Hands-On Ethical Hacking and Network Defense, Edition 4

Chapter 12: Cryptography

Module Objectives

- By the end of this module, you should be able to:
  - Summarize the history and principles of cryptography
  - Describe symmetric and asymmetric encryption algorithms
  - Explain public key infrastructure (PKI)
  - Describe possible attacks on cryptosystems
  - Compare hashing algorithms and how they ensure data integrity

Understanding Cryptography Basics

- Cryptography
  - Process of converting **plaintext** into **ciphertext**
    - Plaintext: Readable text
    - Ciphertext: Unreadable or encrypted text
  - Used to hide data from unauthorized users
- Decryption
  - Process of converting ciphertext back to plaintext

History of Cryptography (1 of 2)

- Has been around for thousands of years
  - Some Egyptian hieroglyphics were encrypted
  - Parts of the Book of Jeremiah were written using a **cipher**
- **Substitution cipher**
  - Replaces one letter with another letter
    - Based on a key
  - Example: Julius Caesar’s cipher
    - Shifted each letter of the alphabet three positions

History of Cryptography (2 of 2)

- **Cryptanalysis**
  - Study of breaking encryption algorithms
  - When a new encryption algorithm is developed, cryptanalysis is used to ensure that breaking the code is impossible
    - Or it would take so much time and so many resources that the attempt would be impractical
  - When cryptanalysis is feasible with a reasonable amount of computing power:
    - An attack on the algorithm is deemed “practical”
      - And the algorithm is considered weak

The War Machines (1 of 2)

- Enigma machine
  - Most famous encryption device
  - Developed by Arthur Scherbius
  - Used by Germans during World War II
  - Most books on cryptography discuss this device
  - Enigma substituted each letter typed by the operator
    - Substitutions were computed using a key and set of switches or rotors
    - When the message was completely encrypted, it was transmitted over the airwaves
  - Code was broken first by Polish cryptographers, and then the British and Americans
    - The machine British and American cryptologists used for breaking the code was called the Bombe

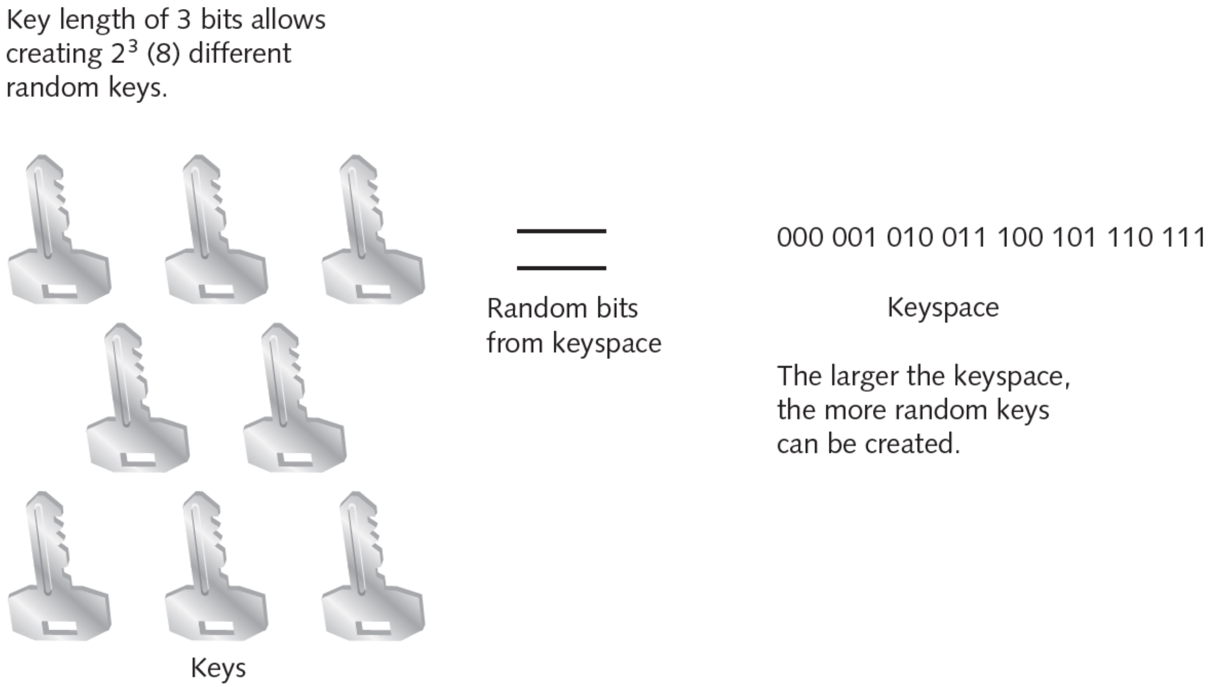
The War Machines (2 of 2)

- The Purple Machine
  - Developed by Japanese during World War II
  - Used techniques discovered by Herbert O. Yardley
  - Code was broken by William Frederick Friedman
    - Known as the Father of U.S. Cryptanalysis
- **Steganography**
  - Process of hiding data in plain view in pictures, graphics, or text

Understanding Symmetric and Asymmetric Algorithms (1 of 2)

- **Encryption algorithm**
  - Mathematical function or program that works with a key
    - The algorithm’s strength and key’s secrecy determine the security of the data
- **Key**
  - Sequence of random bits generated from a range of allowable values called a **keyspace,** which is contained in the algorithm
    - The larger the keyspace, the more keys that can be created
    - The more random keys that can be created, the more difficult it is for hackers to guess the key that was used to encrypt the data
- **Cryptosystem**
  - Converts between plaintext and ciphertext
    - No matter how strong the algorithm or how large the keyspace, if the key isn’t protected, an attacker can decrypt the message

Understanding Symmetric and Asymmetric Algorithms (2 of 2)



Key length of 3 bits allows creating 2^3 (8) different random keys.

The larger the keyspace, the more random keys can be created.

Source: Cengage Learning

Figure 12-1 Selecting random keys from a keyspace

Symmetric, Asymmetric, and Hashing Algorithms

Type of algorithm	Description
Symmetric	Uses a single key to encrypt and decrypt data. Both the sender and receiver must agree on the key before data is transmitted. Symmetric algorithms support confidentiality but not authentication and nonrepudiation (covered later in “Asymmetric Algorithms”). However, they’re at least 1000 times faster than asymmetric algorithms.
Asymmetric	Uses two keys: one to encrypt data and one to decrypt data. Asymmetric algorithms support authentication and nonrepudiation but are slower than symmetric algorithms. Asymmetric algorithms are also known as public key cryptography.
Hashing	Used for verification. Hashing takes a variable-length input and converts it to a fixed-length output string called a hash value or message digest.

Symmetric Algorithms (1 of 8)

- Cryptosystems using **symmetric algorithms** have one key that encrypts and decrypts data
  - Advantages
    - Faster than asymmetric algorithms
    - Difficult to break if a large key size is used
    - Only one key needed to encrypt and decrypt data
  - Disadvantages
    - Require each pair of users to have a unique secret key
      - Makes key management a challenge
    - Difficult to deliver keys without risk of theft
    - Does not support authentication and nonrepudiation for users

Symmetric Algorithms (2 of 8)

- Types of symmetric algorithms
  - **Stream ciphers**
    - Operate on plaintext one bit at a time
  - **Block ciphers**
    - Operate on blocks of bits
      - These blocks are used as input to mathematical functions that perform substitution and transposition of the bits
- **Data Encryption Standard (D E S)**
  - The National Institute of Standards and Technology (N I S T):
    - Wanted a means of protecting sensitive but unclassified data
    - Invited vendors in 1970 to submit encryption algorithms

Symmetric Algorithms (3 of 8)

- Data Encryption Standard (cont’d)
  - IBM proposed Lucifer
    - A 128-bit encryption algorithm
    - The National Security Agency (NSA) reduced key size to 64 bits and created **Data Encryption Algorithm (DEA)**
  - 1988: NSA thought the standard was at risk of being broken
    - Because of its longevity and the increasing power of computers
  - 1998: A computer system was designed to break the encryption key in only three days

Symmetric Algorithms (4 of 8)

- **Triple Data Encryption Standard (3 D E S)**
  - Served as a quick fix for D E S vulnerabilities
  - Performed the original D E S computation three times with different keys
    - Made it much stronger than D E S
  - Takes longer to encrypt and decrypt data than D E S

Symmetric Algorithms (5 of 8)

- **Advanced Encryption Standard**
  - N I S T put out request for a new encryption standard
    - Required submissions for a symmetric block cipher capable of supporting 128-, 192-, and 256-bit keys
  - Five finalists
    - Rijndael (winner)
    - MARS
    - RC6
    - Serpent
    - Twofish

Symmetric Algorithms (6 of 8)

- International Data Encryption Algorithm
  - Block cipher that operates on 64-bit blocks of plaintext
  - Uses 128-bit key
  - Developed by Xuejia Lai and James Massey
  - Designed to work more efficiently in computers used at home and in businesses
  - Free for noncommercial use

Symmetric Algorithms (7 of 8)

- **Blowfish**
  - Block cipher that operates on 64-bit blocks of plaintext
  - Key length can be as large as 448 bits
  - Developed as a public-domain algorithm by Bruce Schneier
- RC4
  - Most widely used stream cipher
  - Used in WEP wireless encryption
  - Finding the key with air-cracking programs is easy
  - Created by Ronald L. Rivest in 1987 for RSA Security

Symmetric Algorithms (8 of 8)

- **RC5**
  - Block cipher
  - Operates on different block sizes: 32, 64, and 128
  - Key size can reach 2048 bits
  - Created by Ronald L. Rivest in 1994 for RSA Security

Asymmetric Algorithms (1 of 6)

- Use two mathematically related keys
  - Data encrypted with one key can only be decrypted with the other
- Also called **public key cryptography**
  - **Public key**: Openly available
  - **Private key:** Secret key known only by the key owner
- Different ways to encrypt a message with asymmetric algorithms, depending on whether the goal is to provide authentication and nonrepudiation
  - **Authentication** verifies that the sender or receiver (or both) is who they claim to be
  - **Nonrepudiation** ensures that the sender or the receiver cannot deny sending or receiving the message

Asymmetric Algorithms (2 of 6)

- How it works
  - User A encrypts a message with a private key and sends it to User B
    - User B decrypts the message with User A’s public key
    - A user’s private and public keys are mathematically related
- If confidentiality is a major concern for User A:
  - User A encrypts a message with User B’s public key
  - User A can assure User B about the authenticity of the message by encrypting the message with their private key

Asymmetric Algorithms (3 of 6)

- RSA
  - Published in 1978 by Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman
  - First algorithm used for both encryption and digital signing
  - Still widely used, particularly in e-commerce
  - Many browsers using the Transport Layer Security (TLS) protocol use the RSA algorithm
    - Based on the difficulty of factoring large numbers
  - Uses a one-way function to generate a key
    - A mathematical formula that is easy to compute in one direction
      - Difficult or impossible to compute in the opposite direction

Asymmetric Algorithms (4 of 6)

- Diffie-Hellman
  - Developed by Whitfield Diffie and Martin Hellman
  - Does not provide encryption
  - Used to establish one secret key shared between two parties
    - Each party generates a shared key based on a mathematical key–agreement relationship
  - If a key is intercepted during transmission, network is vulnerable to attack
    - With a method of sharing a secret key, users can secure their electronic communication without fear of interception

Asymmetric Algorithms (5 of 6)

- Elliptic curve cryptography (ECC)
  - Used for:
    - Encryption
    - Digital signatures and key exchange
  - Efficient algorithm that requires only few resources
    - Memory
    - Disk space
    - Bandwidth
  - Good candidate for wireless devices and cell phones

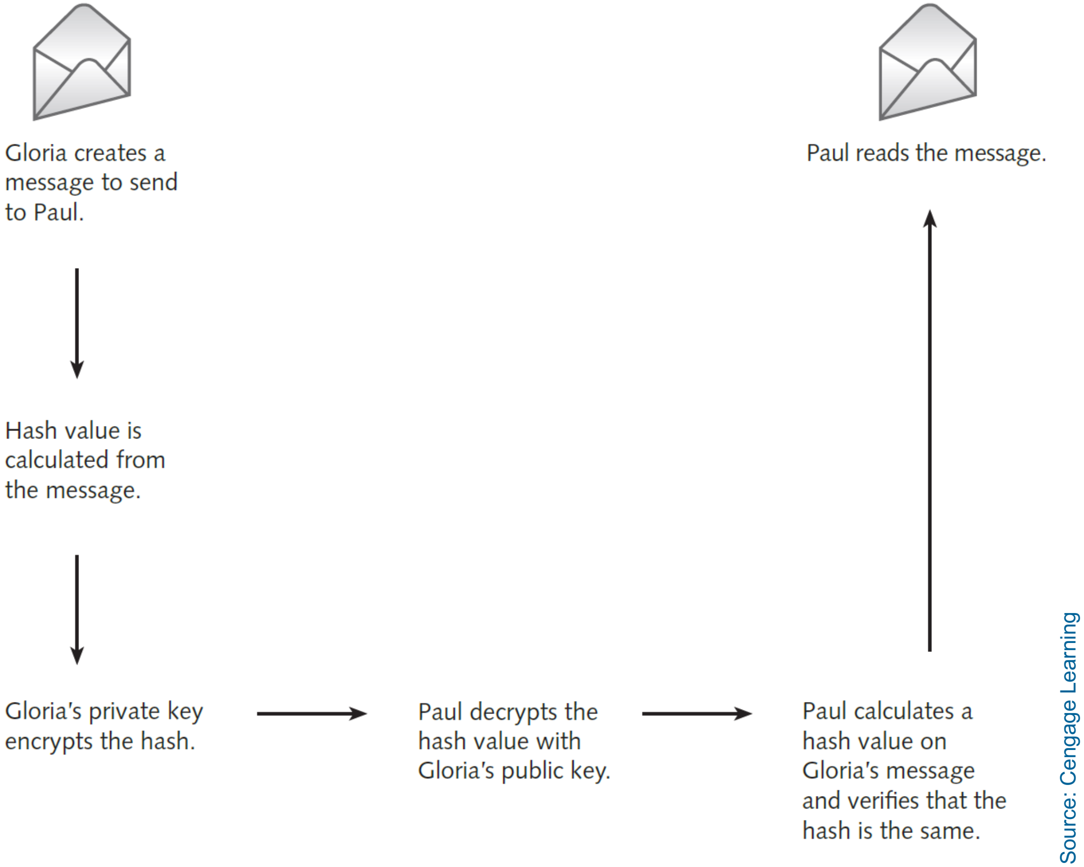
Asymmetric Algorithms (6 of 6)

- ElGamal
  - Used to:
    - Generate keys
    - Encrypt data
    - Create digital signatures
  - Developed by Taher Elgamal in 1985
    - Uses discrete logarithm problems that are complex to solve
      - Solving a discrete logarithm problem can take many years and require CPU-intensive operations

Digital Signatures (1 of 6)

- Asymmetric algorithms
  - Enable a public key to decrypt a message encrypted with a private key or vice versa
  - A public key can only decrypt a message encrypted with the corresponding private key

Digital Signatures (2 of 6)


Gloria creates a message to send to Paul.
Hash value is calculated from the message.
Gloria's private key encrypts the hash.
Paul decrypts the hash value with Gloria's public key.
Paul calculates a hash value on Gloria's message and verifies that the hash is the same.
Paul reads the message.

Source: Cengage Learning

Figure 12-2 Using a digital signature

Digital Signatures (3 of 6)

- Digital Signature Standard
  - Established by N I S T in 1991
    - To ensure that digital signatures can be verified
  - Federal government requirements
    - RSA and Digital Signature Algorithm (DSA) must be used for all digital signatures
    - Hashing algorithm must be used to ensure message integrity
      - N I S T requires using Secure Hash Algorithm (SHA)

Digital Signatures (4 of 6)

- **Pretty Good Privacy (PGP)**
  - Developed by Phil Zimmerman as a free email encryption program
    - Allowed typical users to encrypt email messages
  - Zimmerman was almost arrested for this innovation in the mid-1990s
    - Any kind of “unbreakable” encryption was seen as a weapon
    - Sharing it was compared with selling arms to the enemy
  - The Internet standard for PGP messages is now called **OpenPGP**
    - Uses certificates similar to those in PKI
      - Does not use a centralized CA
      - Verification of CA is not as efficient as PKI

Digital Signatures (5 of 6)

- Pretty Good Privacy (cont’d)
  - Algorithms supported by OpenPGP:
    - AES
    - IDEA
    - RSA
    - DSA
    - SHA

Digital Signatures (6 of 6)

- **Secure Multipurpose Internet Mail Extension (S/MIME)**
  - Another public key encryption standard
    - Used to encrypt and digitally sign email
    - Can encrypt email messages containing attachments
    - Can use PKI certificates for authentication
    - Widely used for email encryption
      - Built into Microsoft Outlook

Sensitive Data Encryption

- Make it a policy to exchange test results and sensitive documents in encrypted form
  - Recommend doing so to clients as well
- Organizations might also need to encrypt **data at rest**
  - Data at rest
    - Data not moving through the network or being used by OS
    - Refers to data stored on workstations, servers, smartphones, removable drives, backup media, and laptop computers

Hashing Algorithms

- Hashing takes a variable-length message and produces a fixed-length hash value (called a **message digest**)
  - Used to verify integrity of the data or message
  - In a sense, it is like a fingerprint of the message
  - If message is changed, hash value also changes
    - The recipient knows that the original message changed during transmission
- Collisions
  - Two different messages producing the same hash value results in collision
    - A good hashing algorithm is one that’s resistant to collisions

Hashing Algorithms (1 of 2)

Algorithm	Description
MD2	Developed by Ronald L. Rivest in 1989, this algorithm was optimized for 8-bit machines.
MD4	Developed by Rivest in 1990. Using a PC, collisions in this version can now be found in less than 1 minute. Microsoft Windows still uses RC4 to store password hashes.
MD5	Developed by Rivest in 1991. It was estimated in 1994 that creating a computer that could find collisions with brute-force attacks would cost $10 million. However, a collision for an MD5 hash can now be found with just a few machines within a few hours.
MD6	Developed by Rivest and his team at MIT in 2008. It uses a Merkle tree-like structure to allow for immense parallel computation of hashes for very long inputs. Speeds in excess of 1 GB/s have been reported as possible for long messages on 16-core CPU architecture. MD6 is not widely used.
SHA-1	SHA-160, commonly known as SHA-1, has been considered broken since 2005 but is now approaching the date when collision attacks will begin to become available. It uses a 160-bit digest and is, at the time this was written, found in many applications in the government and private sector.

Hashing Algorithms (2 of 2)

Algorithm	Description
SHA-2	A collective designation for the longer digest versions of SHA algorithms: SHA-224, SHA-256, SHA-384, and SHA-512. SHA-2 versions use essentially the same algorithm as SHA-160, but the longer digests make collisions harder to find.
SHA-3	Keccak was selected as the SHA-3 standard by N I S T in 2015. Keccak, or SHA-3, can take an input of any size and create an output of any size. SHA-3 is not intended to immediately replace SHA-2 and varies greatly in design from its SHA predecessors. You can read more about the Keccak algorithm on its website (keccak.noekeon.org).

Knowledge Check Activity 12-1

Digital signatures are used to do which of the following?

a. Verify that a message was received
b. Ensure that repudiation is provided
c. Provide authentication and nonrepudiation
d. Encrypt sensitive messages

Knowledge Check Activity 12-1: Answer

Digital signatures are used to do which of the following?

Answer: c. Provide authentication and nonrepudiation

Digital signatures are used to provide authentication and nonrepudiation. In a digital signature, the hash calculated from the message content is encrypted with a private key to ensure authentication and nonrepudiation.

Knowledge Check Activity 12-2

OpenPGP is focused on protecting which of the following?

a. Web content

b. Email messages

c. Database systems

d. IPSec traffic

Knowledge Check Activity 12-2: Answer

OpenPGP is focused on protecting which of the following?

Answer: b. Email messages

OpenPGP is focused on protecting email messages. It is the Internet standard for Pretty Good Privacy (PGP) messages. PGP is a free email encryption program that allows typical users to encrypt email messages.

Discussion Activity 12-1

Discuss about hashing algorithms and hash values.

Discussion Activity 12-1: Answer

Discuss about hashing algorithms and hash values.

Answer: Used for verification

Explanation: Hashing algorithms are used for verification. Hashing takes a variable-length input and converts it to a fixed-length output string called a hash value or message digest.

.

Understanding PKI (1 of 10)

- **PKI**: Structure consisting of programs, protocols, and security protocols for encrypting data
  - Uses public key cryptography to protect data transmitted over the Internet
- Components of PKI
  - **Certificate**
    - Digital document that that the two parties exchanging data over the Internet are really who they claim to be
  - Public keys
    - Issued by a **certification authority (CA)**
  - A certificate that a trusted CA issues binds a public key to the identity of the organization or individual that purchased it

Understanding PKI (2 of 10)

- Expiring, revoking, and suspending certificates
  - A certificate issued by a CA is assigned a period of validity
    - After that date, the certificate expires
    - Certificate can be renewed with a new expiration date
      - If keys are still valid and remain uncompromised
  - Reasons to suspend or revoke a certificate
    - User leaves the company
    - Hardware crash causes a key to be lost
    - Private key is compromised
    - Company no longer exists or supplied false information

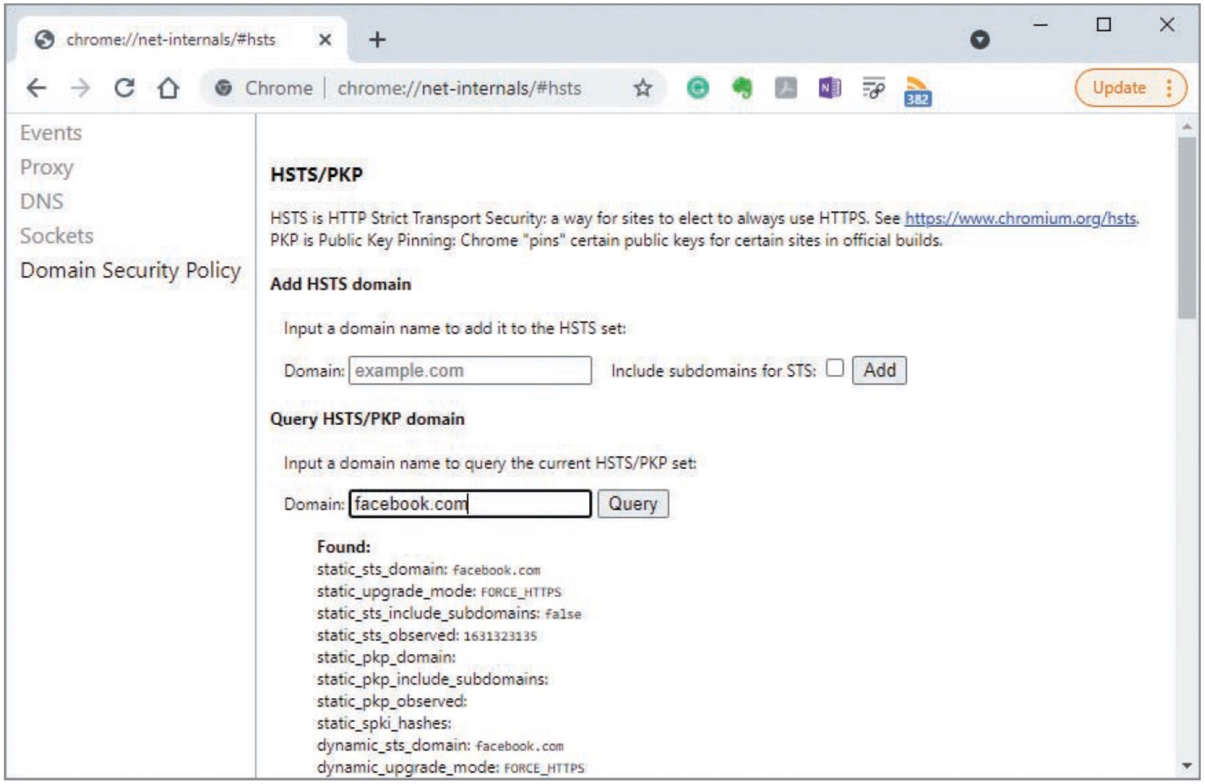
Understanding PKI (3 of 10)

- Expiring, revoking, and suspending certificates (cont’d)
  - The CA compiles a certificate revocation list (CRL)
    - Contains all revoked and suspended certificates
  - Suspension of a certificate occurs when:
    - One or more parties fail to honor agreements set forth when the certificate was issued
      - Makes it easier to restore if parties come to an agreement later

Understanding PKI (4 of 10)

- **HTTP Strict Transport Security (HSTS)**
  - Created in 2012 as a mechanism for web servers to tell clients they require secure communications
  - Forces all traffic between a web browser and a web server to be sent over a secure channel
  - If the web server’s certificate cannot be validated by the web browser, it will disallow access to the website

Understanding PKI (5 of 10)


chrome://net-internals/#hsts
Chrome | chrome://net-internals/#hsts
382
Update
Events
Proxy
DNS
Sockets
Domain Security Policy

HSTS/PKP

HSTS is HTTP Strict Transport Security: a way for sites to elect to always use HTTPS. See https://www.chromium.org/hsts.
PKP is Public Key Pinning: Chrome "pins" certain public keys for certain sites in official builds.

Add HSTS domain

Input a domain name to add it to the HSTS set:

Domain: example.com Include subdomains for STS: □ Add

Query HSTS/PKP domain

Input a domain name to query the current HSTS/PKP set:

Domain: facebook.com Query

Found:
static_sts_domain: facebook.com
static_upgrade_mode: FORCE_HTTPS
static_sts_include_subdomains: false
static_sts_observed: 1631323135
static_pkp_domain:
static_pkp_include_subdomains:
static_pkp_observed:
static_spki_hashes:
dynamic_sts_domain: facebook.com
dynamic_upgrade_mode: FORCE_HTTPS

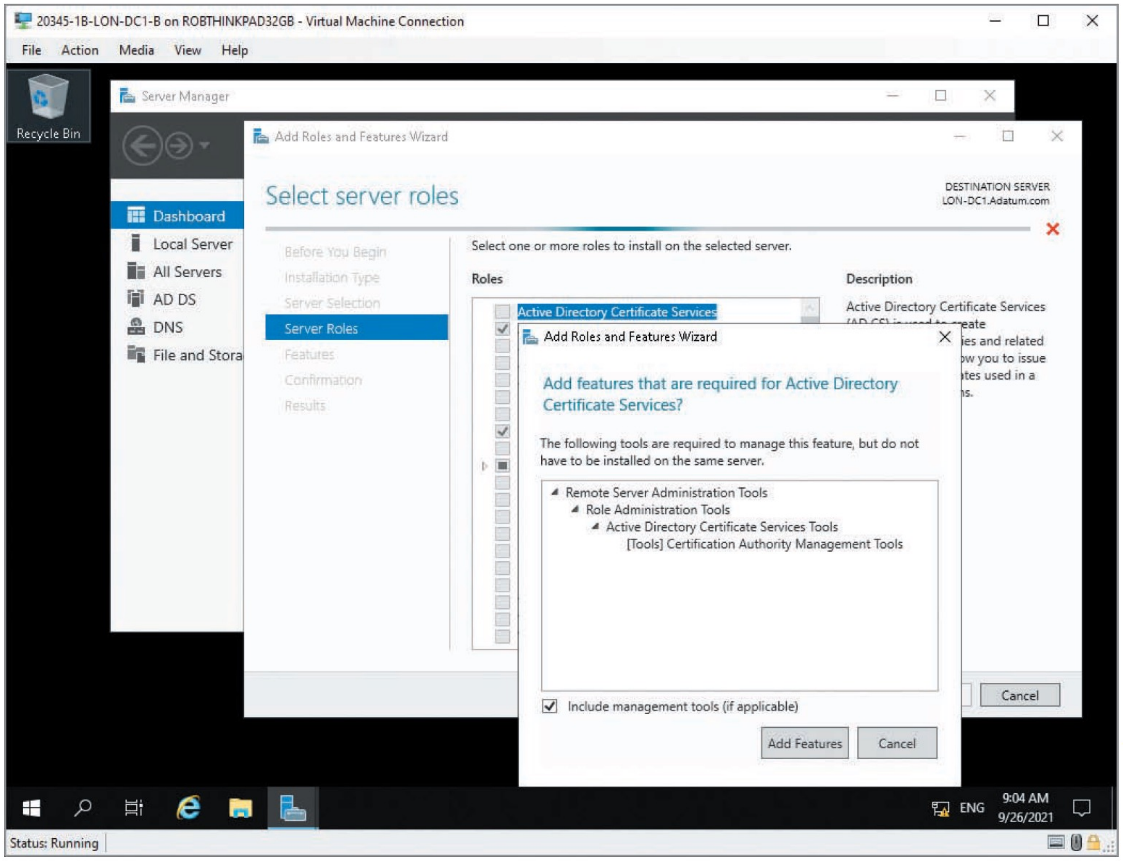
Source: Google Inc.

Figure 12-3 Viewing HSTS/PKP information in Google Chrome

Understanding PKI (6 of 10)

- Backing up keys
  - If keys are destroyed and not backed up, encrypted business-critical information might be irretrievable
  - Companies typically back up their keys and store them offline in a safe or vault
- Microsoft root CA
  - Includes features in its server OSs for configuring a server as a CA
    - Instead of using a third-party CA
  - Example: Using the Add Roles and Features Wizard in Windows Server 2019
    - Administrator can select Active Directory Certificate Services

Understanding PKI (7 of 10)



Select server roles

DESTINATION SERVER
LON-DC1.Adatum.com

Select one or more roles to install on the selected server.

Roles

☐ Active Directory Certificate Services

Description

Active Directory Certificate Services (AD CS) is used to create ...ies and related ...w you to issue ...tes used in a ...s.

Add Roles and Features Wizard

Add features that are required for Active Directory Certificate Services?

The following tools are required to manage this feature, but do not have to be installed on the same server.

- Remote Server Administration Tools
  - Role Administration Tools
    - Active Directory Certificate Services Tools
      - [Tools] Certification Authority Management Tools

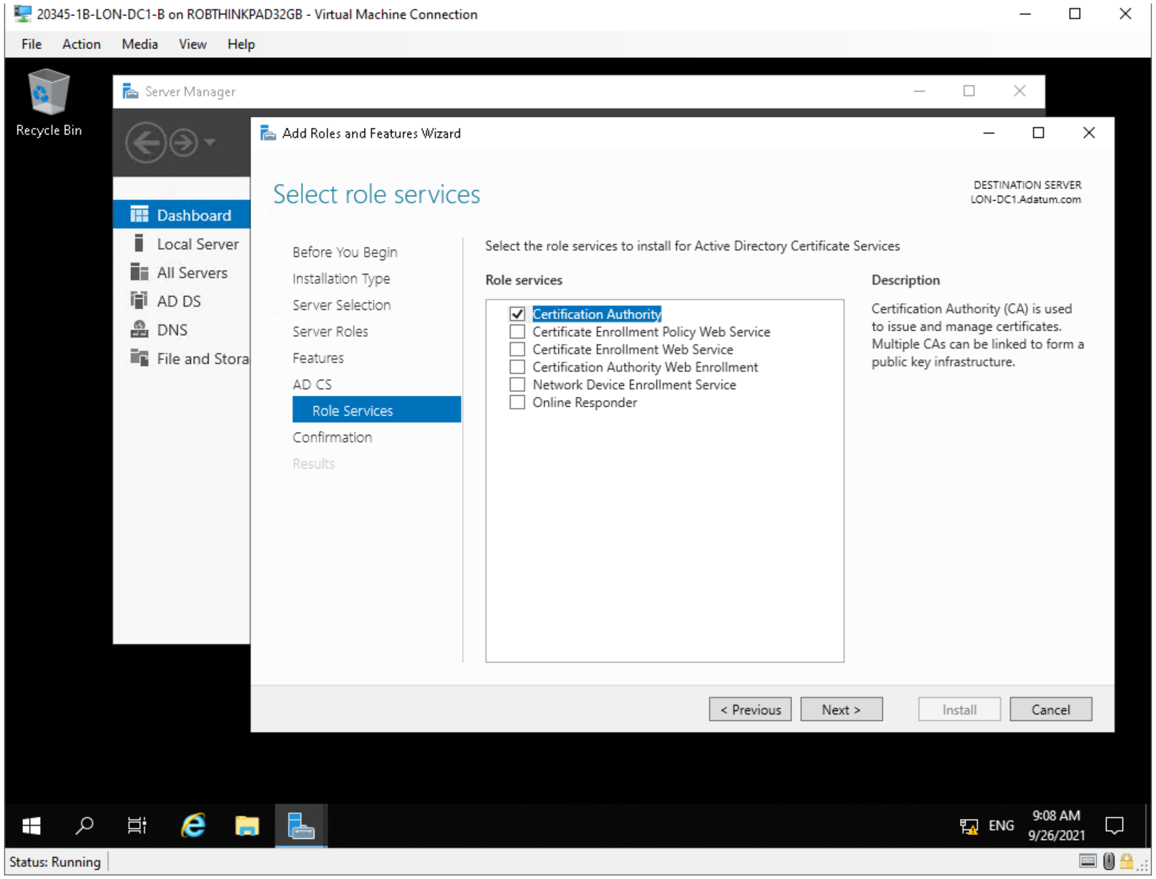
☑ Include management tools (if applicable)

Add Features | Cancel

Source: Microsoft

Figure 12-4 Selecting Active Directory Certificate Services

Understanding PKI (8 of 10)



Select role services

DESTINATION SERVER
LON-DC1.Adatum.com

Select the role services to install for Active Directory Certificate Services

Role services

- ☑ Certification Authority
- □ Certificate Enrollment Policy Web Service
- □ Certificate Enrollment Web Service
- □ Certification Authority Web Enrollment
- □ Network Device Enrollment Service
- □ Online Responder

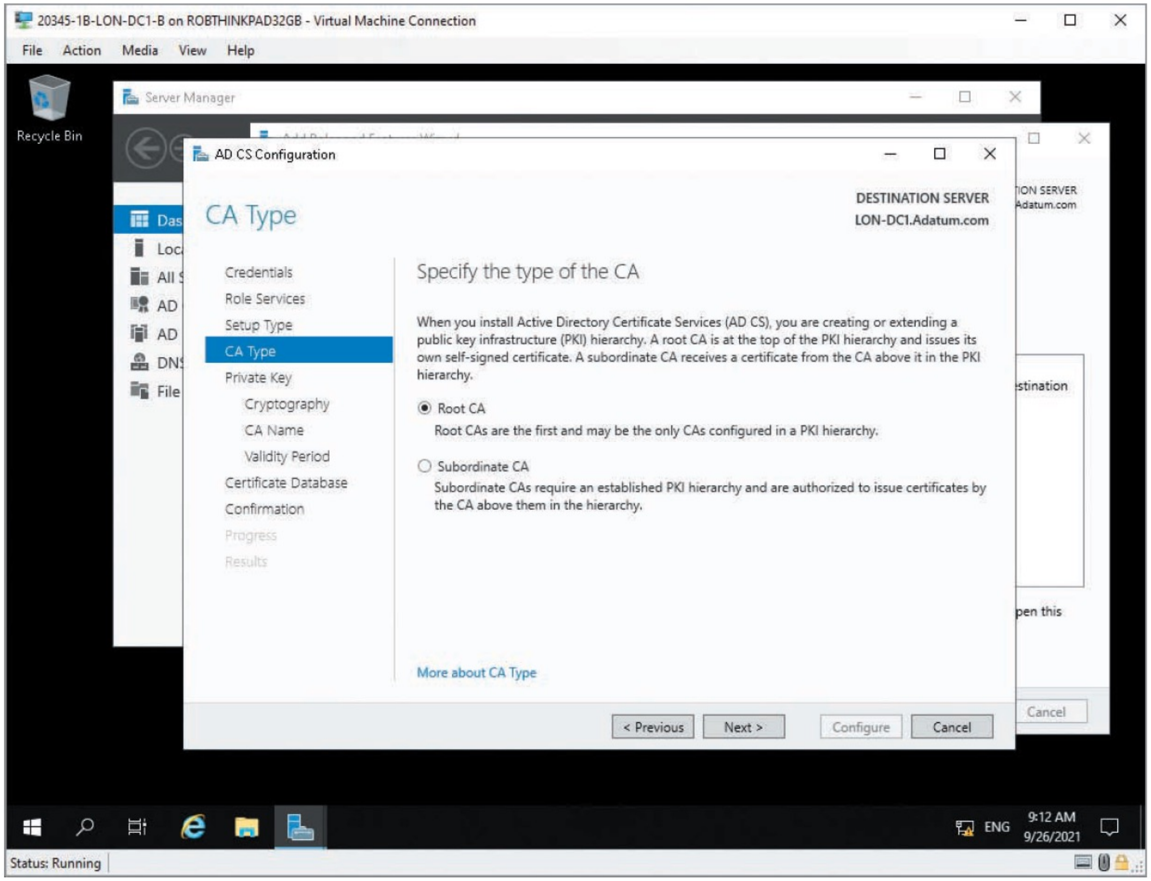
Description

Certification Authority (CA) is used to issue and manage certificates. Multiple CAs can be linked to form a public key infrastructure.

Source: Microsoft

Figure 12-5 Selecting role services to install

Understanding PKI (9 of 10)



CA Type

DESTINATION SERVER
LON-DC1.Adatum.com

Specify the type of the CA

When you install Active Directory Certificate Services (AD CS), you are creating or extending a public key infrastructure (PKI) hierarchy. A root CA is at the top of the PKI hierarchy and issues its own self-signed certificate. A subordinate CA receives a certificate from the CA above it in the PKI hierarchy.

● Root CA

Root CAs are the first and may be the only CAs configured in a PKI hierarchy.

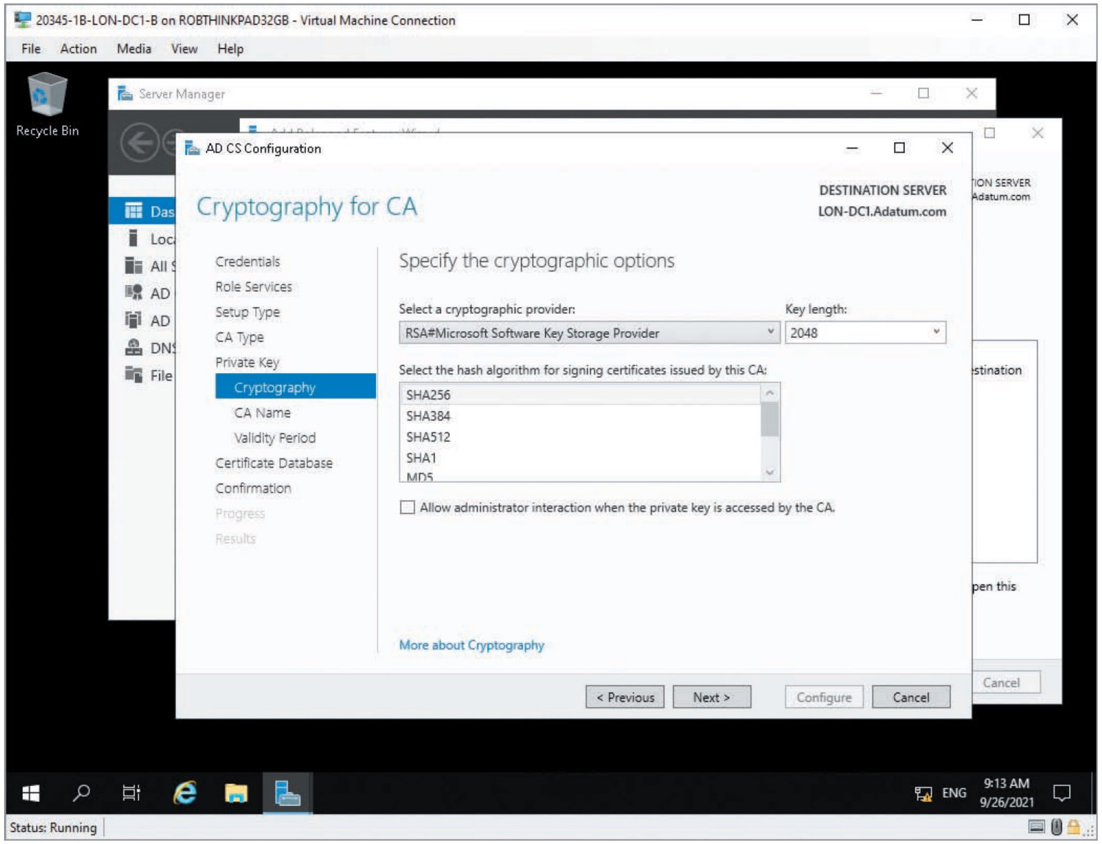
○ Subordinate CA

Subordinate CAs require an established PKI hierarchy and are authorized to issue certificates by the CA above them in the hierarchy.

Source: Cengage

Figure 12-6 Specifying a CA type

Understanding PKI (10 of 10)



Cryptography for CA

DESTINATION SERVER
LON-DC1.Adatum.com

Specify the cryptographic options

Select a cryptographic provider: RSA#Microsoft Software Key Storage Provider

Key length: 2048

Select the hash algorithm for signing certificates issued by this CA:

- SHA256
- SHA384
- SHA512
- SHA1
- MD5

□ Allow administrator interaction when the private key is accessed by the CA.

Source: Microsoft Windows

Figure 12-7 Configuring cryptography settings for a CA

Polling Activity 12-1

What is the standard for PKI certificates?

a. X.500
b. X.400
c. X.509
d. MySQL.409

Polling Activity 12-1: Answer

What is the standard for PKI certificates?

Answer: c. X.509

Each PKI certificate contains a unique serial number and must follow the X.509 standard that describes creating a certificate.

Understanding Cryptography Attacks

- Passive attacks
  - Using tools to eavesdrop
  - The attacker is only collecting data sent to and from a cryptosystem
- Active attacks
  - Attempt to determine the secret key used to encrypt plaintext
    - By actively sending input to a cryptosystem
- Culprit and general public usually know the algorithm
  - It's usually because companies developing encryption algorithms realize that the public might discover vulnerabilities that the company's programmers missed
    - Software engineers who develop open-source code products follow this philosophy

Birthday Attack

- Old adage that if 23 people are in a room, 50% probability that two will share the same birthday
- **Birthday attacks**
  - Used to find the same hash value for two different inputs
  - Used to reveal any mathematical weaknesses in hashing algorithms
- SHA-1
  - Uses a 160-bit digest
  - Finding a collision for a different message (the same birthday for a different person, in this analogy) would require 2^{60} computations

Mathematical Attack

- Properties of the algorithm are attacked by using mathematical computations
- Categories:
  - Ciphertext-only attack
  - Known plaintext attack
  - Chosen-plaintext attack
  - Chosen-ciphertext attack
  - Side-channel attack

Brute-Force Attack

- Attacker tries all possible keys in a keyspace
  - Example: Attacker uses a password-cracking program
    - To attempt every possible combination of characters in an effort to break the password hash
  - Can be launched on any kind of message digest

Man-in-the-Middle Attack

- Attackers place themselves between the victim computer and another host computer
  - They then intercept messages sent from the victim to the host
  - They pretend to be the host computer

SSL/TLS Downgrade Attack

- An attacker intercepts the initial communications between a web server and a web browser
  - Can force a vulnerable server to insecurely renegotiate the encryption being used down to a weaker cipher
  - This works because a web server and a web browser must negotiate which cipher will be used to communicate before they begin

Dictionary Attack

- Attacker runs a password-cracking program that uses a dictionary of known words as an input file
  - Most of these input files are available on the Internet and can be downloaded free
  - Unauthorized password cracking is illegal in most parts of the world, including the United States

Replay Attack

- The attacker captures data and attempts to resubmit the captured data
  - So that the device thinks a legitimate connection is in effect
  - If the captured data is logon information, attacker could gain access and be authenticated
- Many systems have countermeasures to prevent these attacks

Polling Activity 12-2

Intercepting messages destined for another computer and sending back messages while pretending to be the other computer is an example of what type of attack?

a. Man-in-the-middle

b. Smurf

c. Buffer overflow

d. Mathematical

Polling Activity 12-2: Answer

Intercepting messages destined for another computer and sending back messages while pretending to be the other computer is an example of what type of attack?

Answer: a. Man-in-the-middle

In a man-in-the-middle attack, attackers place themselves between the victim computer and another host computer. They, then intercept messages sent from the victim to the host and pretend to be the host computer.

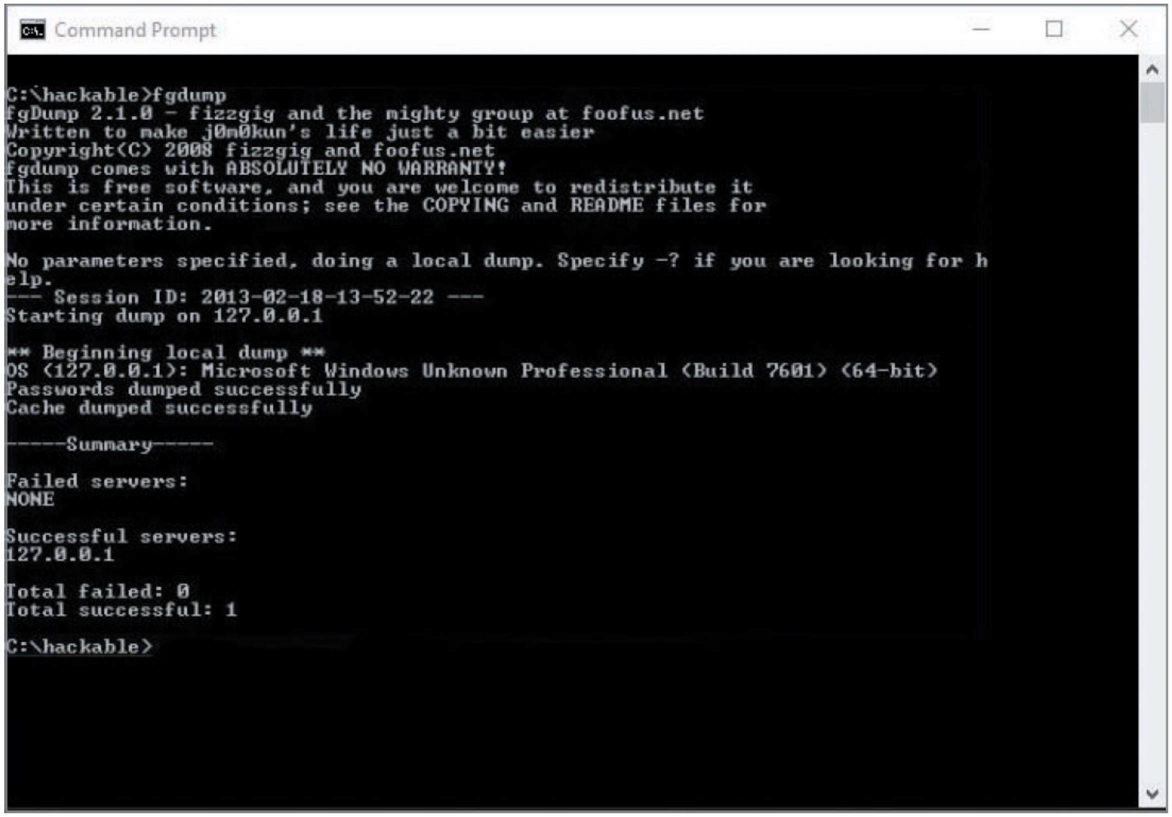
Understanding Password Cracking (1 of 4)

- Password cracking is illegal in the United States
  - It is legal to crack your own password if you forget it
- If a password uses common dictionary words
  - Most password-cracking programs can use a dictionary file to speed up the process
  - Brute force is a common method; one way to speed up a brute-force cracking effort is by using a **rainbow table**
- Salt
  - In cryptographic terms, it is the use of random data alongside plaintext as an input to a hashing function so that the output is unique.
  - Makes pre-calculated rainbow tables worthless
- A graphics processing unit (GPU) greatly exceeds a CPU’s capability to process mathematical calculations

Understanding Password Cracking (2 of 4)

- Password cracking
  - You must first obtain the password file from the system that stores usernames and passwords
  - Stored in /etc/shadow file for *nix systems
- Security Accounts Manager (SAM) file
  - Windows password hashes are stored in this file
- Password-cracking programs used to perform cracking attacks
  - Hashcat
  - John the Ripper
  - 0phcrack
  - EXPECT
  - L0phtcrack
  - Pwdump8

Understanding Password Cracking (3 of 4)

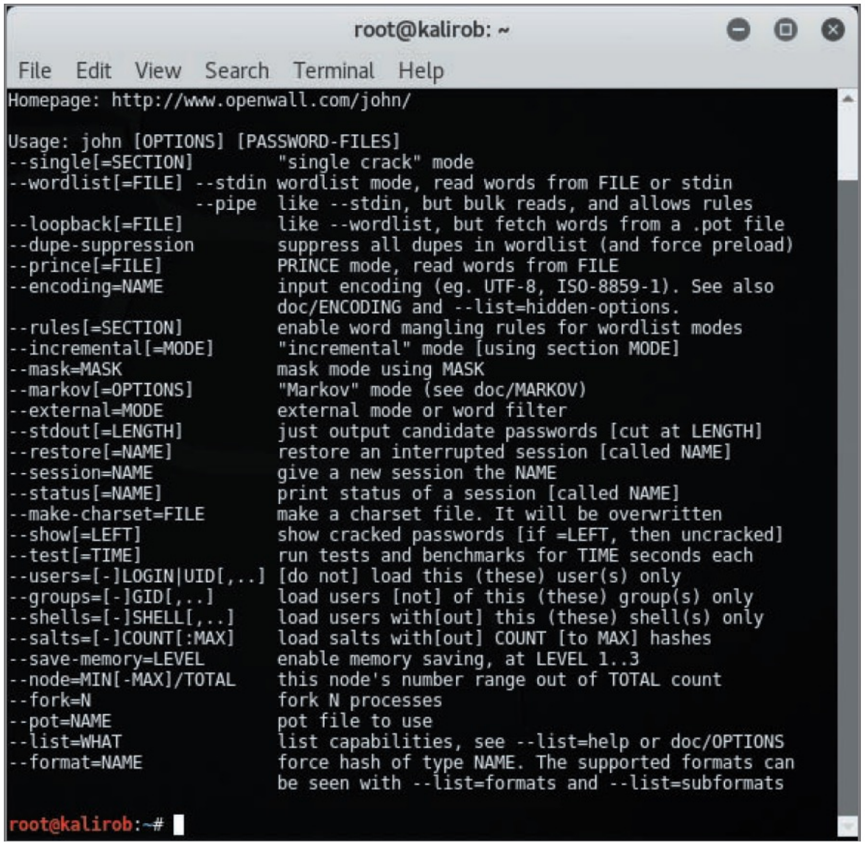


```
C:\hackable>fgdump
fgDump 2.1.0 - fizzgig and the mighty group at foofus.net
Written to make j0m0kun's life just a bit easier
Copyright(C) 2008 fizzgig and foofus.net
fgdump comes with ABSOLUTELY NO WARRANTY!
This is free software, and you are welcome to redistribute it
under certain conditions; see the COPYING and README files for
more information.

No parameters specified, doing a local dump. Specify -? if you are looking for h
elp.
--- Session ID: 2013-02-18-13-52-22 ---
Starting dump on 127.0.0.1

** Beginning local dump **
OS (127.0.0.1): Microsoft Windows Unknown Professional (Build 7601) (64-bit)
Passwords dumped successfully
Cache dumped successfully

-----Summary-----

Failed servers:
NONE

Successful servers:
127.0.0.1

Total failed: 0
Total successful: 1

C:\hackable>
```

Source: Microsoft Windows

Figure 12-8 Running fgdump

Understanding Password Cracking (4 of 4)



```
Homepage: http://www.openwall.com/john/

Usage: john [OPTIONS] [PASSWORD-FILES]
--single[=SECTION]         "single crack" mode
--wordlist[=FILE] --stdin wordlist mode, read words from FILE or stdin
                  --pipe  like --stdin, but bulk reads, and allows rules
--loopback[=FILE]          like --wordlist, but fetch words from a .pot file
--dupe-suppression         suppress all dupes in wordlist (and force preload)
--prince[=FILE]            PRINCE mode, read words from FILE
--encoding=NAME            input encoding (eg. UTF-8, ISO-8859-1). See also
                           doc/ENCODING and --list=hidden-options.
--rules[=SECTION]          enable word mangling rules for wordlist modes
--incremental[=MODE]       "incremental" mode [using section MODE]
--mask=MASK                mask mode using MASK
--markov[=OPTIONS]         "Markov" mode (see doc/MARKOV)
--external=MODE            external mode or word filter
--stdout[=LENGTH]          just output candidate passwords [cut at LENGTH]
--restore[=NAME]           restore an interrupted session [called NAME]
--session=NAME             give a new session the NAME
--status[=NAME]            print status of a session [called NAME]
--make-charset=FILE        make a charset file. It will be overwritten
--show[=LEFT]              show cracked passwords [if =LEFT, then uncracked]
--test[=TIME]              run tests and benchmarks for TIME seconds each
--users=[-]LOGIN|UID[,..] [do not] load this (these) user(s) only
--groups=[-]GID[,..]       load users [not] of this (these) group(s) only
--shells=[-]SHELL[,..]     load users with[out] this (these) shell(s) only
--salts=[-]COUNT[:MAX]     load salts with[out] COUNT [to MAX] hashes
--save-memory=LEVEL        enable memory saving, at LEVEL 1..3
--node=MIN[-MAX]/TOTAL     this node's number range out of TOTAL count
--fork=N                   fork N processes
--pot=NAME                 pot file to use
--list=WHAT                list capabilities, see --list=help or doc/OPTIONS
--format=NAME              force hash of type NAME. The supported formats can
                           be seen with --list=formats and --list=subformats

root@kalirob:~#
```

Source: GNU General Public License (GNU GPL)

Figure 12-9 Using John the Ripper parameters

Self-Assessment

Explain the differences between symmetric algorithms and asymmetric algorithms.

Summary

- Now that the lesson has ended, you should be able to:
  - Summarize the history and principles of cryptography
  - Describe symmetric and asymmetric encryption algorithms
  - Explain public key infrastructure (PKI)
  - Describe possible attacks on cryptosystems
  - Compare hashing algorithms and how they ensure data integrity