patented in several countries.

GPG is a re-write of PGP with the code released under the GNU Public License. GPG does not use the IDEA encryption algorithm, so there are no royalties to pay and it is available for free.

Both PGP and GPG are equal in levels of encryption and are interchangeable with each other in most circumstances. You will find GPG on all versions of Linux, Unix, and OS/X. It is available for Windows as well. It is also available for most smartphones on the market today. In this lesson, PGP and GPG will be discussed together due to their similarity, and True Crypt will be handled on its own.

Recommended Reading

- GPG

What is PGP?

PGP, or *Pretty Good Privacy*, is a hybrid cryptosystem that includes the best features of conventional and public key cryptography.

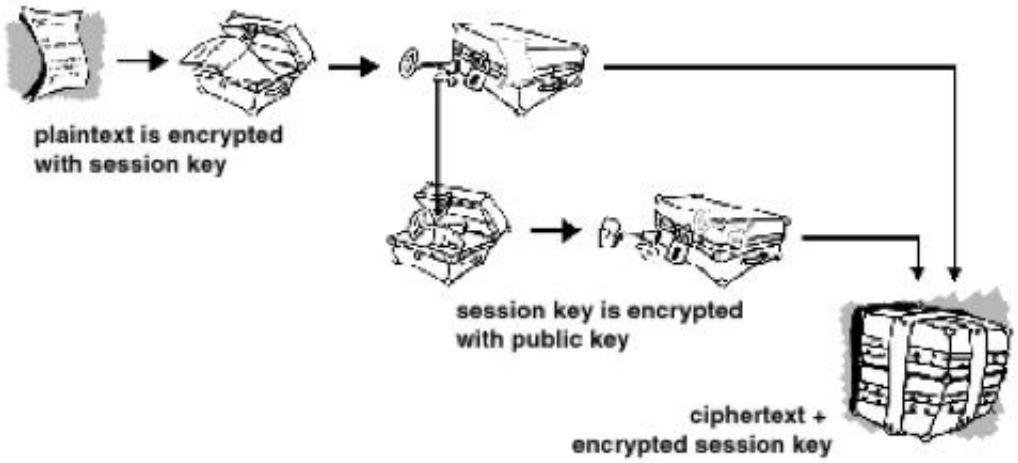
When PGP begins its encryption process, it first compresses the plaintext to make it smaller, to save bandwidth during the transmission process, and to strengthens the cryptographic security. The compression reduces the patterns that can be exploited during an attack and enhances the resistance to cryptoanalysis.

PGP then generates a session key, which is a one-time-only secret key. The session key is a random number generated based on the movement of your mouse or the keystrokes you type. The plaintext is then encrypted using an encryption algorithm to generate ciphertext. Once this ciphertext is generated, the session key is then encrypted to the recipient's public key. The public key-encrypted session key is transmitted along with the ciphertext to the recipient.

Required Reading

- How PGP Works

What is PGP? (Contd)



PGP security process

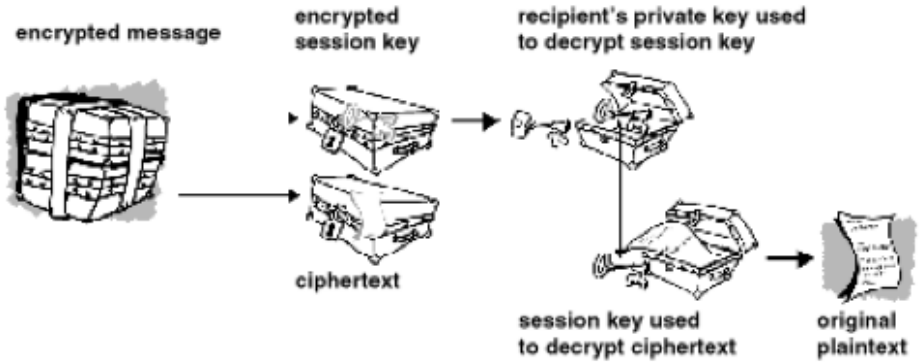
Required Reading

- PGP Use

PGP Decryption Process

Once the recipient receives the data, PGP works in reverse to change the ciphertext back to plaintext.

The recipient's copy of PGP uses his/her private key to recover the temporary session key, which PHP then uses to decrypt the conventionally-encrypted cipher-text. Then PGP expands the compressed data and the resulting file is the plaintext original.



Decryption process for PGP

PGP is considered a hybrid because its session-key could be considered a standard key and must be accessible to both parties. But, the session key is encrypted and transmitted along with the *session key encrypted* ciphertext in a way that is only accessible by the one person with the private key that matches the public key.

Conventional encryption is about 1,000 times faster than public key encryption. Public key encryption in turn, provides the solution to key distribution and data transmission issues. Used in harmony, performance and key distribution are improved without sacrificing security.

What is a Key?

A key is a hard-coded (or static) value that works with a cryptographic algorithm to produce specific cyphertext. Keys are basically very large numbers. Key size is measured in bits and the larger the size of the key, the more secure the ciphertext.

There is no relationship between the security of the public-key versus the security of a conventional-key and security.
Larger keys will tend to be secure for longer periods of time. Of course, how long it will take to break a key using the faster, and more efficient computers that are yet to come. When encryption first became popular, it was thought that a 56-bit key was considered safe, but with more powerful computers, larger keys are necessary.

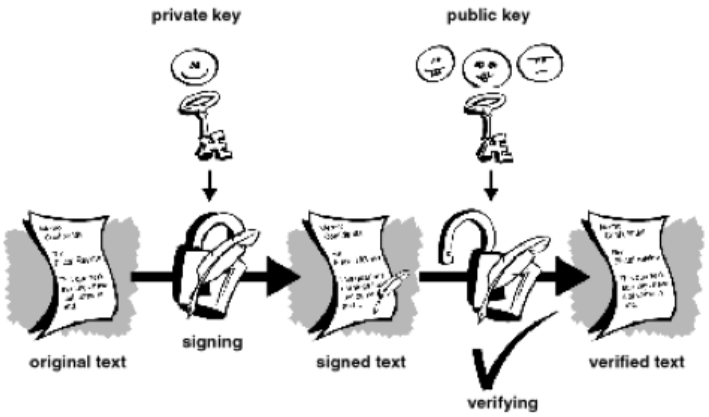
Keys are stored in an encrypted form. PGP will store the keys in two files on your hard-disk. These files are called "keyrings." One of these keyrings is for public keys of anyone to whom you send data. The other file holds your private keys, and these are required to de-crypt your data. If you lose your private keyring, you will be unable to decrypt information encrypted to keys on that ring.

Required Reading

- Encryption Key

What is a Digital Signature?

One of the many benefits of public key cryptography is that it provides a digital signature that allows a recipient to authenticate the source of the data, and it also verifies that the data was transmitted in an intact form. This public key digital signature provides authentication and data integrity. The digital signature also provides non-repudiation, which prevents the sender from claiming that he or she did not send the signed data.



The digital signature is used for the same purpose as a handwritten signature. However, a handwritten signature is easy to counterfeit. The digital signature is superior to the handwritten signature in that it is almost impossible to counterfeit, plus it attests to the contents of the information as well as to the identity of the signer.

When data is signed by you, it is encrypted with your private key (the one that ONLY you have a copy). If the data can be decrypted with your public key, it must have originated with you.

The first step of PGP is to make sure it is installed and then generate your key.

Installing PGP

Configuration and Updates

Updating System and Installing PGP

The PGP distributed with the Linux system is a free version called GNU PGP. This is the same software and the same level of encryption available for PGP, but it is free and the source code is available in accordance with Open Source licensing.

To install GNU PGP on your Ubuntu system, follow these directions:

1. Log into your Ubuntu system.
2. Open a terminal window.
3. Type: **sudo apt-get update**
4. Enter your password if requested.
5. Type: **sudo apt-get upgrade**
6. Accept any software packages and upgrades suggested.
7. Type: **sudo apt-get install gnupg2**
8. Notice the documents suggested at this point. We recommend you install them.

Instructions continued on next slide . . .

Select **PLAY** below for a video on configuring PGP.

View Video VideoLesson3ConfigPGP(C5L3S25).mp4

Updating System and Installing PGP

Continued from previous . . .

9. Type: Select **N** to cancel the install.
10. Press your up-arrow key once and change the line to read:
    sudo apt-get install gnupg2 gnupg-doc
9. Accept all dependencies that need to be installed.
10. Once complete type: **pgp**
11. If you get an error message about command not found type:
    sudo apt-get install pgpgpg
9. Accept dependencies that need to be installed.

So, what did we just do?

We updated your package list and then installed upgraded packages. Then, we installed the GNU PG package and tested it. If for some reason the link was not made to PGP, we installed the package PGPGPG that provides a link to GPG from the PGP command.

Next we will setup and generate your keys.

Required Reading

- Setting Ubuntu with GPG
- GNU Privacy

Generate Public / Private Keys

The first step of our encryption process is to generate public and private keys. Follow these directions: (*Note: We will NOT be using sudo in these commands; the encryption keys must be generated from your own user ID.)*

1. Open a terminal window, type **cd**
2. Type: **gpg --gen-key** (To initialize your data directory; look at the messages and take note of where your keychains will be stored.)
3. Type: **gpg --gen-key**
4. Accept the default for type of key when asked.
5. Type the maximum suggested key size when asked.
6. Enter zero (**0**) for how long the key should be valid (does not expire).
7. Type your real name and email address when prompted to generate the keys' USER ID.
8. Enter a passphrase that will protect your private key in case someone gets access to your terminal. DO NOT FORGET this passphrase or your private key will not work!
9. When prompted, move the mouse, surf the web, type on your keyboard to give the application some random numbers to use. You can type anything, but be sure to include a number.
10. Type: **ls -l ~/.gnupg** to see your keyrings
11. Type: **gpg --fingerprint**

Select **PLAY** below for a video on generating encryption keys.

View Video VideoLesson3GenerateKeys(C5L3S27).mp4

Required Reading

- Generating PGP Keys

Export Public Keys

Your public key fingerprint and key information should now be showing. We will now make your public key available on a key server.

1. Type: **gpg -output username.gpg -export emailaddress**
2. In the command above, substitute **username** with the user name you used to generate the key on the previous screen. Substitute **emailaddress** with the same email address you used to generate the key on the previous screen.
3. Type: **gpg -armor -export emailaddress**
4. Again, substitute **emailaddress** with the email address you used previously to generate the key.
5. Type: **gpg -list-keys**

Select **PLAY** below for a video on exporting public keys.

View Video VideoLesson3PublicKey (C5L3S28).mp4

Required Reading

- Generating PGP Keys

How to Encrypt a File?

One of the primary purposes of GPG is to encrypt files for transmission and decrypt files received. Follow these directions to encrypt a file:

1. Open a terminal window.
2. Type: **vi testfile.txt**
3. Enter some text in this document using the insert mode of vi. You can even cut and paste from a web site.
4. Type: **gpg --output testfile.gpg --encrypt --recipient emailaddress doc**
5. Substitute the **emailaddress** in the line above with your email address (normally the email address of the recipient). In this example, we are using the public key from our own user id.
6. When complete, type: **ls –l**
7. Notice the size difference between both the encrypted (.gpg) and the unencrypted files.

Use vi and open the encrypted file. What do you see? It is important to always include your own email on the recipient's list otherwise, you will not be able to decrypt the file generated.

Now, we need to learn to unencrypt a file.

Select **PLAY** below for a video on encrypting a file.

View Video VideoLesson3EncryptFile(C5L3S29).mp4

Recommended Reading

- Encrypting and Decrypting

How to Decrypt a File?

In this example, we will decrypt the testfile.gpg . Follow these directions:

1. From your terminal window, type: **rm testfile.txt**
2. Type: **ls –l** (verify the file is gone)
3. Type: **gpg --output testfilenew.txt --decrypt testfile.gpg**
4. Type: **vi testfilenew.txt** (verify that the file is now readable)

The file will only be decrypted if you have the private key for the public key that was used on the recipient's line of the gpg encrypt command.

We now know how to encrypt and decrypt files. In the next series of slides, we will learn to encrypt an entire directory using a new tool called TrueCrypt.

Select **PLAY** below for a video on decrypting a file.

View Video VideoLesson3DecryptFile(C5L3S30).mp4

Recommended Reading

- Encrypting and Decrypting

Introducing TrueCrypt

Encrypting / Decrypting Directories

What is TrueCrypt?

PGP and GPG allow you to encrypt a single document, picture, or archive for transmission over the Internet, or for safe keeping locally. But what if you wanted a directory or a file tree to be encrypted for on-the-fly or real-time privacy?

TrueCrypt is an open source software package that helps you setup and maintain this real-time encrypted volume (or directory structure). On-the-fly encryption means that data is automatically encrypted or decrypted right before it is loaded or saved without the need for the user to enter command line commands or load software and keys.

To access files encrypted by TrueCrypt, the user must have a keyfile and/or password to access the data stored on the encrypted volume. TrueCrypt encrypts the entire filesystem including file names, folders, folder names, and the content of each file within it.

While our example in this lesson will show how to configure TrueCrypt for Linux, it is available for Windows and OS/X.

Important note: *Do not lose the keyfile or password because without these you are unable to access your data.*

Required Reading

- TrueCrypt on Ubuntu

What is TrueCrypt?

TrueCrypt is not available for install as a Debian package and the development team has requested that distribution of TrueCrypt be done from its web site at http://www.truecrypt.org.

To install TrueCrypt, complete the following:

1. From the web browser on your Linux machine, go to: http://www.truecrypt.org/downloads and download the appropriate tar.gz file for your system.
2. Open a terminal window
3. Type: **cd Downloads**
4. Type: **tar xvzf truecrypt-7.0a-linux-x86.tar.gz**
5. Type: **sudo ./truecrypt-7.0a-setup-x86**
6. Follow the prompts for installation and accept all the defaults.

Now, Truecrypt is installed on your system. Next, we are going to create a volume on which to store private information.

Select **PLAY** below for a video on TrueCrypt.

View Video VideoLesson3TrueCrypt (C5L3S33).mp4

Required Reading

- Ubuntu & TrueCrypt

Creating and Mounting a Volume

TrueCrypt can encrypt a full hard-drive volume, or it can create a virtual volume that is created inside a file on your hard drive. In our example, we are going to use a virtual volume that lives within our home directory. Follow these directions:

From your main GUI window, select **Applications →Accessories→ TrueCrypt**

1. Click **Create Volume**
2. Click **Create Encrypted File Container**
3. Click **Standard True-Crypt Volume**
4. Click on **Select File**
5. Give the file a name and select the directory you want to use.
6. Click **Save** and then **Next**
7. On **Encryption Options** accept the defaults and click **Next**
8. Give it a size, in our case use **5 GB**
9. Click **Next**
10. Enter a volume password. If under 20 characters, you will receive some warning messages, click **OK** on them.
11. Click Next.
12. On **Large Files**, accept the default of small files (< 4GB) and click **Next**
13. On format options, choose **Ext3** from the choices.

Required Reading

- Configure TrueCrypt

Creating and Mounting a Volume

Continued from previous . . .

14. Click **Next**
15. Select "**I will mount the volume on other devices**" and click **Next**
16. Click **Format** when the button becomes available.
17. When formatting is complete, the software will ask you to enter your user password (NOT THE DISK PASSWORD). The user password is needed to run the "mount" command.
18. At the *Volume Created* screen, click **Exit**.
19. You will be back at the TrueCrypt Screen.
20. Click on **Slot 1**
21. Click on **Select File**
22. Select the disk volume file you just created.
23. Click **Mount**.
24. Enter your *Disk password* and your *user password* when prompted.
25. You will see your volume on the file browser and on your desktop.
26. Use the directory like you would any other drive or directory structure.
27. To unmount the drive, return to TrueCrypt using the icon on the top left status bar of your GUI and click on the volume and select **Dismount**.

Remember, if you lose the disk password, you cannot access the data on the drive. The encrypted directory can be copied to a flash drive, external drive, or other storage media. The data can be used on another computer but only if the user has the TrueCrypt password to access it. Otherwise, the data is secure. No-one can access the data without your password—at least . . . not yet!

Lesson Summary

During this lesson, we discussed cyphers, encryption methods, and some of the tools used to provide data encryption. The importance of data encryption is both financial and social. People need to be able to have some privacy, and by nature electronic forms of communication are not private.

We also installed and explored the GNU PGP package. We explained that PGP and GPG are similar tools with the exception that GPG is open source while PGP is not. Additionally, GPG is free while PGP is no longer free for anything except the use of the command line.

In our GPG demonstration, you learned to generate a public/private key pair and export that public key in both binary and ASCII format. We showed you the location of one public keyserver at http://pgp.mit.edu and its associated search functionality. Then we showed you how to encrypt and decrypt files using GPG.

We also showed you how to install and use TrueCrypt, an encryption utility used to encrypt and decrypt entire directory trees (or file systems). TrueCrypt volumes or directories can be moved from one machine to another, but you need the keyfile or disk password to access the data.

File encryption is only as good as the security of the computer with the private key. If for some reason, you have spyware running on the machine that stores your password or keyfiles, your data is no longer secure and encryption will only give you a false sense of security. It is also important to protect the private key and password. If either is compromised, an attacker will be able to access your data at will.