Module 20

Cryptography

EC-Council
Official Curricula

EC-Council C|EH v13 Certified Ethical Hacker

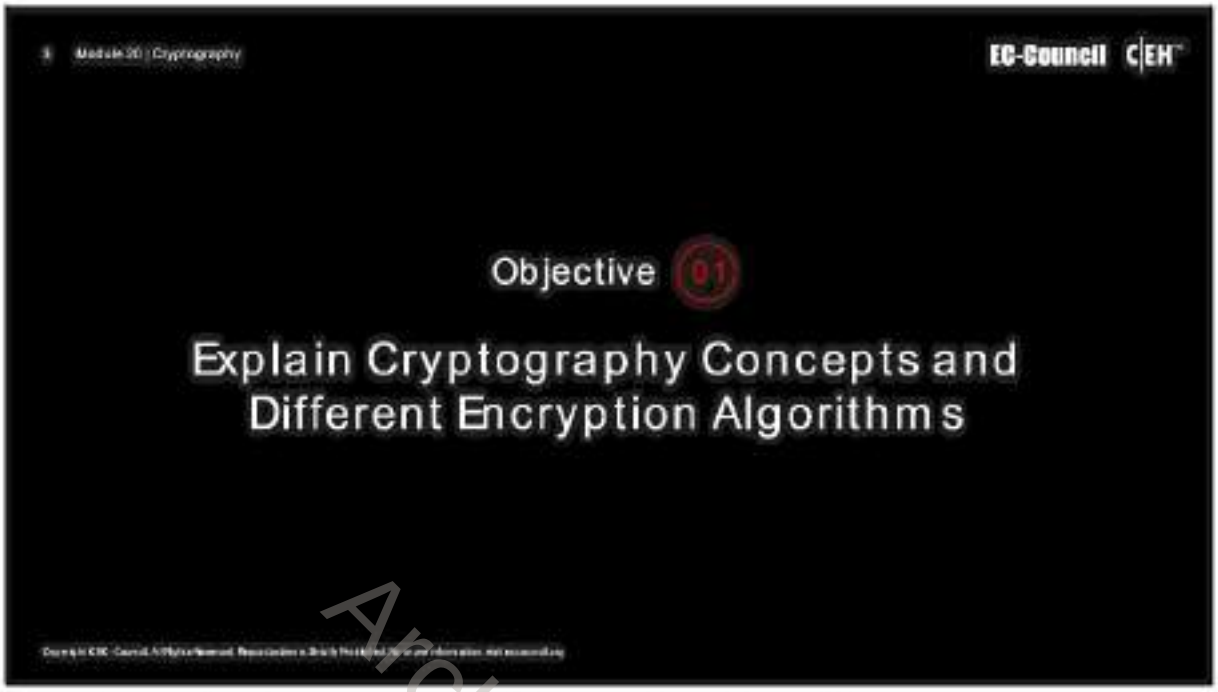
Learning Objectives

With the increasing adoption of the Internet (World Wide Web) for business and personal communication, securing sensitive information such as credit card details, PINs, bank account numbers, and private messages is becoming increasingly important, albeit more difficult to achieve. Today's information-based organizations extensively use the Internet for e-commerce, market research, customer support, and a variety of other activities. Data security is critical to online business and communication privacy.

Cryptography and cryptographic ("crypto") systems help in securing data against interception and compromise during online transmissions. This module provides a comprehensive understanding of different cryptosystems and algorithms, one-way hash functions, public-key infrastructures (PKIs), and the different ways in which cryptography can ensure the privacy and security of online communication. It also covers various tools used to encrypt sensitive data.

At the end of this module, you will be able to

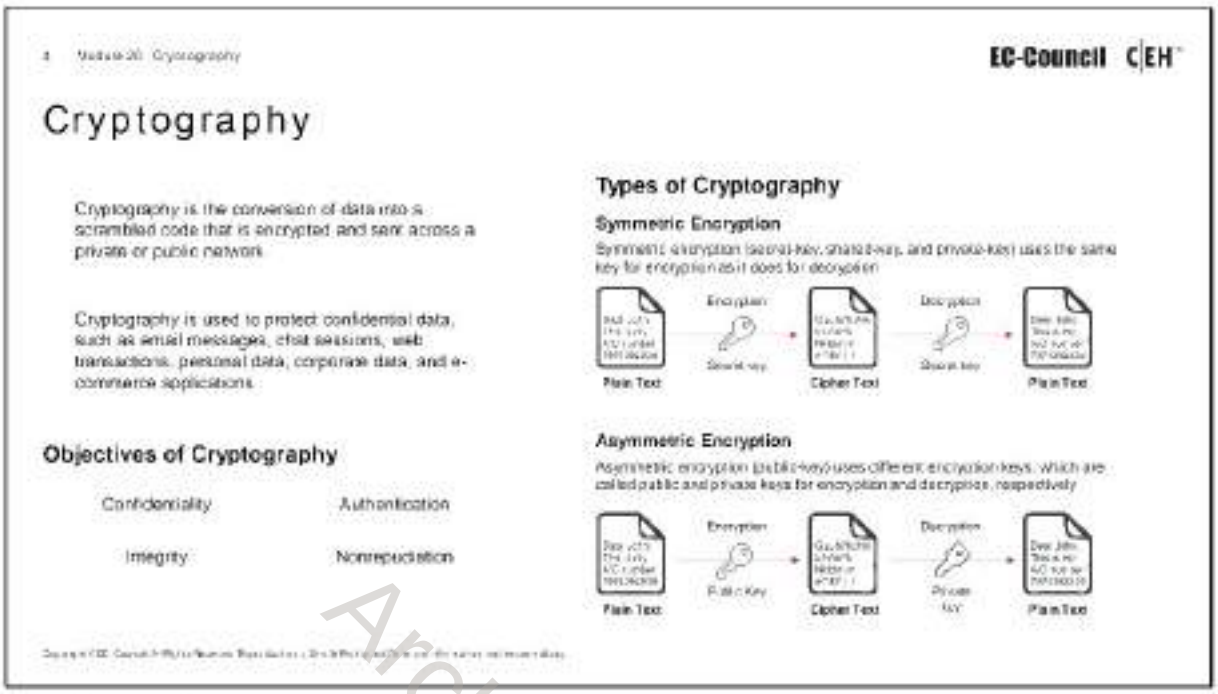
- Describe cryptography concepts
- Understand different encryption algorithms
- Use different cryptography tools
- Apply various applications of cryptography
- Describe various cryptography attacks
- Use different cryptanalysis tools



Cryptography Concepts and Encryption Algorithms

Cryptography enables one to secure transactions, communications, and other processes performed in the electronic world. Encryption is the process of converting readable plaintext into an unreadable ciphertext using a set of complex algorithms that transform the data into blocks or streams of random alphanumeric characters.

This section deals with cryptography and its associated concepts, which will enable you to understand the advanced topics covered later in this module. It also deals with ciphers and various encryption algorithms such as DES, AES, RC4, RC5, RC6, DSA, RSA, MD5, SHA, etc.



Cryptography

"Cryptography" comes from the Greek words ***kryptos***, meaning "concealed, hidden, veiled, secret, or mysterious," and ***graphia***, meaning "writing"; thus, cryptography is "the art of secret writing."

Cryptography is the practice of concealing information by converting plaintext (readable format) into ciphertext (unreadable format) using a key or encryption scheme. It is the process of converting data into a scrambled code that is encrypted and sent across a private or public network. Cryptography protects confidential data such as email messages, chat sessions, web transactions, personal data, corporate data, e-commerce applications, and many other types of communication. Encrypted messages can, at times, be decrypted by cryptanalysis (code breaking), even though modern encryption techniques are virtually unbreakable.

Objectives of Cryptography

- **Confidentiality**: Assurance that the information is accessible only to those authorized to access it.
- **Integrity**: Trustworthiness of data or resources in terms of preventing improper and unauthorized changes.
- **Authentication**: Assurance that the communication, document, or data is genuine.
- **Nonrepudiation**: Guarantee that the sender of a message cannot later deny having sent the message and that the recipient cannot deny having received the message.

Cryptography Process

Plaintext (readable format) is encrypted by means of encryption algorithms such as RSA, DES, and AES, resulting in a ciphertext (unreadable format) that, on reaching the destination, is decrypted into readable plaintext.

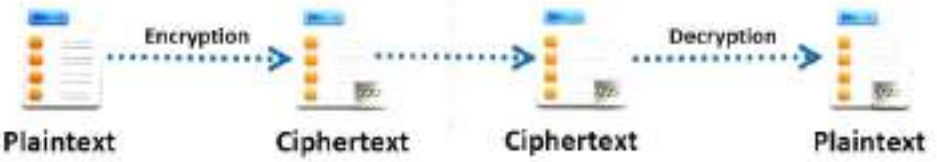


Figure 20.1: Example of Cryptography

Types of Cryptography

Cryptography is categorized into two types according to the number of keys employed for encryption and decryption:

- **Symmetric Encryption**

  Symmetric encryption requires that both the sender and the receiver of the message possess the same encryption key. The sender uses a key to encrypt the plaintext and sends the resultant ciphertext to the recipient, who uses the same key (used for encryption) to decrypt the ciphertext into plaintext. Symmetric encryption is also known as secret-key cryptography, as it uses only one secret key to encrypt and decrypt the data. This type of cryptography works well when you are communicating with only a few people.

  Because the sender and receiver must share the key before sending any messages, this technique is of limited use for the Internet, where individuals who have not had prior contact frequently require a secure means of communication. The solution to this problem is asymmetric encryption (public-key cryptography).

Figure 20.2: Symmetric Encryption

- **Asymmetric Encryption**

  The concept of asymmetric encryption (also known as public-key cryptography) was introduced to solve key-management problems. Asymmetric encryption involves both a public key and a private key. The public key is publicly available, whereas the sender keeps the private key secret.

  An asymmetric-key system is an encryption method that uses a key pair comprising a public key available to anyone and a private key held only by the key owner, which helps to provide confidentiality, integrity, authentication, and nonrepudiation in data management.

Asymmetric encryption uses the following sequence to send a message:

1. An individual finds the public key of the person he or she wants to contact in a directory.
2. This public key is used to encrypt a message that is then sent to the intended recipient.
3. The receiver uses the private key to decrypt the message and reads it.

No one but the holder of the private key can decrypt a message encrypted with the corresponding public key. This increases the security of the information because all communications involve only public keys; the message sender never transmits or shares the private keys. The sender must link public keys with usernames in a secure manner to ensure that individuals claiming to be the intended recipient do not intercept the information. To meet the need for authentication, one can use digital signatures.

Figure 20.3: Asymmetric Encryption

Strengths and Weaknesses of Crypto Methods

	Symmetric Encryption	Asymmetric Encryption
Strengths	Faster and easier to implement, as the same key is used to encrypt and decrypt data Requires less processing power Can be implemented in application-specific integrated chip (ASIC).	Convenient to use, as the distribution of keys to encrypt messages is not required
	Prevents widespread message security compromise as different secret keys are used to communicate with different parties	Enhanced security, as one need not share or transmit private keys to anyone
	The key is not bound to the data being transferred on the link; therefore, even if the data are intercepted, it is not possible to decrypt it	Provides digital signatures that cannot be repudiated
Weaknesses	**Symmetric Encryption**	**Asymmetric Encryption**
	Lack of secure channel to exchange the secret key	Slow in processing and requires high processing power
	Difficult to manage and secure too many shared keys that are generated to communicate with different parties	Widespread message security compromise is possible (i.e., an attacker can read complete messages if the private key is compromised)

	Provides no assurance about the origin and authenticity of a message, as the same key is used by both the sender and the receiver	Messages received cannot be decrypted if the private key is lost
	Vulnerable to dictionary attacks and brute-force attacks	Vulnerable to man-in-the-middle and brute-force attacks

Table 20.1: Strengths and weaknesses of crypto methods

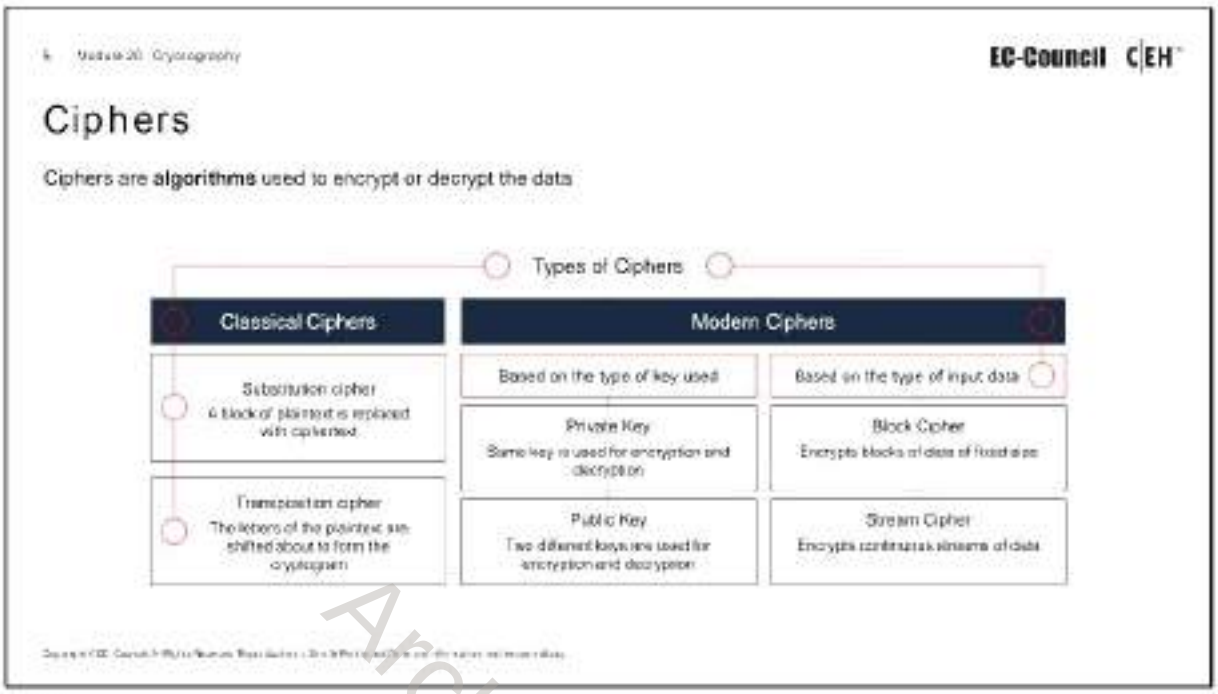
Government Access to Keys (GAK)

Government Access to Keys (GAK) refers to the statutory obligation of individuals and organizations to disclose their cryptographic keys to government agencies. It means that software companies will give copies of all keys (or at least enough of the key such that the remainder can be cracked) to the government. Law enforcement agencies around the world acquire and use these cryptographic keys to monitor suspicious communication and collect evidence of cybercrimes in the interests of national security. The government promises that it will hold on to the keys in a secure manner and only use them when a court issues a warrant to do so. To the government, this issue is similar to the ability to wiretap phones.

Government agencies often use key escrow for uninterrupted access to keys. Key escrow is a key exchange arrangement in which essential cryptographic keys are stored with a third party in escrow. The third party can use or allow others to use the encryption keys under certain predefined circumstances. The third party, with regard to GAK, is generally a government agency that may use the encryption keys to decipher digital evidence under authorization or a warrant from a court of law. However, there is growing concern about the privacy and security of cryptographic keys and information. Government agencies are responsible for protecting these keys. Such agencies generally use a single key to protect other keys, which is not a good idea, as revealing a single key could expose the other keys.

These agencies are not aware of how confidential the information protected by the keys is, which makes it difficult to judge how much protection is required. In cases where seized keys also protect other information that these agencies have no right to access, the consequences of key revelation cannot be determined, because government agencies are not aware of the information that the keys protect. In such cases, the key owner is liable for the consequences of key revelation. Before owners hand over their keys to government agencies, they need to be assured that the government agencies will protect these keys according to a sufficiently strong standard to protect their interests.

Figure 20.4: Illustration of GAK



Ciphers

In cryptography, a cipher is an algorithm (a series of well-defined steps) for performing encryption and decryption. Encipherment is the process of converting plaintext into a cipher or code; the reverse process is called decipherment. A message encrypted using a cipher is rendered unreadable unless its recipient knows the secret key required to decrypt it. Communication technologies (e.g., Internet, cell phones) rely on ciphers to maintain both security and privacy. Cipher algorithms may be open-source (the algorithmic process is in the public domain while the key is selected by a user and is private) or closed-source (the process is developed for use in specific domains, such as the military, and the algorithm itself is not in the public domain). Furthermore, ciphers may be free for public use or licensed.

Types of ciphers

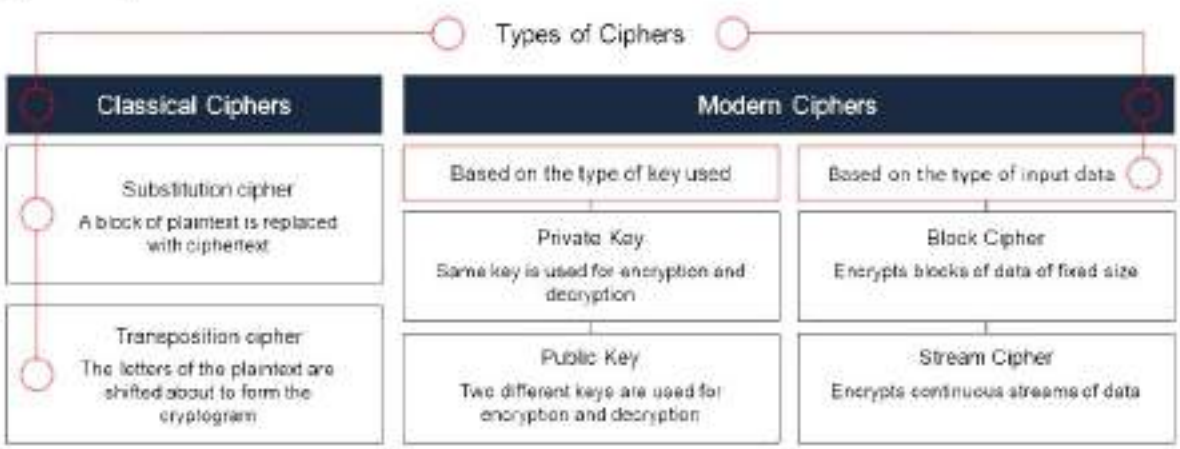


Figure 20.5: Classification of ciphers

Ciphers are of two main types: classical and modern.

- **Classical Ciphers**

  Classical ciphers are the most basic type of ciphers, which operate on letters of the alphabet (A–Z). These ciphers are generally implemented either by hand or with simple mechanical devices. Because these ciphers are easily deciphered, they are generally unreliable.

  Types of classical ciphers

  - **Substitution cipher**: The user replaces units of plaintext with ciphertext according to a regular system. The units may be single letters, pairs of letters, or combinations of them, and so on. The recipient performs inverse substitution to decipher the text. Examples include the Beale cipher, autokey cipher, Gronsfeld cipher, and Hill cipher.

    For example, "**HELLO WORLD**" can be encrypted as "**PSTER HGFST**" (i.e., H=P, E=S, etc.).

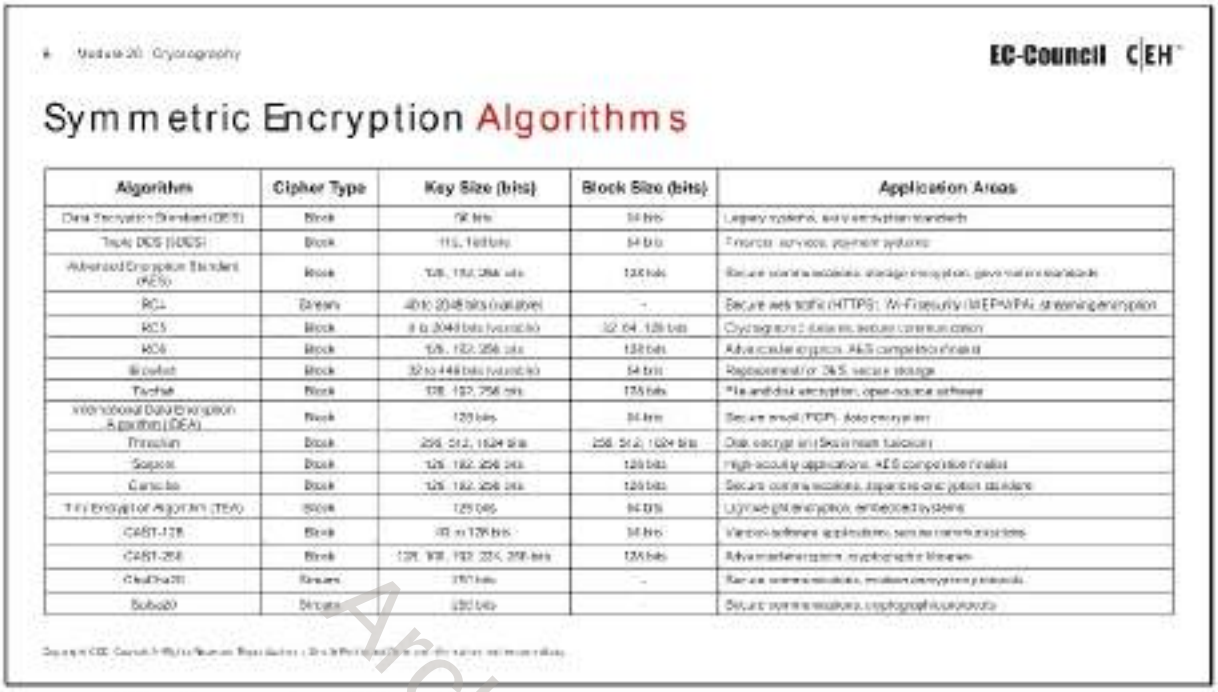
  - **Transposition cipher**: Here, letters in the plaintext are rearranged according to a regular system to produce the ciphertext. For example, "**CRYPTOGRAPHY**" when encrypted becomes "**AOYCRGPTYRHP**." Examples include the rail fence cipher, route cipher, and Myszkowski transposition.

- **Modern Ciphers**

  Modern ciphers are designed to withstand a wide range of attacks. They provide message secrecy, integrity, and authentication of the sender. A user can calculate a modern cipher using a one-way mathematical function that is capable of factoring large prime numbers.

  Types of Modern ciphers

  - **Based on the type of key used**
    - **Symmetric-key algorithms (Private-key cryptography)**: Use the same key for encryption and decryption.
    - **Asymmetric-key algorithms (Public-key cryptography)**: Use two different keys for encryption and decryption.
  - **Based on the type of input data**
    - **Block cipher**: Deterministic algorithms operating on a block (a group of bits) of fixed size with an unvarying transformation specified by a symmetric key. Most modern ciphers are block ciphers. They are widely used to encrypt bulk data. Examples include DES, AES, IDEA, etc. When the block size is less than that used by the cipher, padding is employed to achieve a fixed block size.
    - **Stream cipher**: Symmetric-key ciphers are plaintext digits combined with a key stream (pseudorandom cipher digit stream). Here, the user applies the key to each bit, one at a time. Examples include RC4, SEAL, etc.



Algorithm	Cipher Type	Key Size (bits)	Block Size (bits)	Application Areas
Data Encryption Standard (DES)	Block	56 bits	64 bits	Legacy systems, early encryption standards
Triple DES (3DES)	Block	112, 168 bits	64 bits	Financial services, payment systems
Advanced Encryption Standard (AES)	Block	128, 192, 256 bits	128 bits	Secure communications, storage encryption, government standards
RC4	Stream	40 to 2048 bits (variable)	-	Secure web traffic (HTTPS), Wi-Fi security (WEP/WPA), streaming encryption
RC5	Block	0 to 2040 bits (variable)	32, 64, 128 bits	Cryptographic libraries, secure communication
RC6	Block	128, 192, 256 bits	128 bits	Advanced encryption, AES competition finalist
Blowfish	Block	32 to 448 bits (variable)	64 bits	Replacement for DES, secure storage
Twofish	Block	128, 192, 256 bits	128 bits	File and disk encryption, open-source software
International Data Encryption Algorithm (IDEA)	Block	128 bits	64 bits	Secure email (PGP), data encryption
Threefish	Block	256, 512, 1024 bits	256, 512, 1024 bits	Disk encryption (Skein hash function)
Serpent	Block	128, 192, 256 bits	128 bits	High-security applications, AES competition finalist
Camellia	Block	128, 192, 256 bits	128 bits	Secure communications, Japanese encryption standard
TEA (Tiny Encryption Algorithm)	Block	128 bits	64 bits	Lightweight encryption, embedded systems
CAST-128	Block	40 to 128 bits	64 bits	Various software applications, secure communications
CAST-256	Block	128, 160, 192, 224, 256 bits	128 bits	Advanced encryption, cryptographic libraries
ChaCha20	Stream	256 bits	-	Secure communications, modern encryption protocols
Salsa20	Stream	256 bits		Secure communications, cryptographic protocols

Symmetric Encryption Algorithms

The table below shows specified symmetric encryption algorithms, including information such as cipher type, key size, block size, and application areas.

Algorithm	Cipher Type	Key Size (bits)	Block Size (bits)	Application Areas
Data Encryption Standard (DES)	Block	56 bits	64 bits	Legacy systems, early encryption standards
Triple DES (3DES)	Block	112, 168 bits	64 bits	Financial services, payment systems
Advanced Encryption Standard (AES)	Block	128, 192, 256 bits	128 bits	Secure communications, storage encryption, government standards
RC4	Stream	40 to 2048 bits (variable)	-	Secure web traffic (HTTPS), Wi-Fi security (WEP/WPA), streaming encryption
RC5	Block	0 to 2040 bits (variable)	32, 64, 128 bits	Cryptographic libraries, secure communication
RC6	Block	128, 192, 256 bits	128 bits	Advanced encryption, AES competition finalist
Blowfish	Block	32 to 448 bits (variable)	64 bits	Replacement for DES, secure storage

Twofish	Block	128, 192, 256 bits	128 bits	File and disk encryption, open-source software
International Data Encryption Algorithm (IDEA)	Block	128 bits	64 bits	Secure email (PGP), data encryption
Threefish	Block	256, 512, 1024 bits	256, 512, 1024 bits	Disk encryption (Skein hash function)
Serpent	Block	128, 192, 256 bits	128 bits	High-security applications, AES competition finalist
Camellia	Block	128, 192, 256 bits	128 bits	Secure communications, Japanese encryption standard
Tiny Encryption Algorithm (TEA)	Block	128 bits	64 bits	Lightweight encryption, embedded systems
CAST-128	Block	40 to 128 bits	64 bits	Various software applications, secure communications
CAST-256	Block	128, 160, 192, 224, 256 bits	128 bits	Advanced encryption, cryptographic libraries
ChaCha20	Stream	256 bits	-	Secure communications, modern encryption protocols
Salsa20	Stream	256 bits	-	Secure communications, cryptographic protocols

Table 20.2: Symmetric encryption algorithms

Data Encryption Standard (DES)

DES is a standard for data encryption that uses a secret key for both encryption and decryption (symmetric cryptosystem). DES uses a 64-bit secret key, of which 56 bits are generated randomly and the other 8 bits are used for error detection. It uses a data encryption algorithm (DEA), a secret key block cipher employing a 56-bit key operating on 64-bit blocks. DES is the archetypal block cipher—an algorithm that takes a fixed-length string of plaintext bits and transforms it into a ciphertext bit string of the same length. The design of DES allows users to implement it in hardware and use it for single-user encryption, such as to store files on a hard disk in encrypted form.

DES provides 72 quadrillion or more possible encryption keys and chooses a random key for the encryption of each message. Because of the inherent weakness of DES vis-à-vis today's technologies, some organizations use triple DES (3DES), in which they repeat the process three times for added strength until they can afford to update their equipment to AES capabilities.

Triple Data Encryption Standard (3DES)

Eventually, it became obvious that DES would no longer be secure. The U.S. Federal Government began a contest seeking a replacement cryptography algorithm. However, in the meantime, 3DES was created as an interim solution. Essentially, it performs DES three times with three different

keys. 3DES uses a "key bundle" that comprises three DES keys, K1, K2, and K3. Each key is a standard 56-bit DES key. It then performs the following process:

DES *encrypt* with K1, DES *decrypt* with K2, DES *encrypt* with K3

There are three options for the keys. In the first option, all three keys are independent and different. In the second option, K1 and K3 are identical. In the third option, all three keys are the same; therefore, you are literally applying the same DES algorithm three times with the same key. The first option is the most secure, while the third is the least secure.

Advanced Encryption Standard (AES)

The Advanced Encryption Standard (AES) is a National Institute of Standards and Technology (NIST) specification for the encryption of electronic data. It also helps to encrypt digital information such as telecommunications, financial, and government data. US government agencies have been using it to secure sensitive but unclassified material.

AES consists of a symmetric-key algorithm: both encryption and decryption are performed using the same key. It is an iterated block cipher that works by repeating the defined steps multiple times. It has a 128-bit block size, with key sizes of 128, 192, and 256 bits for AES-128, AES-192, and AES-256, respectively. The design of AES makes its use efficient in both software and hardware. It works simultaneously at multiple network layers.

AES Pseudocode

Initially, the system copies the cipher input into the internal state and then adds an initial round key. The system transforms the state by iterating a round function in a number of cycles. The number of cycles may vary with the block size and key length. After completing rounding, the system copies the final state into the cipher output.

```
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
```

RC4, RC5, and RC6 Algorithms

Symmetric encryption algorithms developed by RSA Security are discussed below.

- **RC4**

  RC4 is a variable key-size symmetric-key stream cipher with byte-oriented operations, and it is based on the use of a random permutation. According to some analyses, the period of the cipher is likely to be greater than 10,100. Each output byte uses 8 to 16 system operations; thus, the cipher can run fast when used in software. RC4 enables safe communications such as for traffic encryption (which secures websites) and for websites that use the SSL protocol.

- **RC5**

  RC5 is a fast symmetric-key block cipher designed by Ronald Rivest for RSA Data Security (now RSA Security). The algorithm is a parameterized algorithm with a variable block size, a variable key size, and a variable number of rounds. The block sizes can be 32, 64, or 128 bits. The range of the rounds can vary from 0 to 255, and the size of the key can vary from 0 to 2,040 bits. This built-in variability can offer flexibility at all levels of security. The routines used in RC5 are key expansion, encryption, and decryption.

  In the key expansion routine, the secret key that a user provides is expanded to fill the key table (the size of which depends on the number of rounds). RC5 uses a key table for both encryption and decryption. The encryption routine has three fundamental operations: integer addition, bitwise XOR, and variable rotation. The intensive use of data-dependent rotation and the combination of different operations make RC5 a secure encryption algorithm.

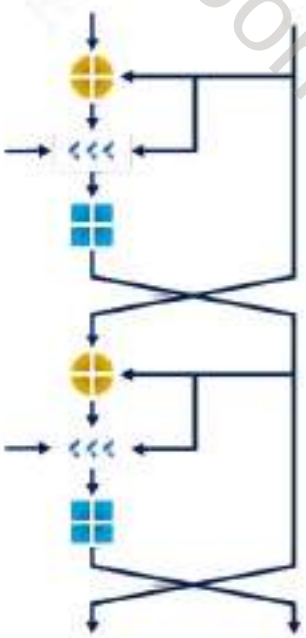


Figure 20.6: Block diagram of the RC5 algorithm

- **RC6**

  RC6 is a symmetric-key block cipher derived from RC5. It is a parameterized algorithm with a variable block size, key size, and number of rounds. Two features that differentiate RC6 from RC5 are integer multiplication (which is used to increase the diffusion, achieved in fewer rounds with increased speed of the cipher) and the use of four 4-bit working registers rather than two 2-bit registers. RC6 uses four 4-bit registers instead of two 2-bit registers because the block size of the AES is 128 bits.

Blowfish

Blowfish is a type of symmetric block cipher algorithm designed to replace DES or IDEA algorithms. It uses the same secret key to encrypt and decrypt data. This algorithm splits the data into a block length of 64 bits and produces a key ranging from 32 bits to 448 bits. Due to its high speed and overall efficiency, blowfish is used in software ranging from password protection tools to e-commerce websites for securing payments.

It is a 16-round Feistel cipher working on 64-bit blocks. However, unlike DES, its key size ranges from 32 bits to 448 bits.

This algorithm has two parts. The first part handles the expansion of the key. The second part actually encrypts the data.

The key expansion is handled in several steps. The first step is to break the original key into a set of subkeys. Specifically, a key of no more than 448 bits is separated into 4,168 bytes. There is a P-array and four 32-bit S-boxes. The P-array contains 18 32-bit subkeys, while each S-box contains 256 entries.

Key expansion is performed as follows:

1. The first step is to initialize the **P-array** and **S-boxes**.
2. Then, XOR the P-array with the key bits. For example, **P1 XOR** (first 32 bits of the key), **P2 XOR** (next 32 bits of the key).
3. Use the above method to encrypt the **all-zero string**.
4. This new output is now P1 and P2.
5. Encrypt the new P1 and P2 with the **modified subkeys.**
6. This new output is now P3 and P4.
7. Repeat the process **521 times** to calculate new subkeys for the P-array and the four S-boxes.

The round function splits the 32-bit input into four 8-bit quarters and uses the quarters as input to the S-boxes. The outputs are added modulo **2^32** and **XORed** to produce the final **32-bit** output.

Twofish

The Twofish algorithm was one of the U.S. Government's five finalists to replace DES, but it was not chosen. It was designed by Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, and Niels Ferguson.

TwoFish is a 128-bit block cipher. It is one of the most conceptually simple algorithms that uses a single key for both encryption and decryption for any length up to 256 bits. It is a Feistel cipher. It not only works fast for CPU or hardware but is also flexible for network-based applications. Furthermore, it allows various levels of performance trade-off on parameters such as encryption speed, hardware gate count, memory usage, etc. This technique of enabling different implementations improves the relative performance of the algorithm. Any user can optimize the performance based on the key scheduling.

Threefish

Threefish was developed in 2008 and it is a part of the Skein algorithm. It was enrolled in NIST's SHA-3 (hash function) contest. It is a large tweakable symmetric-key block cipher in which the block and key sizes are equal, i.e., 256, 512, and 1024. Threefish involves only three operations, i.e., ARX (addition-rotation-XOR), which makes the coding simple, and all these operations work on 64-bit words. Threefish blocks 256, 512, and 1024 involve 72, 72, and 80 rounds of computations, respectively, to achieve the final security goal. This algorithm does not use S-boxes to prevent cache timing attacks.

Serpent

Like Blowfish, Serpent is a symmetric-key block cipher that was a finalist in the AES contest. This algorithm was designed by Ross Anderson, Eli Biham, and Lars Knudsen. It uses a 128-bit symmetric block cipher with key sizes of 128, 192, or 256 bits. It can be integrated into software or hardware programs without any restrictions.

Serpent involves 32 rounds of computational operations that include substitution and permutation operations on four 32-bit word blocks using 8-variable S-boxes with 4-bit entry and 4-bit exit. All S-boxes work parallelly 32 times. Although, Serpent is one of the most secure encryption mechanisms in AES contests, researchers have chosen Rijndael over Serpent due to its moderate encryption speed (owing to the number of rounds it uses) and complexity. Serpent minimizes the correlation between encoded images or plaintexts to a greater extent compared to Twofish and Rijndael. Therefore, Rijndael is the stand-out AES competitor and is now being used as AES.

TEA

The tiny encryption algorithm (TEA) was created by David Wheeler and Roger Needham, and it was publicly presented for the first time in 1994. It is a simple algorithm, easy to implement in code. It is a Feistel cipher that uses 64 rounds (note that this is a suggestion; it can be implemented with fewer or more rounds). The number of rounds should be even since they are implemented in pairs called cycles.

TEA uses a 128-bit key operating on a 64-bit block. It also uses a constant that is defined as 232/the golden ratio. This constant is referred to as delta, and in each round, a multiple of delta

is used. The 128-bit key is split into four different 32-bit subkeys labeled K[0], K[1], K[2], and K[3]. Instead of using the XOR operation, TEA uses addition and subtraction, but with mod 232. The block is divided into two halves, R and L. R is processed through the round function.

The round function takes the R half and performs a left shift of 4. Then, the result of this operation is added to K[0]. Next, the result of this operation is added to delta (recall that delta is the current multiple of 232/the golden ratio). The result of this operation is then shifted right by 5 and added to K[1]. This is the round function. As with all Feistel ciphers, the result of the round function is XORed with L, and L and R are then swapped for the next round.

CAST-128

CAST-128, also called CAST5, is a symmetric-key block cipher having a classical 12- or 16-round Feistel network with a block size of 64 bits. CAST-128 uses a key size varying from 40 bits to 128 bits in 8-bit increments. The CAST-128 components include large 8×32-bit S-boxes (S1, S2, S3, S4) based on bent functions, modular addition and subtraction, key-dependent rotation, and XOR operations. CAST-128 uses a masking key (K_{m1}) and a rotation key (K_{r1}) for performing its functions. The round function consists of three alternating types to perform addition, subtraction, or XOR operations in different stages. It used as a default cipher in GPG (GNU Privacy Guard) and PGP (Pretty Good Privacy).

CAST-256 is an extension of CAST-128 that uses the same design procedure. CAST-256 has a 128-bit block size and uses key sizes varying from 128 to 256 bits. Furthermore, it uses zero-correlation cryptanalysis, which can break 28 rounds with time = $2^{246.9}$ and data = $2^{98.8}$.

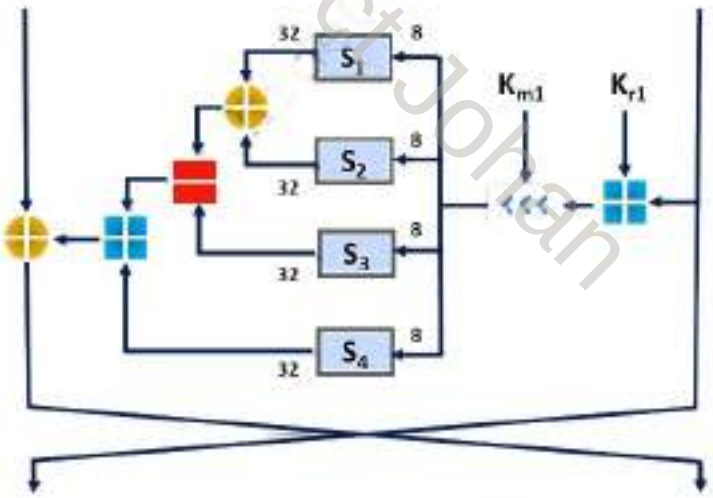


Figure 20.7: Block diagram of CAST-128

GOST Block Cipher

The GOST (Government Standard) block cipher, also called Magma, is a symmetric-key block cipher having a 32-round Feistel network working on 64-bit blocks with a 256-bit key length. It consists of an S-box that can be kept secret and it contains around 354 bits of secret information. GOST is a simple encryption algorithm, where the round function 32-bit subkey modulo 2^32 is added and put in the layer of S-boxes and the rotate left shift operation is used for shifting 11 bits, thereby providing the output of the round function.

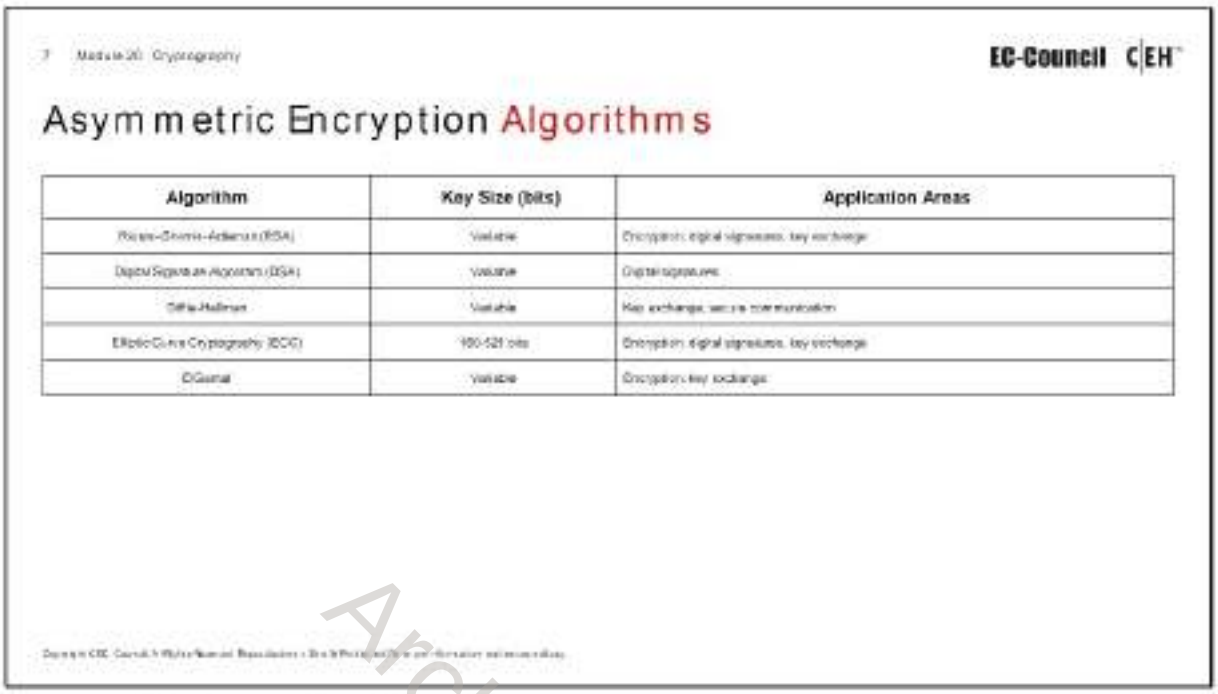
The key scheduling of the GOST block cipher is performed by breaking the 256-bit key into eight 32-bit subkeys, where each subkey is used four times. In this algorithm, the key words are used in order for the first 24 rounds and they are used in reverse order for the last 8 rounds.

Kuznyechik is the latest extension of GOST, which uses 128-bit blocks.

Camellia

Camellia is a symmetric-key block cipher having either 18 rounds (for 128-bit keys) or 24 rounds (for 256-bit keys). It is a Feistel cipher with a block size of 128 bits and a key size of 128, 192, and 256 bits. Camellia uses four 8×8-bit S-boxes that perform affine transformations and logical operations. A logical transformation layer FL-function or its inverse is applied every six rounds. Camellia uses the key whitening technique for increased security.

Camellia is a part of the Transport Layer Security (TLS) protocol, which is used to deliver secure communication. Camellia cannot be brute-forced even with the latest technology although it uses a smaller key size of 128 bits, thus making it a safe cipher. In addition, Camellia offers high security and its processing skills are equivalent to those of AES or Rijndael.



Asymmetric Encryption Algorithms

The table below shows specified asymmetric encryption algorithms, including information such as key size and application areas.

Algorithm	Key Size (bits)	Application Areas
Rivest–Shamir–Adleman (RSA)	Variable	Encryption, digital signatures, key exchange
Digital Signature Algorithm (DSA)	Variable	Digital signatures
Diffie-Hellman	Variable	Key exchange, secure communication
Elliptic Curve Cryptography (ECC)	160-521 bits	Encryption, digital signatures, key exchange
ElGamal	Variable	Encryption, key exchange

Table 20.3: Asymmetric encryption algorithms

DSA and Related Signature Schemes

The Digital Signature Algorithm (DSA) is a Federal Information Processing Standard for digital signatures. The NIST proposed the DSA for use in the Digital Signature Standard (DSS), adopted as FIPS 186. The DSA helps in the generation and verification of digital signatures for sensitive and unclassified applications. It creates a 320-bit digital signature with 512–1024-bit security.

A digital signature is a mathematical scheme used for the authentication of digital messages. Computation of the digital signature uses a set of rules (i.e., the DSA) and a set of parameters in that the user can verify the identity of the signatory and the integrity of the data.

Processes involved in DSA:

- **Signature Generation Process**: The private key is used to know who has signed it.
- **Signature Verification Process**: The public key is used to verify whether the given digital signature is genuine.

DSA is a public-key cryptosystem, as it involves the use of both private and public keys.

Benefits of DSA:

- Less chances of forgery compared with a written signature
- Quick and easy method of business transactions
- Fake currency problem can be mitigated considerably

DSA Algorithm:

Each entity A does the following:

1. Select a prime number q such that $2^{159} < q < 2^{160}$
2. Choose t such that $0 \leq t \leq 8$, and select a prime number p where $2^{511+64t}<p< 2^{512+64t}$, with the property that q divides $(p-1)$
3. Select a generator α of the unique cyclic group of order q in Z^*_p by choosing an element $g \in Z^*_p$ and then computing $\alpha = g^{(p-1)/q} \bmod p$ until $\alpha \neq 1$
4. Select a random integer d such that $1 \leq d \leq q-1$
5. Compute $y = \alpha^d \bmod p$
6. A's public key is (p, q, α, y); A's private key is d.

To sign a message m, A does the following:

1. Select a random secret integer k, $0<k<q$.
2. Compute $r=(\alpha^k \bmod p) \bmod q$
3. Compute $k^{-1} \bmod q$
4. Compute $s=k^{-1}\{h(m) + dr\} \bmod q$, where h is the Secure Hash Algorithm
5. A's signature for m is the pair (r, s)

To verify A's signature (r, s) on m, B should do the following:

1. Obtain A's authentic public key (p,q,α,y)
2. Verify that $0<r<q$ and $0<s<q$; if not, then reject the signature
3. Compute $w = s^{-1} \bmod q$ and $h(m)$
4. Compute $u_1 = w*h(m) \bmod q$ and $u_2 = rw \bmod q$
5. Compute $v = (\alpha^{u1} y^{u2} \bmod p) \bmod q$
6. Accept the signature if and only if $v=r$

Rivest Shamir Adleman (RSA)

Ron Rivest, Adi Shamir, and Leonard Adleman formulated RSA, a public-key cryptosystem for Internet encryption and authentication. RSA uses modular arithmetic and elementary number theories to perform computations using two large prime numbers. The RSA system is widely used in a variety of products, platforms, and industries. It is one of the de-facto encryption standards. Companies such as Microsoft, Apple, Sun, and Novell build RSA algorithms into their operating systems. RSA can also be found on hardware-secured telephones, Ethernet network cards, and smart cards.

RSA works as follows:

1. Two large prime numbers are taken (a and b), and their product is determined (c = ab, where "c" is called the modulus).
2. RSA chooses a number "e" that it is less than "c" and relatively prime to (a-1)(b-1). Therefore, e and (a-1)(b-1) have no common factor except 1.
3. Furthermore, RSA chooses a number "f" such that (ef - 1) is divisible by (a-1)(b-1).
4. The values "e" and "f" are the public and private exponents, respectively.
5. The public key is the pair (c, e); the private key is the pair (c, f).
6. It is difficult to obtain the private key (c, f) from the public key (c, e). However, if someone can factor "c" into "a" and "b", then that person can decipher the private key (c, f).

The security of the RSA system depends on the assumption that such factoring is difficult to carry out, making the cryptographic technique safe.

An example of how cryptography uses RSA algorithms in a practical interchange is illustrated by the following sequence:

1. The sender of a message encrypts it using a randomly chosen DES symmetric key. DES (Data Encryption Standard) is a relatively insecure symmetric-key system using 64-bit encryption (56 bits for key size, 8 bits for cyclic redundancy check) to encrypt data.
2. The sender will then look up the recipient's public key and use it to encrypt the DES key using the RSA system.
3. The sender transmits an RSA digital envelope, consisting of a DES-encrypted message and an RSA-encrypted DES key, to the recipient.
4. The recipient will decrypt the DES key and then use the DES key to decrypt the message itself.

This system combines the high speed of DES with the key management convenience of the RSA system.

RSA Signature Scheme

Cryptography uses RSA for public key encryption and for a digital signature (to sign a message and verify it). The RSA signature scheme is the first technique used to generate digital signatures.

It is a deterministic digital signature scheme that provides message recovery from the signature itself, making it the most practical and versatile technique available.

RSA involves both a public key and a private key. The public key, as the name indicates, can be used by anyone for encrypting messages. The messages that the user encrypts with the public key require the private key for decryption.

Consider that John encrypts his document M using his private key S_A, thereby creating a signature $S_{john}(M)$. John sends M along with the signature $S_{john}(M)$ to Alice. Alice decrypts the document using John's public key, thereby verifying John's signature.

RSA Key Generation

The procedure for RSA key generation is common to all the RSA-based signature schemes. To generate an RSA key pair, i.e., both an RSA public key and the corresponding private key, each entity A should do the following:

- Generate two large distinct primes p and q arbitrarily, each with roughly the same bit length
- Compute $n = pq$ and $\phi = (p-1)(q-1)$
- Choose a random integer e, $1 < e < \phi$, such that $\gcd(e, \phi) = 1$

  GCD = Greatest Common Divisor
- Use the extended Euclidean algorithm to compute the unique integer d, $1 < d < \phi$, such that $ed \equiv 1 \pmod{\phi}$
- A's public key is (n, e); A's private key is d

Destroy p and q at the end of the key generation

The RSA algorithm generates and verifies the RSA signature as follows way:

Entity A signs a message $m \in M$. Any entity B can verify A's signature and recover the message m from the signature.

1. **Signature Generation**

   To sign a message m, entity A should do the following:

   - Compute $\tilde{m} = R(m)$, an integer in the range $[0, n-1]$
   - Compute $s = \tilde{m}^d \bmod n$
   - A's signature form is s

2. **Signature Verification**

   To verify A's signature s and recover the message m, B should do the following:

   - Obtain A's authentic public key (n, e)
   - Compute $\tilde{m} = s^e \bmod n$
   - Verify that $\tilde{m} \in M_R$; if not, reject the signature
   - Recover $m = R^{-1}(\tilde{m})$

Example of RSA Algorithm

The math underlying RSA public-key encryption is described below:

1. Find P and Q, two large (e.g., 1024-bit) prime numbers.
2. Choose E such that E is greater than 1, E is less than PQ, and E and (P-1)(Q-1) are relatively prime, which means that they have no prime factors in common. E does not have to be prime, but it must be odd. (P-1)(Q-1) cannot be prime because it is an even number.
3. Compute D such that (DE - 1) is evenly divisible by (P-1)(Q-1). Mathematicians write this as DE = 1 (mod (P-1)(Q-1)), and they call D the multiplicative inverse of E. This is easy to do—simply find an integer X that causes D = (X(P-1)(Q-1) + 1)/E to be an integer and then use that value of D.
4. The encryption function is C = (T^E) mod PQ, where C is the ciphertext (a positive integer), T is the plaintext (a positive integer), and ^ indicates exponentiation. During the encryption of the message, T must be less than the modulus, PQ.
5. The decryption function is T = (C^D) mod PQ, where C is the ciphertext (a positive integer), T is the plaintext (a positive integer), and ^ indicates exponentiation.

Your public key is the pair (PQ, E). Your private key is the number D (do not reveal it to anyone). The product PQ is the modulus. E is the public exponent. D is the secret exponent.

You can publish your public key freely because there are no known easy methods of calculating D, P, or Q given only (PQ, E) (your public key).

Given below is an example of the RSA algorithm:

P = 61<= first prime number (destroy this after computing E and D)

Q = 53<= second prime number (destroy this after computing E and D)

PQ = 3233<= modulus (give this to others)

E = 17<= public exponent (give this to others)

D = 2753<= private exponent (keep this secret)

Your **public** key is **(E,PQ)**

Your **private** key is **D**

The encryption function is:

encrypt(T) = (T^E) mod PQ

= (T^17) mod 3233

The decryption function is:

decrypt(C) = (C^D) mod PQ

= (C^2753) mod 3233

To encrypt the plaintext value 123, do this:

encrypt(123) = (123^17) mod 3233

= 337587917446653715596592958817679803 mod 3233

= 855

To decrypt the ciphertext value 855, do this:

decrypt(855) = (855*2753) mod 3233

= 123

One way to compute the value of **855^2753 mod 3233** is as follows:

Consider these powers of 855:

- 855^1 = 855 (mod 3233)
- 855^2 = 367 (mod 3233)
- 855^4 = 367^2 (mod 3233) = 2136 (mod 3233)
- 855^8 = 2136^2 (mod 3233) = 733 (mod 3233)
- 855^16 = 733^2 (mod 3233) = 611 (mod 3233)
- 855^32 = 611^2 (mod 3233) = 1526 (mod 3233)
- 855^64 = 1526^2 (mod 3233) = 916 (mod 3233)
- 855^128 = 916^2 (mod 3233) = 1709 (mod 3233)
- 855^256 = 1709^2 (mod 3233) = 1282 (mod 3233)
- 855^512 = 1282^2 (mod 3233) = 1160 (mod 3233)
- 855^1024 = 1160^2 (mod 3233) = 672 (mod 3233)
- 855^2048 = 672^2 (mod 3233) = 2197 (mod 3233)

Given the above, we know the following:

855^2753 (mod 3233)

= 855^(1 + 64 + 128 + 512 + 2048) (mod 3233)

= 855^1 * 855^64 * 855^128 * 855^512 * 855^2048 (mod 3233)

= 855 * 916 * 1709 * 1160 * 2197 (mod 3233)

= 794 * 1709 * 1160 * 2197 (mod 3233)

= 2319 * 1160 * 2197 (mod 3233)

= 184 * 2197 (mod 3233)

= 123 (mod 3233)

= 123

Diffie–Hellman

It is a cryptographic protocol that allows two parties to establish a shared key over an insecure channel. It was developed and published by Whitfield Diffie and Martin Hellman in 1976. Actually, it was independently developed a few years earlier by Malcolm J. Williamson of the British Intelligence Service, but it was classified at that time.

Diffie–Hellman Algorithm

The system has two parameters called *p* and *g*

- Parameter *p* is a prime number
- Parameter *g* (usually called a generator) is an integer less than *p*, with the following property: for every number *n* between 1 and *p-1* (both inclusive), there is a power k of g such that $n = g^k mod\ p$

Many cryptography textbooks use the fictitious characters "Alice" and "Bob" to illustrate cryptography; we will do the same here as well:

- Alice generates a random private value **a**, and Bob generates a random private value **b**. Both a and b are drawn from the **set of integers**
- They derive their public values using parameters p and g and their private values. Alice's public value is $g^a \bmod p$, and Bob's public value is $g^b \bmod p$.
- They exchange their public values
- Alice computes $g^{ab} = (g^b)^a \bmod p$, and Bob computes $g^{ba} = (g^a)^b \bmod p$
- Since $g^{ab} = g^{ba} = k$, Alice and Bob now have a shared secret key **k**

The Diffie–Hellman algorithm does not provide any authentication for the key exchange and is vulnerable to many cryptographic attacks. Nevertheless, it is the basis of many authentication mechanisms; for example, it provides forward secrecy in the TLS protocol's ephemeral modes depending on the cipher spec.

Elliptic Curve Cryptography (ECC)

ECC is a modern public-key cryptography developed to avoid larger cryptographic key usage. The asymmetric cryptosystem depends on number theory and mathematical elliptic curves (algebraic structure) to generate short, quick, and robust cryptographic keys. RSA is an incumbent public-key algorithm, but its key size is large. The speed of the encryption always depends on the key size: a smaller key length allows faster encryption. To minimize the key size, elliptic curve cryptography has been proposed as a replacement for the RSA algorithm.

The operational key sizes of both algorithms to achieve similar goals are listed below:

ECC Key size	RSA Key Size
160-223	1024
224-255	2048
256-383	3072
384-511	7680
512+	15360

Table 20.4: Comparison of ECC and RSA key size

While RSA uses a key size of 1024 to encrypt the data, ECC provides equal security with a comparatively smaller key size ranging between 160 to 223. For high-level computing, RSA uses a key size of 7680 to implement security, whereas ECC can provide the same level of security with a key size ranging between 384 to 511.

YAK

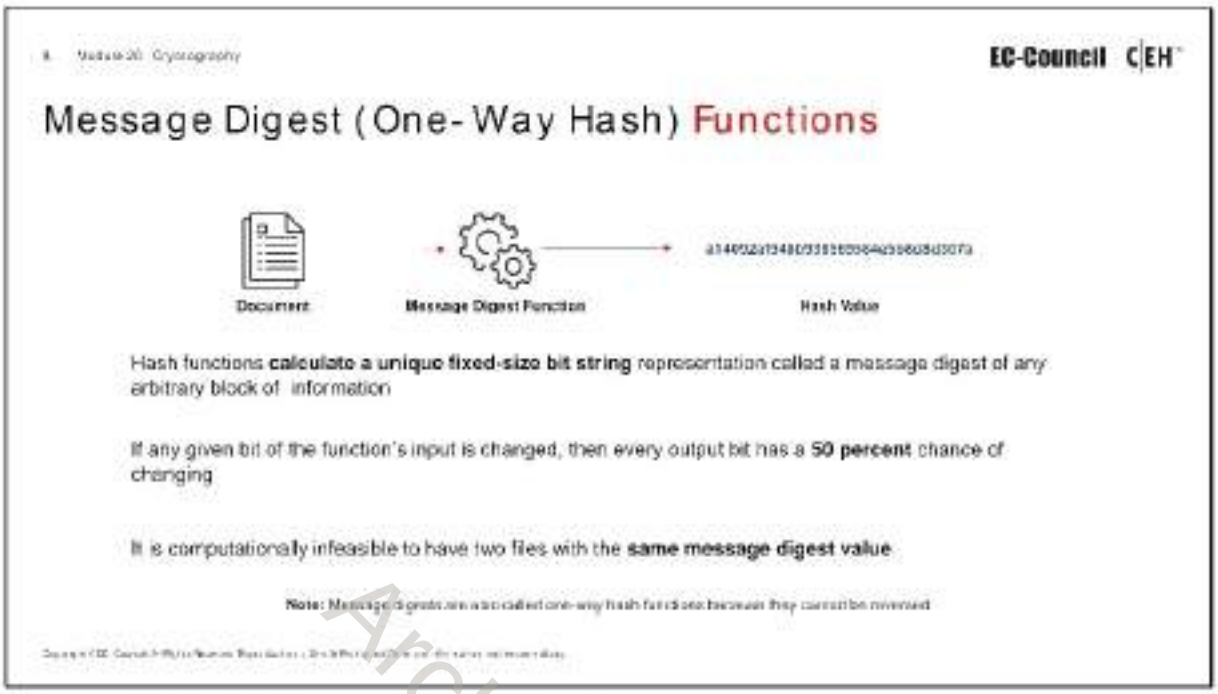
YAK is a public-key-based Authenticated Key Exchange (AKE) protocol. The authentication of YAK is based on public key pairs, and it needs PKI to distribute authentic public keys. YAK is a variant of the two-pass Hashed Menezes-Qu-Vanstone (HMQV) protocol using zero-knowledge proofs (ZKP) for proving the knowledge of ephemeral secret keys from both parties. The YAK protocol lacks joint key control and perfect forward secrecy attributes.

The YAK protocol implementation between two parties Alice and Bob is described as follows:

1. Alice chooses a random number **x** such that $x \in_R [0, q-1]$, computes $X = g^x$, and generates ZKP of **x**, denoted by **KP{x}**. Alice sends **X** and **KP{x}** to Bob.
2. Bob chooses a random number **y** such that $y \in_R [0, q-1]$, computes $Y = g^y$, and generates ZKP of **y**, denoted by **KP{y}**. Bob sends **Y** and **KP{y}** to Alice.
3. Alice verifies the received **KP{x}** and computes the session key after verification as $k = H((Y.PK_B)^{x+a})$, where **H** is a hash function.
4. Bob verifies the received **KP{y}** and computes the session key after verification as $k = H((X.PK_A)^{y+b})$.
5. They authenticate each other, and both obtain the same session key $k = H(g^{(x+a)(y+b)})$.

The YAK protocol can accomplish the following objectives:

- Private key security
- Full forward secrecy
- Session key security



Message Digest (One-way Hash) Functions

Hash functions calculate a unique fixed-size bit string representation, called a message digest, of any arbitrary block of information. Message digest functions distill the information contained in a file (small or large) into a single fixed-length number, typically between 128 and 256 bits. If any given bit of the function's input is changed, every output bit has a 50% chance of changing. Given an input file and its corresponding message digest, it should be nearly impossible to find another file with the same message digest value, as it is computationally infeasible to have two files with the same message digest value.

Message digest functions are also called one-way hash functions because they produce values that are nearly impossible to invert, resistant to attack, mostly unique, and widely distributed. Message digest algorithms themselves do not participate in encryption and decryption operations. They allow the creation of digital signatures and message authentication codes (MACs) as well as the derivation of encryption keys from passphrases.

The main role of a cryptographic hash function is to provide integrity in document management. Cryptographic hash functions are an integral part of digital signatures. They are relatively faster than digital signature algorithms; hence, their characteristic feature is to calculate the signature of the document's hash value, which is smaller than the document. In addition, digests help to hide the contents or source of the document.

Widely used message digest functions include the following algorithms:

- MD5
- SHA

Note: Message digests are also called one-way hash functions because they cannot be reversed.

Figure 20.8: Block Diagram of One-way Hash Function

Module 20: Cryptography

EC-Council C|EH

Message Digest Functions

Algorithm	Output Size (bits)	Internal State Size (bits)	Block Size (bits)	Max Message Size (bits)	Rounds	Operations	Security (bits)	Application Areas
MD2	128	128	128	2^64	18	Permutation, Substitution	128	Legacy applications, checksum validation
MD4	128	128	512	2^64	48	Logical operations (AND, OR, XOR)	64	Obsolete, early cryptographic hash functions
MD5	128	128	512	2^64	64	Logical operations (AND, OR, XOR)	64	File verification, checksum, digital signatures
MD6	224, 256, 384, 512	1024	512	Unlimited	Variable	Logical operations (AND, OR, XOR)	128-256	Cryptographic applications, data integrity
SHA-0	160	160	512	2^64	80	Bitwise logical operations	0	Obsolete, replaced by SHA-1
SHA-1	160	160	512	2^64	80	Bitwise logical operations	80	Legacy systems, software updates, TLS
SHA-2	224, 256, 384, 512	256, 512	512, 1024	2^128	64, 80	Logical operations (AND, OR, XOR)	112-256	Secure applications, digital signatures, SSL
SHA-3	224, 256, 384, 512	1600	1088, 576	Unlimited	Variable	Sponge construction	112-256	Secure applications, next-generation cryptographic functions
RIPEMD-160	160	160	512	2^64	160	Logical operations (AND, OR, XOR)	80	Cryptographic applications, data integrity
WHIRLPOOL	512	512	512	2^256	10	Matrix operations (block cipher)	256	Secure hashing, cryptographic applications
Tiger	192	192	512	Unlimited	24	Logical operations (AND, OR, XOR)	192	High-speed applications, checksum validation
BLAKE2	256, 512	256, 512	512, 1024	Unlimited	10-14	Logical operations (AND, OR, XOR)	128-256	High-speed hashing, secure applications
BLAKE3	256	256	512	Unlimited	Variable	Logical operations (AND, OR, XOR)	128	High-performance cryptographic applications

Message Digest Functions

The table below shows message digest functions, detailing their output size, internal state size, block size, maximum message size, rounds, operations, security level, and application areas:

Algorithm	Output Size (bits)	Internal State Size (bits)	Block Size (bits)	Max Message Size (bits)	Rounds	Operations	Security (bits)	Application Areas
MD2	128	128	128	2^64	18	Permutation, Substitution	128	Legacy applications, checksum validation
MD4	128	128	512	2^64	48	Logical operations (AND, OR, XOR)	64	Obsolete, early cryptographic hash functions
MD5	128	128	512	2^64	64	Logical operations (AND, OR, XOR)	64	File verification, checksum, digital signatures
MD6	224, 256, 384, 512	1024	512	Unlimited	Variable	Logical operations (AND, OR, XOR)	128-256	Cryptographic applications, data integrity

SHA-0	160	160	512	2^64	80	Bitwise logical operations	0	Obsolete, replaced by SHA-1
SHA-1	160	160	512	2^64	80	Bitwise logical operations	80	Legacy systems, software updates, TLS
SHA-2	224, 256, 384, 512	256, 512	512, 1024	2^128	64, 80	Logical operations (AND, OR, XOR)	112-256	Secure applications, digital signatures, SSL
SHA-3	224, 256, 384, 512	1600	1088, 576	Unlimited	Variable	Sponge construction	112-256	Secure applications, next-generation cryptographic functions
RIPEMD-160	160	160	512	2^64	160	Logical operations (AND, OR, XOR)	80	Cryptographic applications, data integrity
WHIRLPOOL	512	512	512	2^256	10	Matrix operations, substitution	256	Secure hashing, cryptographic applications
Tiger	192	192	512	Unlimited	24	Logical operations (AND, OR, XOR)	192	High-speed applications, checksum validation
BLAKE2	256, 512	256, 512	512, 1024	Unlimited	10-14	Logical operations (AND, OR, XOR)	128-256	High-speed hashing, secure applications
BLAKE3	256	256	512	Unlimited	Variable	Logical operations (AND, OR, XOR)	128	High-performance cryptographic applications

Table 20.5: Message digest functions

Message Digest Function: MD5 and MD6

MD2, MD4, MD5, and MD6 are **message digest algorithms** used in digital signature applications to compress a document securely before the system signs it with a private key. The algorithms can be of variable length, but the resulting message digest always has a size of 128 bits.

The structures of all three algorithms (MD2, MD4, and MD5) appear similar, although the design of MD2 is reasonably different from that of MD4 and MD5. MD2 supports 8-bit machines, while MD4 and MD5 support 32-bit machines. The algorithm pads the message with extra bits to

ensure that the number of bits is divisible by 512. The extra bits may include a 64-bit binary message.

Attacks on versions of MD4 have become increasingly successful. Research has shown how an attacker launches collision attacks on the full version of MD4 within a minute on a typical PC. MD5 is slightly more secure but is slower than MD4. However, both the message digest size and the padding requirements remain the same.

MD5 is a widely used cryptographic hash function that takes a message of arbitrary length as input and outputs a 128-bit (16-byte) fingerprint or message digest of the input. MD5 can be used in a wide variety of cryptographic applications and is useful for digital signature applications, file integrity checking, and storing passwords. However, MD5 is not collision-resistant; therefore, it is better to use the latest algorithms, such as MD6, SHA-2, and SHA-3.

MD6 uses a Merkle-tree-like structure to allow for large-scale parallel computation of hashes for very long inputs. It is resistant to differential cryptanalysis attacks.

To calculate the effectiveness of hash functions, check the output produced when the algorithm randomizes an arbitrary input message.

The following are examples of minimally different message digests:

- echo "There is CHF1500 in the blue bo" | md5sum
  e41a323bdf20eadafd3f0e4f72055d36
- echo "There is CHF1500 in the blue box" | md5sum
  7a0da864a41fd0200ae0ae97afd3279d
- echo "There is CHF1500 in the blue box." | md5sum
  2db1ff7a70245309e9f2165c6c34999d

Even minimally different texts produce radically different MD5 codes.

Figure 20.9: Verifying MD5 Hash

- **QuickHash-GUI**

  Source: *https://www.quickhash-gui.org*

  QuickHash-GUI is a graphical interface data hashing tool for Linux, Windows, and macOS. The tool is capable of hashing segments of text or dynamic hashing as you type into the text field.

Message Digest Function: Secure Hashing Algorithm (SHA)

The NIST has developed the Secure Hash Algorithm (SHA), specified in the **Secure Hash Standard (SHS)** and published as a federal information-processing standard (FIPS PUB 180). It generates a cryptographically secure one-way hash. Rivest developed the SHA, which is similar to the message digest algorithm family of hash functions. It is slightly slower than MD5, but its larger message digest makes it more secure against brute-force collision and inversion attacks.

SHA encryption is a series of five different cryptographic functions, and it currently has three generations: SHA-1, SHA-2, and SHA-3.

- **SHA-0**: A retronym applied to the original version of the 160-bit hash function published in 1993 under the name SHA, which was withdrawn from trade due to an undisclosed "**significant flaw**" in it. It was replaced with a slightly revised version, namely SHA-1.
- **SHA-1**: It is a 160-bit hash function that resembles the former MD5 algorithm developed by Ron Rivest. It produces a 160-bit digest from a message with a maximum length of (264 – 1) bits. It was designed by the National Security Agency (NSA) to be part of the Digital Signature Algorithm (DSA). It is most commonly used in security protocols such as PGP, TLS, SSH, and SSL. As of 2010, SHA-1 is no longer approved for cryptographic use because of its cryptographic weaknesses.
- **SHA-2**: SHA2 is a family of two similar hash functions with different block sizes, namely SHA-256, which uses 32-bit words, and SHA-512, which uses 64-bit words. The truncated versions of each standard are SHA-224 and SHA-384.
- **SHA-3**: SHA-3 uses sponge construction in which message blocks are XORed into the initial bits of the state, which the algorithm then invertibly permutes. It supports the same hash lengths as SHA-2 but differs in its internal structure considerably from the rest of the SHA family.

RIPEMD-160

RACE Integrity Primitives Evaluation Message Digest (RIPEMD) is a 160-bit hash algorithm developed by Hans Dobbertin, Antoon Bosselaers, and Bart Preneel. There exist 128-, 256-, and 320-bit versions of this algorithm, called RIPEMD-128, RIPEMD-256, and RIPEMD-320, respectively. These algorithms replace the original RIPEMD, which was found to have a collision issue. They do not follow any standard security policies or guidelines.

RIPEMD-160 is a more secure version of the RIPEMED algorithm. In this algorithm, the compression function consists of 80 stages, i.e., 5 blocks that execute 16 times each. This process repeats twice by combining the results at the bottom using modulo 32 addition.

HMAC

Hash-based message authentication code (HMAC) is a type of message authentication code (MAC) that uses a cryptographic key along with a cryptographic hash function. It is widely used to verify the integrity of data and authentication of a message. This algorithm includes an embedded hash function such as SHA-1 or MD5. The strength of HMAC depends on the embedded hash function, key size, and size of the hash output.

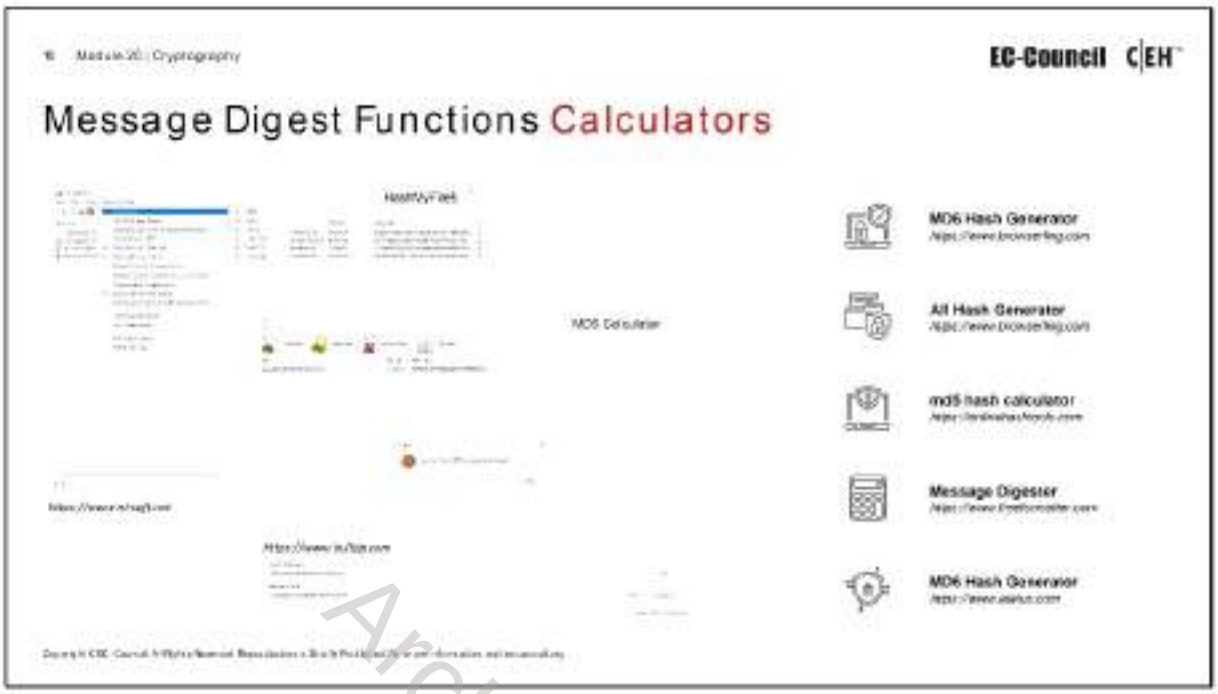
HMAC includes two stages for computing the hash. The input key is processed to produce two keys, namely the inner key and the outer key. The first stage of the algorithm inputs the inner key and message to produce an internal hash. The second stage of the algorithm inputs the output from the first stage and outer key, and produces the final HMAC code.

As HMAC executes the underlying hash function twice, it offers protection against various length extension attacks. The size of the key and the output depends on the embedded hash function; e.g., 128 or 160 bits in the case of MD5 or SHA-1, respectively.

GOST – Hash Function

This hash algorithm was initially defined in the Russian national standard GOST R 34.11-94 "Information Technology - Cryptographic Information Security - Hash Function."

It produces a fixed-length output of 256 bits. The input message is broken up into chunks of 256-bit blocks. If a block is less than 256 bits, then the message is padded by appending as many zeros to it as are required to make the length of the message 256 bits. The remaining bits are filled with a 256-bit integer arithmetic sum of all previously hashed blocks. Then, a 256-bit integer representing the length of the original message, in bits, is produced.



Message Digest Functions Calculators

Message digest functions calculators that use different hash algorithms to convert plaintext into its equivalent hash value are discussed below.

- **MD5 Calculator**

  Source: *https://www.bullzip.com*

  MD5 Calculator is a simple application that calculates the MD5 hash of a given file. It can be used with large files (e.g., several gigabytes in size). It features a progress counter and a text field from which the final MD5 hash can be easily copied to the clipboard. MD5 Calculator can be used to check the integrity of a file.

  It allows you to calculate the MD5 hash value of the selected file. Right-click the file and choose "**MD5 Calculator;**" the program will calculate the MD5 hash. The MD5 Digest field contains the calculated value. To compare this MD5 digest with another, one can paste the other value into the Compare To field. Obviously, an equal to sign ("=") appears between the two values if they are equal; otherwise, the less than ("<") or greater than (">") sign will tell you that the values are different.

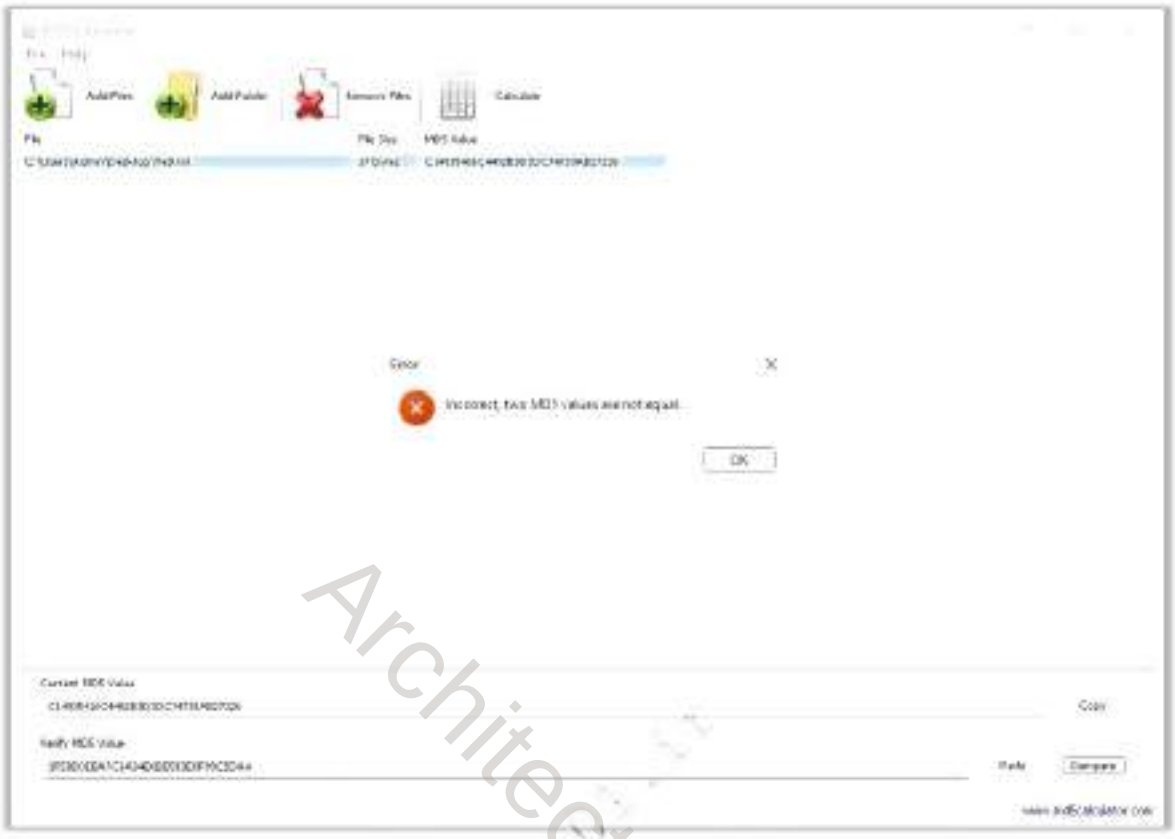


Figure 20.10: Screenshot of MD5 Calculator

- **HashMyFiles**

  Source: *https://www.nirsoft.net*

  HashMyFiles is a utility that allows you to calculate the MD5 and SHA1 hashes of one or more files in the system. It allows you to copy the MD5/SHA1 hash list to the clipboard or save it in a text/html/xml file. You can launch HashMyFiles from the context menu of Windows Explorer and display the MD5/SHA1 hashes of the selected files or folders.

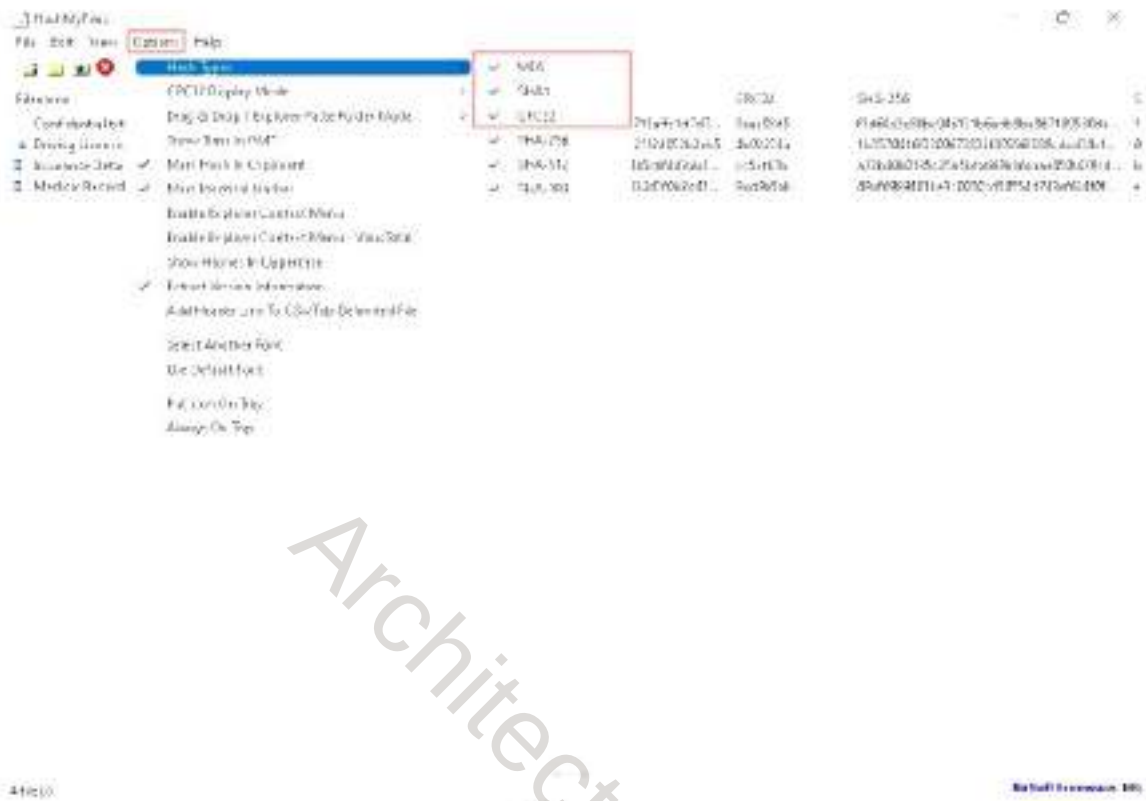
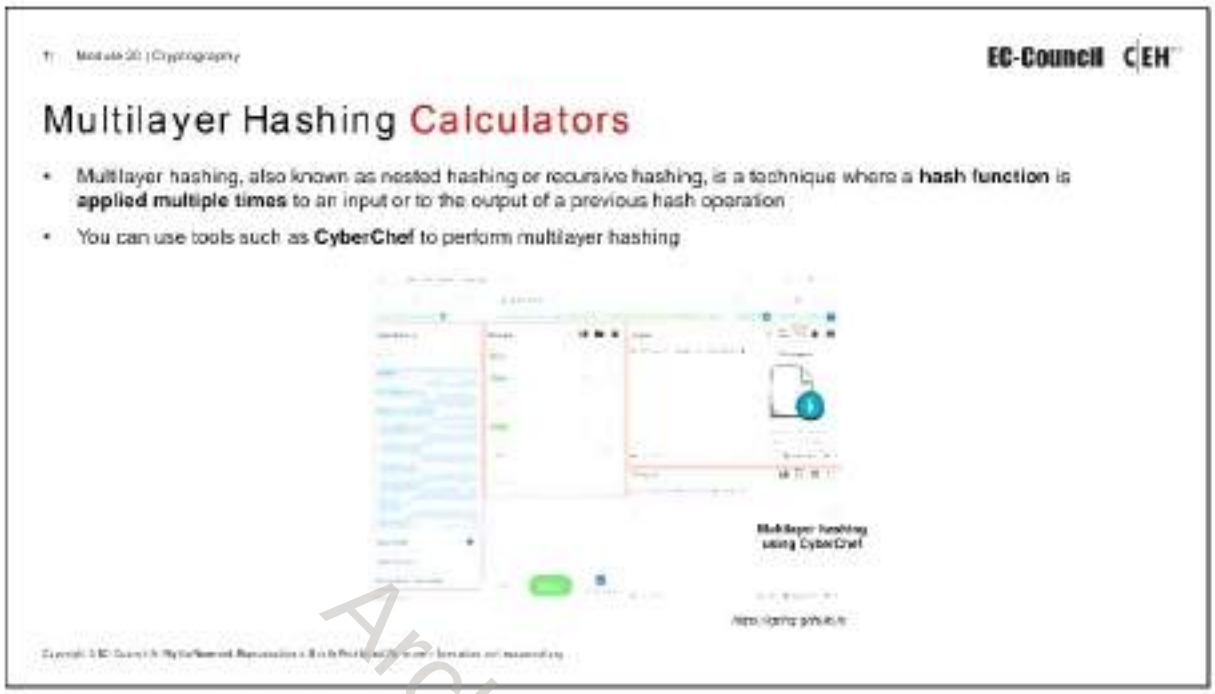


Figure 20.11: Screenshot of HashMyFiles

Some additional MD5 and MD6 hash calculators are as follows:

- MD6 Hash Generator (*https://www.browserling.com*)
- All Hash Generator (*https://www.browserling.com*)
- md5 hash calculator (*https://onlinehashtools.com*)
- Message Digester (*https://www.freeformatter.com*)
- MD6 Hash Generator (*https://www.atatus.com*)



Multilayer Hashing Calculators

Multilayer hashing, also known as nested hashing or recursive hashing, is a technique in which a hash function is applied multiple times to the input or output of a previous hash operation. This approach can enhance the security and create more complex hash structures. Tools such as CyberChef can be used to perform multilayer hashing.

This multilayer hashing process can make it more difficult for attackers to reverse engineer the original input data from the hash value, as it adds an additional layer of complexity, which makes brute-force attacks more challenging.

Working of Multilayer Hashing

- Initial Hashing: The original input data (e.g., a message or file) are hashed using a cryptographic hash function such as SHA-256, SHA-3, or MD5. The result is a fixed-size hash value that is typically a string of characters.
- Subsequent Hashing: The hash value obtained from the initial hashing step is again hashed using the same or different hash functions. This process can be repeated several times to create multiple hashing layers.
- Final Hash Value: After a predetermined number of hashing iterations, the final hash value is obtained. This value is used as the final output of the multilayer hashing process.

Steps to Perform Multilayer Hashing Using CyberChef

- **Step 1**: Create a sample file and open the CyberChef (*https://gchq.github.io/CyberChef*).
- **Step 2**: Click on the "Open file as input" option to upload the sample file.

- **Step 3**: Search for the desired hashing algorithm in the Operations panel (e.g., MD5), and drag and drop it into the Recipe panel. The output is shown in the Output panel.
- **Step 4**: Using the MD5 hash value as the input, select another hashing algorithm (e.g., SHA1) and drag and drop it into the Recipe panel.
- **Step 5**: Based on these requirements, select another hashing algorithm using the output of the SHA1 hash as the input and perform hashing recursively, as shown in the screenshot.

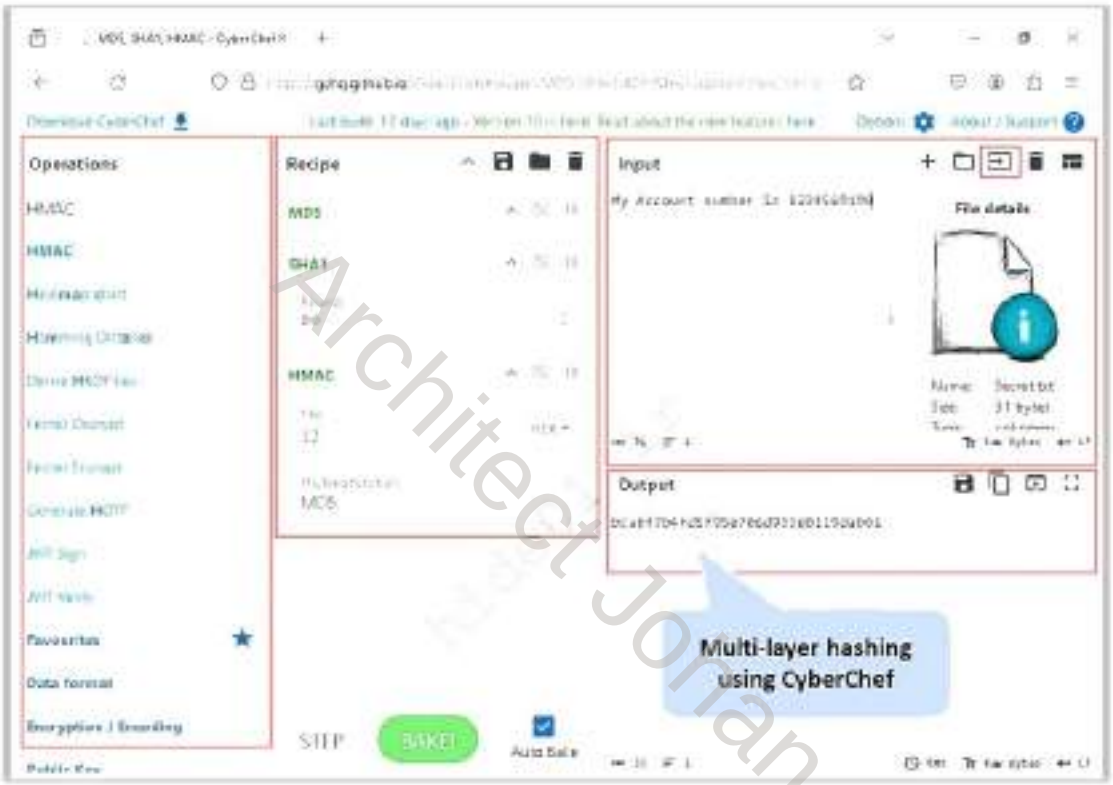


Figure 20.12: Screenshot of CyberChef performing multilayer hashing

Hardware-Based Encryption

Hardware-based encryption is a technique that uses computer hardware for assisting or replacing the software when the data encryption process is being performed. Devices that offer encryption techniques can be considered as hardware-based encryption devices. In the implementation of hardware-based encryption, the cryptography technique workload is transferred to the hardware processors, making the system resources free for performing other functions. These devices can also store encryption keys and other sensitive information in secured areas of RAM or other nonvolatile storage devices such as flash memory.

Hardware encryption devices reduce instruction sets, where only the authorized code can be executed. These devices do not support third-party software, thereby preventing the execution of any malicious programs. Hardware encryption offers many advantages over software encryption, as it can perform rapid processing of algorithm. It provides tamper-resistant key storage and avoids unauthorized code. Some hardware-based encryption devices are wireless access points, Nitrokey, credit card terminals, and network bulk encryptors.

Types of hardware encryption devices

- **TPM**

  Trusted Platform Module (TPM) is a crypto-processor or a chip that is present in the motherboard. It can securely store the encryption keys and perform many cryptographic operations. TPM offers various features such as authenticating platform integrity, providing full disk encryption capabilities, performing password storage, and providing software license protection.

- **HSM**

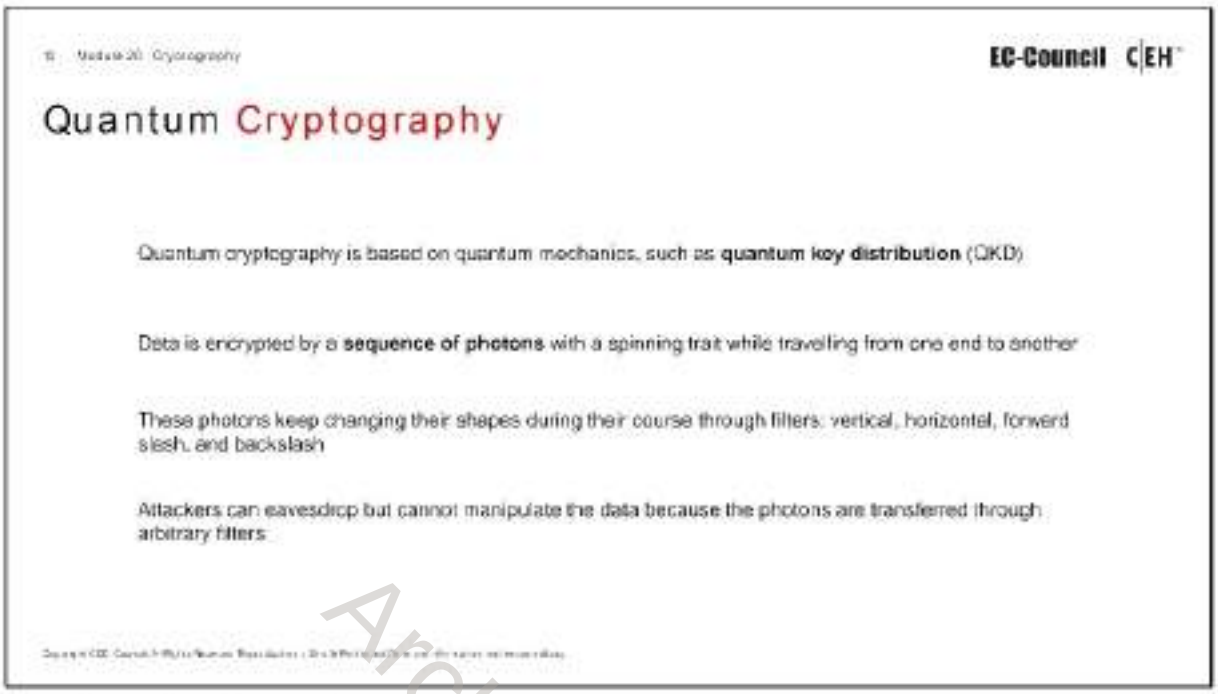
  A hardware security module (HSM) is an additional external security device that is used in a system for crypto-processing, and it can be used for managing, generating, and securely storing cryptographic keys. HSM offers enhanced encryption computation that is useful for symmetric keys longer than 256 bits. High-performance HSM devices are connected to the network using TCP/IP. Some HSM devices include Thales Luna Network HSM, nSheild HSM, Utimaco HSM, and Cryptosec Dekaton PCI.

- **USB Encryption**

  USB encryption is an additional feature for USB storage devices, which offers onboard encryption services. Encrypted USB devices need an on-device credential system or software- or hardware-based credentials from a computer. USB encryption provides protection against malware distribution over USB and helps in preventing data loss and data leakage. Some hardware USB-encrypted devices include Encrypted USB, Kingston Ironkey D300S, and diskAshur Pro.

- **Hard Drive Encryption**

  Hard drive encryption is a technology whereby the data stored in the hardware can be encrypted using a wide range of encryption options. Hard drive encryption devices cannot use an on-device keyboard or fingerprint reader; instead, they need a TPM or an HSM. These devices can be installed as an internal drive on a computer. Some hard drive encryption devices include military-grade 256-bit AES Hardware Encryption and DiskCypher AES Sata Hard Drive Encryption.



Quantum Cryptography

As the world is increasingly adopting online information sharing, cryptosystems are witnessing a sharp increase in security attacks. Since mathematical encryption uses binary digits (0 and 1), it can be easily eavesdropped on or manipulated using various techniques. Hence, quantum cryptography has been introduced to protect data against midway thefts (i.e., MITM attacks). This cryptography is processed based on quantum mechanics, such as quantum key distribution (QKD), using photons instead of mathematics as a part of encryption.

In quantum cryptography, the data elements are encrypted by a sequence of photons that have a spinning trait while traveling from one end to another end. These photons keep changing their shapes during their course through filters: vertical, horizontal, forward slash, and backslash. Here, vertical and backslash spins imply "ones," while horizontal and forward slash spins imply "zeros."

- Horizontal (–): 0
- Vertical (|): 1
- Backslash (/): 1
- Forward slash (\): 0

Attackers can eavesdrop on but cannot manipulate the data because the photons are transferred through arbitrary filters. To breach this mechanism, attackers must know the exact shape of the photons; if they fail to choose the right transmission, the photon polarization is distorted, and the receiver detects an error indicating the eavesdrop.

Other Encryption Techniques

- **Homomorphic Encryption**

  Homomorphic encryption differs from conventional encryption mechanisms, where math operations are performed to encrypt the plaintext. Homographic encryption allows users to secure and leave their data in an encrypted format even while it is being processed or manipulated. In this technique, encryption and decryption are performed by the same key holder. The homomorphic mechanism enables the user/sender to encrypt the confidential data and out-source it to an enterprise via cloud services to process the given data.

  How homomorphic encryption differs from other encryption mechanisms:

  In private key encryption:

  - Only keyholders can generate and decrypt ciphertexts using similar keys.

  In public key encryption:

  - Only the public keyholder generates the ciphertext and the secret keyholder decrypts the ciphertext.

  In homomorphic encryption:

  - The keyholder can generate the ciphertext and anyone can alter the ciphertext, but only the keyholder again can decrypt the data.

  The reason for using this cryptography is that an untrusted entity can manipulate the data. Hence, this mechanism allows the sender himself/herself to encrypt and decrypt the data, allowing anyone to perform mathematical operations on the ciphertext with respect to the rules applied by the sender.

- **Post-quantum Cryptography**

  Post-quantum cryptography is also known as quantum-resistant and quantum-proof cryptography, as it is an advanced cryptographic algorithm (mostly public-key based) designed to protect security systems against attacks initiated from both conventional and quantum computers. It can also work in conjunction with underlying communication protocols and operating networks. Moreover, post-quantum cryptography can serve as a stand-alone encryption algorithm that replaces current vulnerable cryptosystems complying with standard security policies.

  Post-quantum cryptography is intended to provide secure wide-range communication, secure secret-key processing, public-key-based signatures, and public-key-based encryption for high-end activities such as secure e-voting. The cryptography comprises several low-cost, secure systems and other systems that are usually employed for online communications. Post-quantum cryptography is aimed at preparedness for the age of quantum computing by updating specific algorithms and standards.

- **Lightweight Cryptography**

  A major challenge in current cryptography techniques is its usage in low-powered devices. Researchers are attempting to develop a compact algorithm that is quantum-safe and can be operated effectively on low-powered devices. Most current cryptographic algorithms are suitable for servers and desktops, but lightweight cryptographic algorithms are aimed at low-complexity applications such as RFID tags, sensor-based applications, and other IoT-based applications. The main objective of developing lightweight cryptography is to use less power and less resources without compromising security.

Cipher Modes of Operation

Cipher modes of operation, also known as block cipher modes of operation, are used to encrypt a fixed block of plaintext using a secret key and, in some modes, an initialization vector. These modes of operation can ensure the confidentiality and authenticity of data. The client and server exchange an encrypted symmetric key securely to facilitate encryption and decryption. Discussed below are the four block cipher modes of operation that explain how source-side encryption and destination-side decryption work.

- **Electronic Code Book (ECB) Mode**

  The ECB mode is a straightforward process of encryption and decryption that requires plaintext, a secret key, and a block cipher encryption algorithm. The plaintext is divided into a fixed length of blocks, which is equal to the size of the secret key. In the first stage, the encryption starts by taking the first block of the plaintext, and the secret key is taken as input to the block cipher encryption algorithm; the output is the first block of ciphertext. The process is repeated for all the plaintext blocks.

  On the destination side, decryption is performed in the same manner as generation of the first block of ciphertext. The secret key is taken as input to the block cipher decryption algorithm, which outputs the first block of plaintext. This process is repeated for all the ciphertext blocks. However, this mode has a flaw: if the equally partitioned blocks of plaintext contain the same data, then the output cipher blocks also contain the same ciphertext, providing analysts a chance to predict the plaintext.

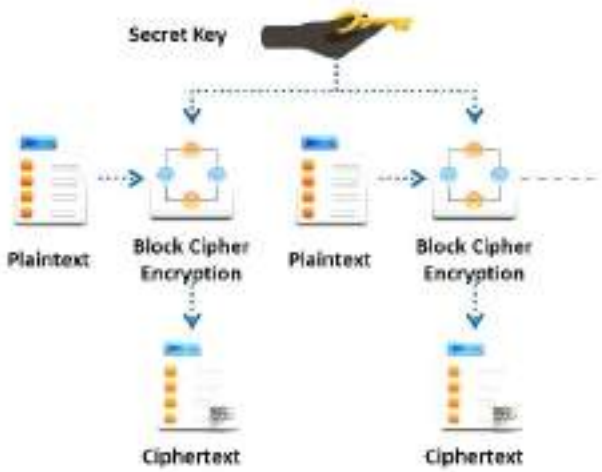


Figure 20.13: Electronic Code Book (ECB) mode encryption

- **Cipher Block Chaining (CBC) Mode**

  The CBC mode is an improvement over ECB that rectifies most of the security flaws in ECB. In the CBC mode, the process of encryption requires an initialization vector and a secret key. First, the plaintext is divided into blocks of the same size. The first block is XOR with the initialization vector (IV), and the resultant is sent as input to the block cipher encryption algorithm, along with the secret key. The output is the first block of ciphertext. This cipher block is used to perform XOR with the next plaintext block; the chain process continues till the last block of plaintext.

  On the destination side, the first block of ciphertext and secret key is sent to the block cipher decryption algorithm, and the result is XOR with the same IV. The output is the first block of plaintext. For the next cipher blocks, in the place of the IV, the previously used cipher block is input to perform XOR; this process continues for the remaining cipher blocks. However, this mode also has a problem: if one generated ciphertext block has an error, it propagates to the subsequent cipher blocks.

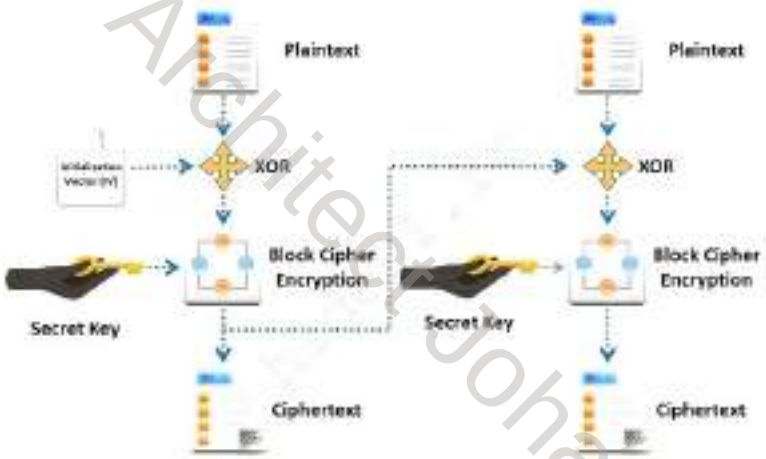


Figure 20.14: Cipher Block Chaining (CBC) mode encryption

- **Cipher Feedback (CFB) Mode**

  In the CFB mode, previously generated ciphertext is used as feedback for the encryption algorithm to encrypt the next plaintext block to ciphertext. First, the initialization vector (IV) is stored in a shift register and sent to the encryption algorithm, along with a secret key. From the result of that encryption, the first S bits are selected, and the XOR operation is performed with a plaintext block of size S. The resultant output is the ciphertext block. For the next encryption block, the previous cipher block is given as the input to the shift register; it shifts S bits to the left, and the process is continued till the end of the plaintext.

  On the destination side, the decryption process is the same till the XOR operation. The XOR operation is performed for the first S bits from the result of the encryption algorithm and the first cipher block, and the output is the first block of plaintext. For the subsequent blocks, the previously used cipher block is taken as the input for the shift register, and the process continues till the last cipher block. The advantage of this mode is that it makes cryptanalysis difficult as it has some data loss because of the use of shift registers.

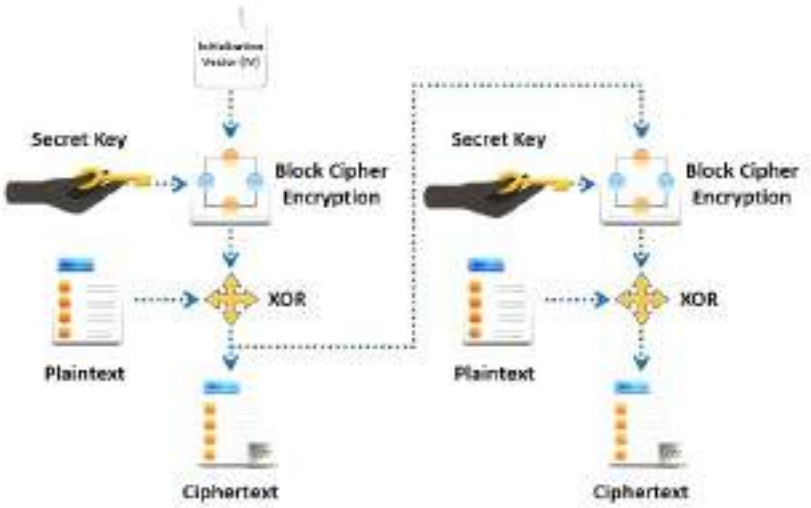


Figure 20.15: Cipher Feedback (CFB) mode encryption

- **Counter Mode**

  The counter mode is a block cipher mode of operation that uses a counter value in the encryption and decryption process. A counter value is initiated and sent as the input to the block cipher encryption algorithm with a secret key, and the result is subjected to the XOR operation with a block of plaintext. The output is the ciphertext block. This process is performed in a sequential manner to encrypt all the other plaintext blocks.

  On the destination side, this mode uses the same counter values and secret keys. The same encryption algorithm is used to encrypt the counter value and secret key, the result is subjected to the XOR operation with the obtained ciphertext block, and the output contains plaintext.

  The counter mode eliminates the problem of error propagation because it does not use previously generated ciphertext in encryption or decryption. The counter mode requires synchronized counter values on both the source and destination sides.

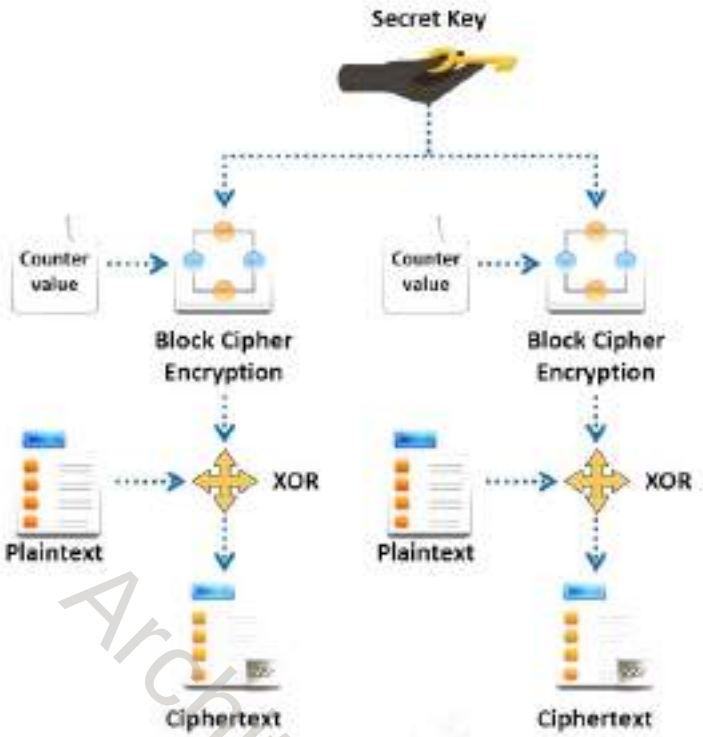


Figure 20.16: Counter mode encryption

Modes of Authenticated Encryption

Authenticated encryption (AE) modes of operation provide integrity and confidentiality for a transmitted message. In any cipher mode of operation, encryption/decryption is possible only with the shared secret key, preventing man-in-the-middle (MITM) attacks. However, the attacker can perform a chosen ciphertext attack to crack the encryption schema. The AE schema rectifies the problem of chosen cipher attacks. In AE modes, the ciphertext is combined with a message authentication code (MAC). Therefore, choosing a part in the ciphertext is not possible, and the AE scheme rejects improper ciphertexts while decrypting.

Authenticated encryption with Message Authentication Code (MAC)

A MAC is a value obtained by hashing a plaintext message using a shared secret key. It provides integrity for the message, and the receiver can verify the message using the hash value attached to it. The following are the three different ways of using a MAC while encrypting a message.

- **Encrypt-then-MAC (EtM)**

  In this approach, the plaintext is first encrypted using a secret key. For the obtained ciphertext, a hash value called message authentication code (MAC) is generated. The MAC is attached to the ciphertext and transmitted. This approach provides higher security for the transmitted message than other AE approaches.

- **Encrypt-and-MAC (E&M)**

  In the E&M approach, a MAC is first generated for the plaintext, following which the plaintext is encrypted using a secret key. Finally, both the ciphertext and MAC are combined and transmitted.

- **MAC-then-encrypt (MtE)**

  In the MtE approach, a MAC is first generated for the plaintext using the hash function, and the MAC is combined with the plaintext. The combination of the plaintext and MAC is encrypted with a secret key to produce ciphertext. The ciphertext contains the encrypted MAC.

Authenticated Encryption with Associated Data (AEAD)

AEAD is another approach used to ensure the integrity and authenticity of a message that contains both encrypted and unencrypted data. This approach adds additional data to the ciphertext at certain places to thwart chosen ciphertext attacks. The message header is kept unencrypted so that the receiver can verify the source of the message, and the payload is encrypted to ensure confidentiality.

Cryptography Tools

You can use various cryptographic tools for encrypting and decrypting your information, files, etc. These tools implement different types of encryption algorithms.

- **BCTextEncoder**

  Source: *https://www.jetico.com*

  The BCTextEncoder utility simplifies the encoding and decoding of text data. It compresses, encrypts, and converts plaintext data into text format, which the user can then copy to the clipboard or save as a text file. It uses public key encryption methods as well as password-based encryption. Furthermore, it uses strong and approved symmetric and public-key algorithms for data encryption.

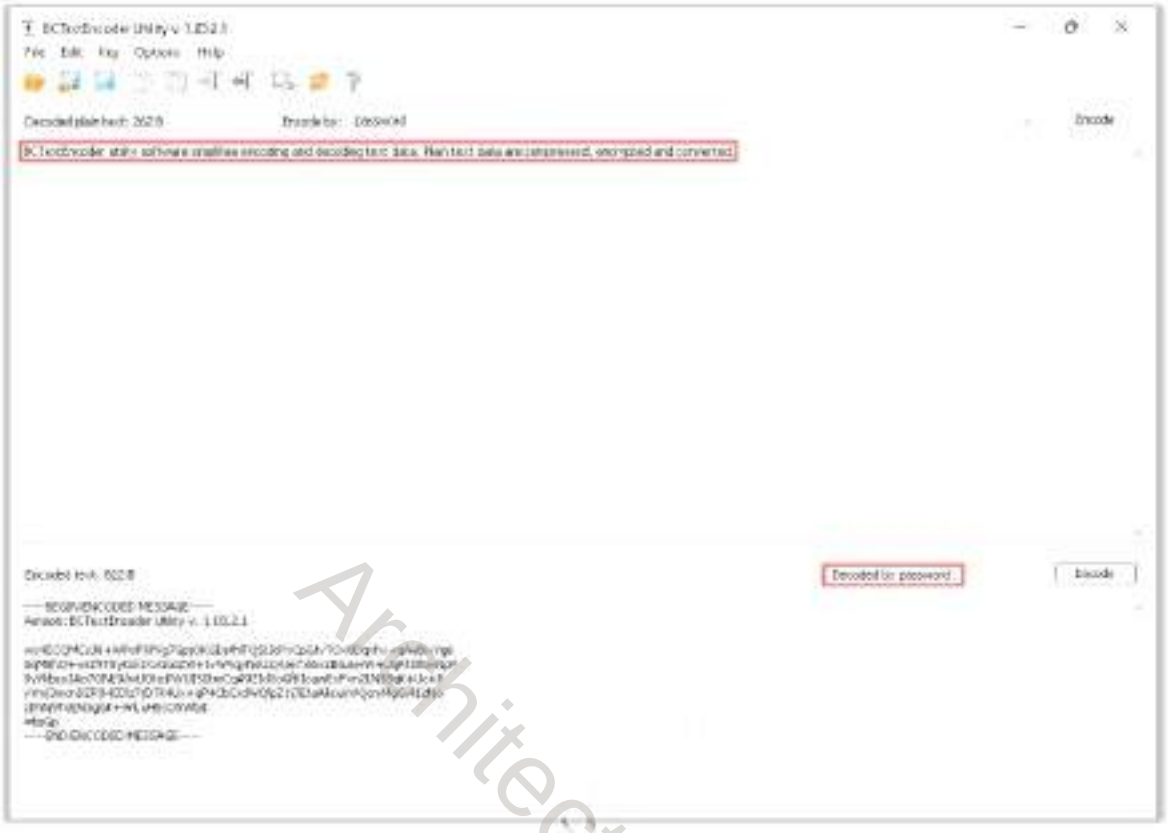


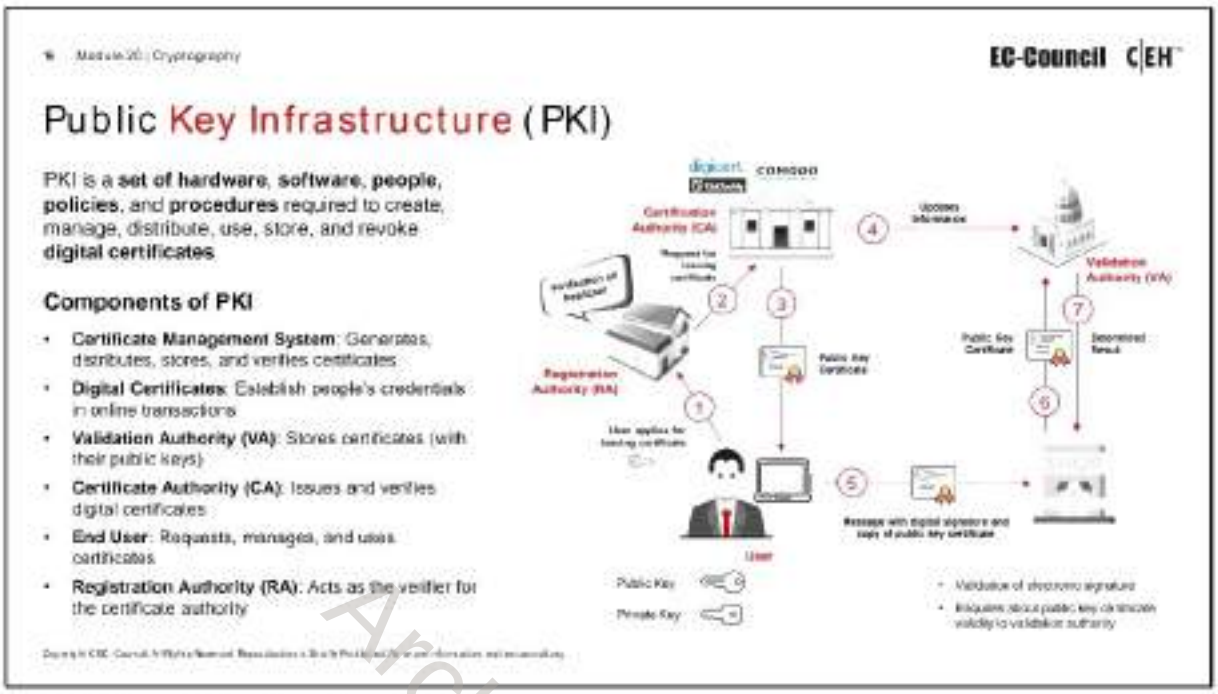
Figure 20.17: Screenshot of BCTextEncoder

Some additional cryptography tools are as follows:

- CryptoForge (*https://www.cryptoforge.com*)
- AxCrypt (*https://axcrypt.net*)
- Microsoft Cryptography Tools (*https://www.microsoft.com*)
- Concealer (*https://www.belightsoft.com*)
- SensiGuard (*https://www.sensiguard.com*)
- Cypherix (*https://www.cypherix.com*)

Applications of Cryptography

Cryptography plays a crucial role in securing communication and data in various applications across multiple domains. This section deals with key applications of cryptography, such as digital signature, Secure Sockets Layer (SSL), Transport Layer Security (TLS), Pretty Good Privacy (PGP), email encryption, disk encryption, and blockchain.



Public Key Infrastructure (PKI)

PKI is a security architecture developed to increase the confidentiality of information exchanged over the insecure Internet. It includes hardware, software, people, policies, and procedures required to create, manage, distribute, use, store, and revoke digital certificates. In cryptography, the PKI helps to bind public keys with corresponding user identities by means of a certification authority (CA).

Components of PKI

- **Certificate Management System**: Generates, distributes, stores, and verifies certificates
- **Digital Certificates**: Establishes credentials of a person when performing online transactions
- **Validation Authority (VA)**: Stores certificates (with their public keys)
- **Certification Authority (CA)**: Issues and verifies digital certificates
- **End User**: Requests, manages, and uses certificates
- **Registration Authority (RA)**: Acts as the verifier for the CA

PKI is a comprehensive system that allows the use of public-key encryption and digital signature services across a wide variety of applications. PKI authentication depends on digital certificates (also known as public-key certificates) that CAs sign and provide. A digital certificate is a digitally signed statement with a public key and the subject (user, company, or system) name in it.

PKI uses public-key cryptography, which is widely used on the Internet to encrypt messages or authenticate message senders. In public-key cryptography, a CA generates public and private keys with the same algorithm simultaneously. The private key is held only by the subject (user,

company, or system) mentioned in the certificate, while the public key is made publicly available in a directory that all parties can access. The subject keeps the private key secret and uses it to decrypt the text encrypted by someone else using the corresponding public key (available in a public directory). Thus, others encrypt messages for the user with the user's public key, and the user decrypts it with his/her private key.

The steps involved in the PKI process are as follows:

1. The subject (user, company, or system) intending to exchange information securely applies for a certificate to the registration authority (RA).
2. The RA receives the request from the subject, verifies the subject's identity, and requests the CA to issue a public key certificate to the user.
3. The CA issues the public key certificate binding the subject's identity with the subject's public key; then, the updated information is sent to the validation authority (VA).
4. When a user makes a transaction, the user duly signs the message digitally using his private key and sends the message to the client.

   (Here, the public key certificate is also included with the signed message, which allows the client to verify the signature.)
5. The client verifies the authenticity of the user by inquiring with the VA about the validity of the user's public key certificate (which was updated in step 3).
6. The VA compares the public key certificate of the user with that of the updated information provided by the CA and determines the certificate (valid or invalid).

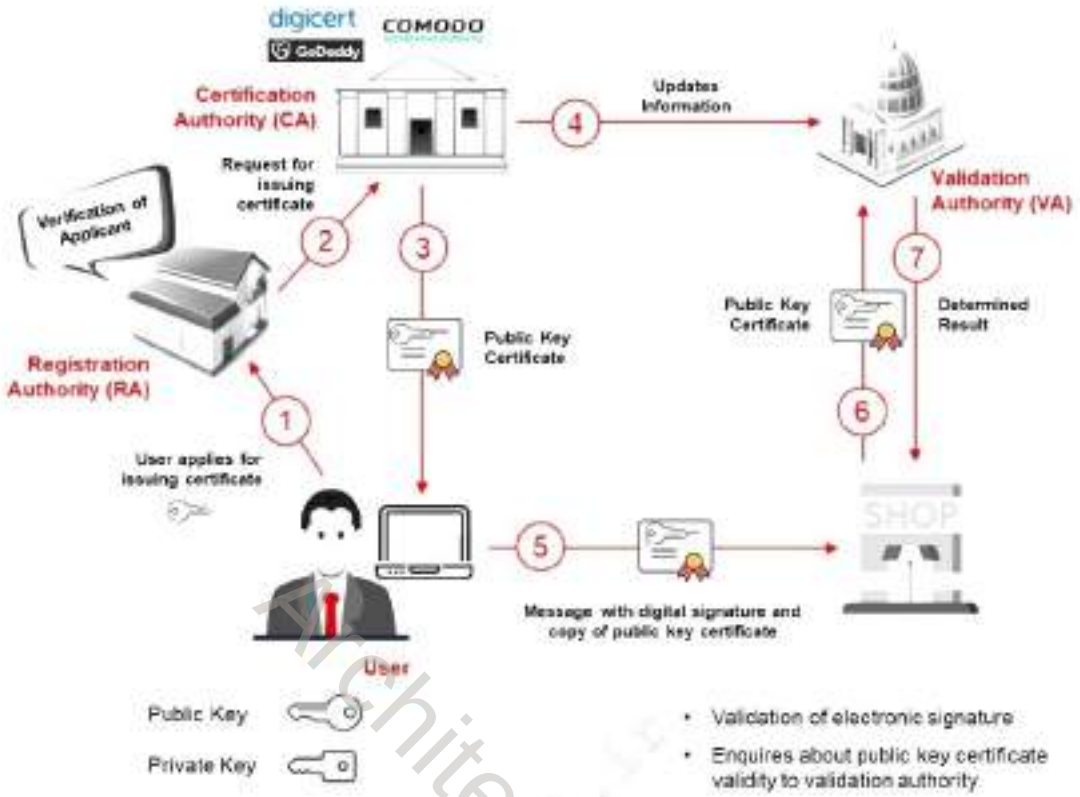


Figure 20.18: Public Key Infrastructure (PKI)

Certification Authorities

Certification authorities (CAs) are trusted entities that issue digital certificates. The digital certificate certifies the possession of the public key by the subject (user, company, or system) specified in the certificate. This aids others to trust signatures or statements made by the private key that is associated with the certified public key.

Some popular CAs are discussed below:

- **Comodo**

  Source: *https://www.comodoca.com*

  Comodo offers a range of PKI digital certificates with strong SSL encryption (128/256 available) with Server-Gated Cryptography (SGC). It ensures standards of confidentiality, system reliability, and pertinent business practices as judged via qualified independent audits. It offers PKI management solutions such as Comodo Certificate Manager and Comodo EPKI Manager.

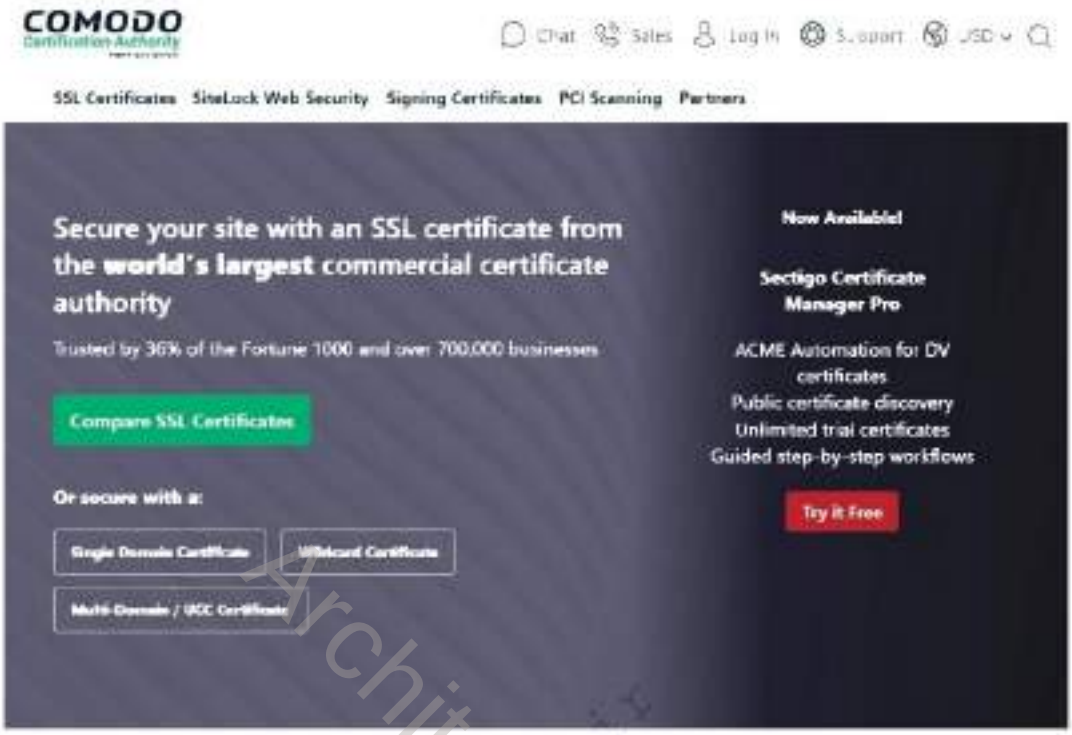


Figure 20.19: Screenshot of Comodo Website

- **IdenTrust**

  Source: *https://www.identrust.com*

  IdenTrust is a trusted third party that provides CA services for many sectors such as banks, corporates, governments, and healthcare. It provides solutions such as digital signing and sealing, compliance with NIST SP 800-171, global identity networks, and managed PKI hosting services.

Figure 20.20: Screenshot of IdenTrust Website

- **DigiCert CertCentral**

  Source: *https://www.digicert.com*

  CertCentral simplifies the entire lifecycle by consolidating tasks for issuing, installing, inspecting, remediating, and renewing TLS/SSL certificates. It manages high-volume TLS/SSL certificate issuance for multiple individuals and teams.

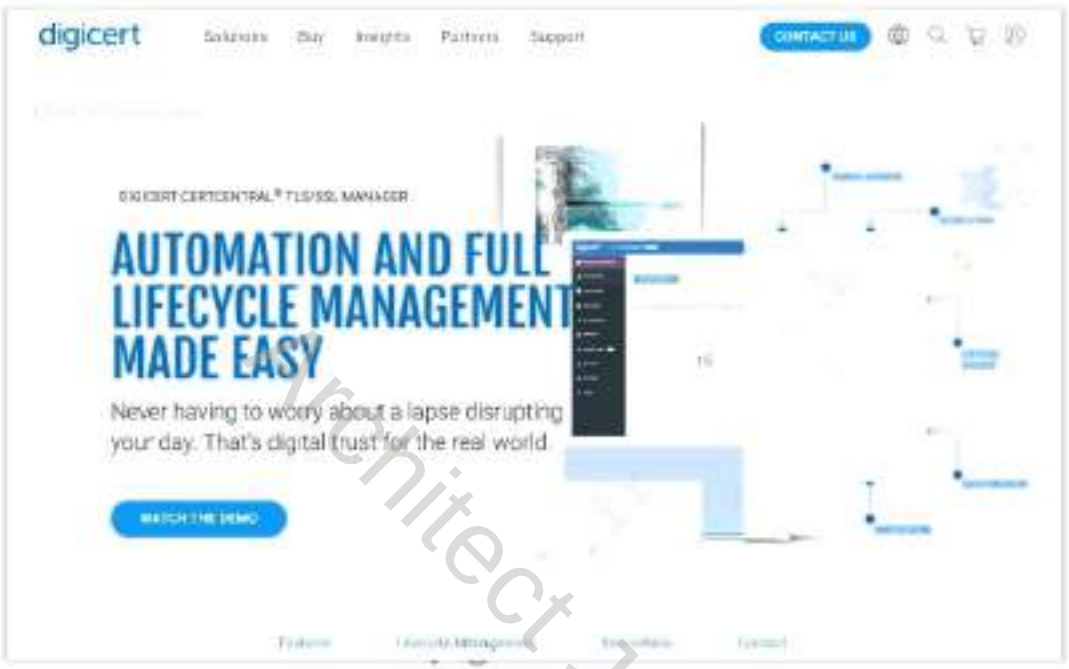


Figure 20.21: Screenshot of DigiCert Website

- **GoDaddy**

  Source: *https://www.godaddy.com*

  GoDaddy SSL Certificates offer a complete range of certificates that comply with CA/Browser Forum guidelines. They provide the SHA-2 hash algorithm and 2048-bit encryption, protection of unlimited servers, etc.

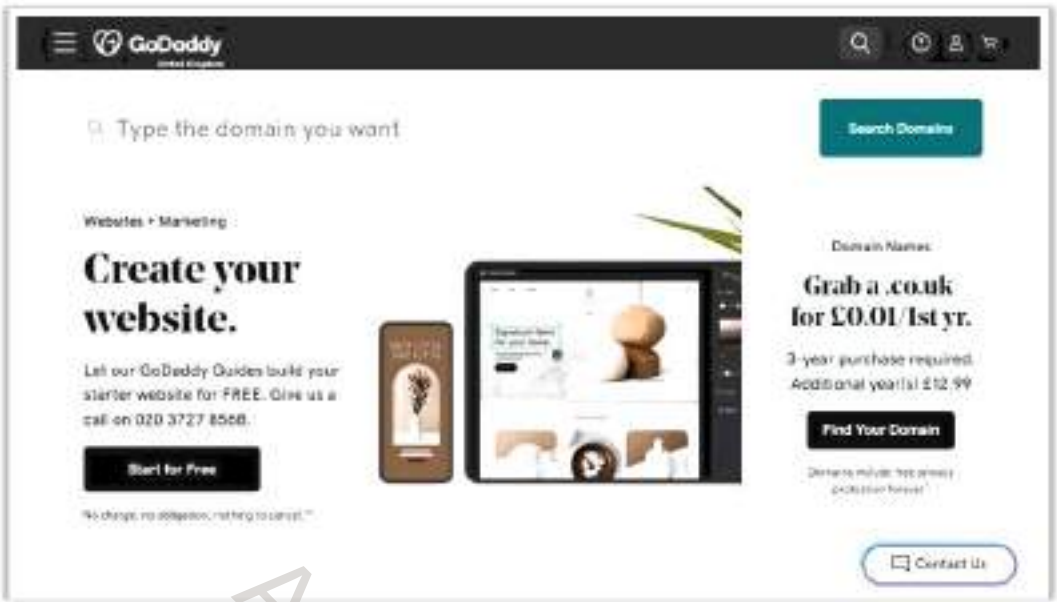
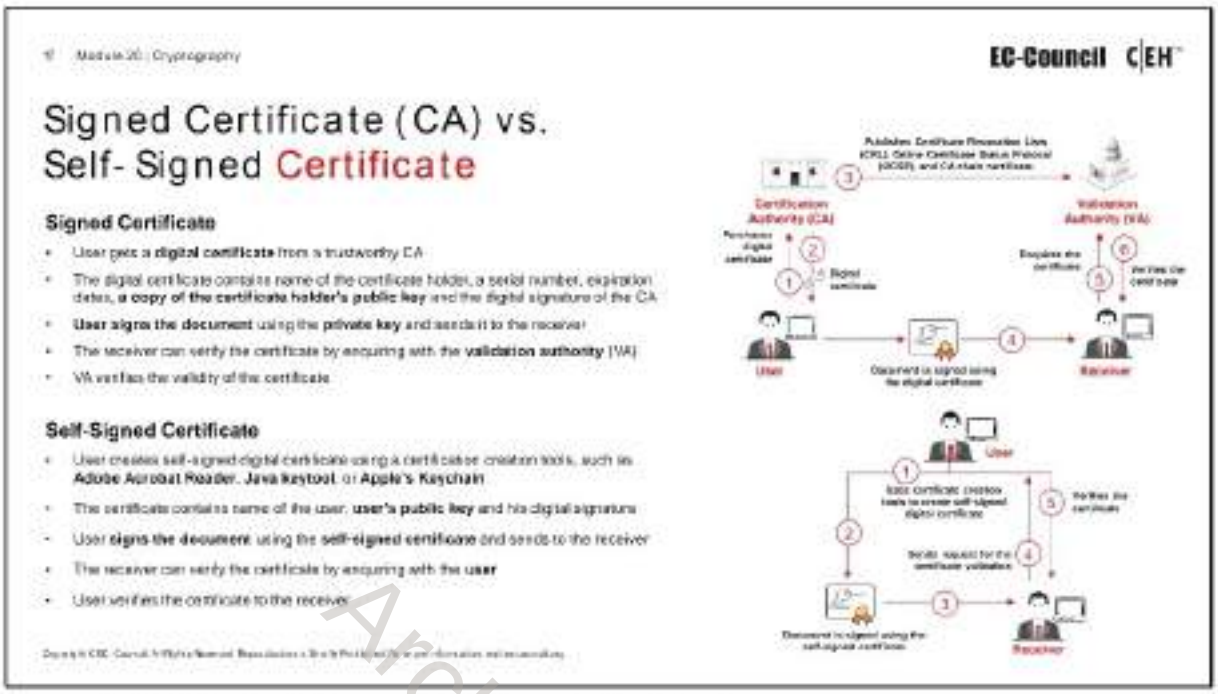


Figure 20.22: Screenshot of GoDaddy Website



Signed Certificate (CA) vs. Self-Signed Certificate

- **Signed Certificate**

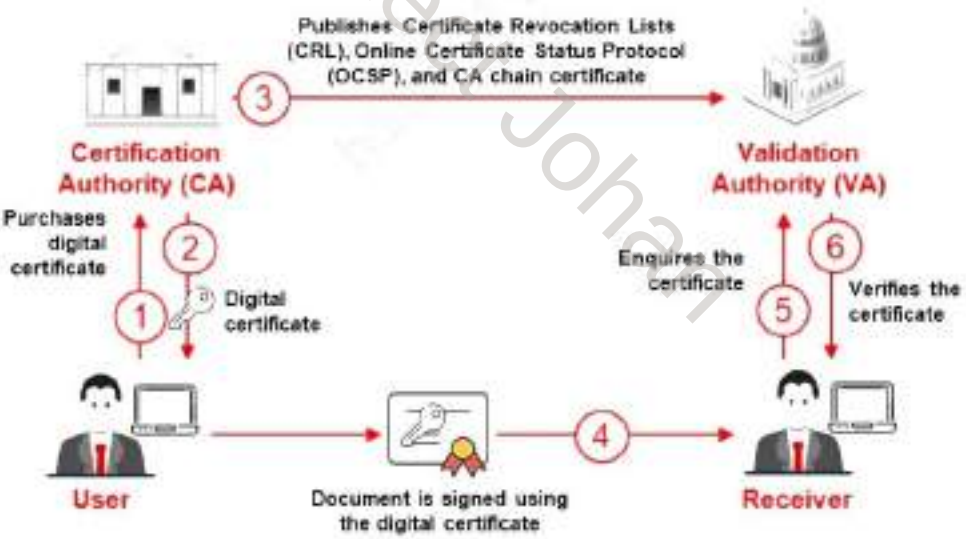


Figure 20.23: Process of obtaining signed certificates

As shown in the diagram above, the user gets a digital certificate from a trustworthy CA. The digital certificate contains name of the certificate holder, a serial number, expiration dates, a copy of the certificate holder's public key and the digital signature of the CA. The user signs the document using the private key and sends to the receiver. The receiver can verify the certificate by enquiring with validation authority (VA). VA verifies the validity of the certificate.

- **Self-Signed Certificate**

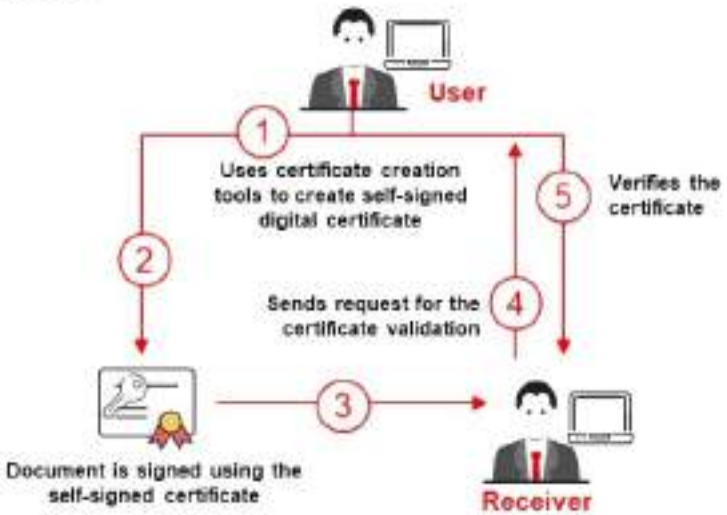
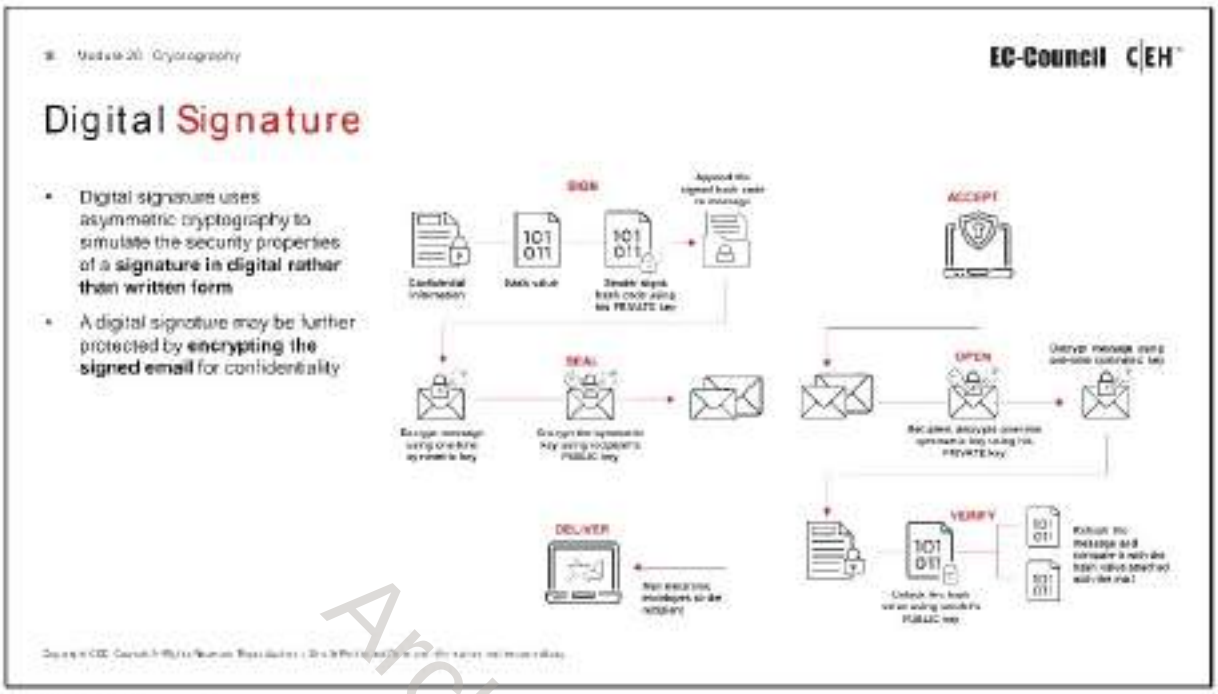


Figure 20.24: Process of generating self-signed certificates

As shown in the diagram above, the user creates self-signed digital certificate using certification creation tools, such as Adobe Acrobat Reader, Java keytool, or Apple's Keychain. The certificate contains name of the user, user's public key and his digital signature. The user signs the document using the self-signed certificate and sends to the receiver. The receiver can verify the certificate by enquiring with the user. The user verifies the certificate to the receiver.



Digital Signature

A digital signature uses asymmetric cryptography to simulate the security properties of a signature in digital form rather than in written form. A digital signature is a cryptographic means of authentication. Public-key cryptography uses asymmetric encryption and helps the user to create a digital signature. The two types of keys in public-key cryptography are the private key (only the signer knows this key and uses it to create a digital signature) and the public key (it is widely known and the relying party uses it to verify the digital signature).

A hash function is an algorithm that helps a user to create and verify a digital signature. This algorithm creates a digital representation, also known as the message fingerprint. This fingerprint has a hash value that is much smaller than the message, but one that is unique to it. If the attacker changes the message, the hash function will automatically produce a different hash value.

To verify the digital signature, one needs the hash value of the original message and the encryption algorithm used to create the digital signature. Using both the public key and the new result, the verifier checks to see if the digital signature was created with the related private key and whether the new hash value is the same as the original one. A digital signature may be further protected by encrypting the signed email for confidentiality.

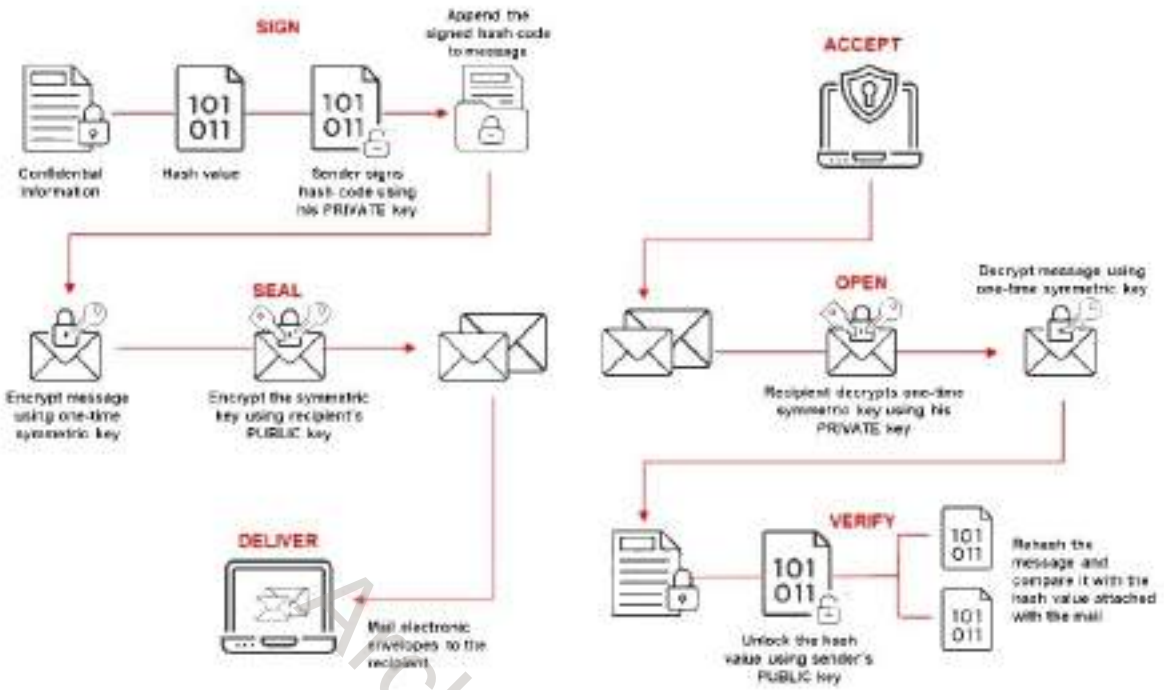
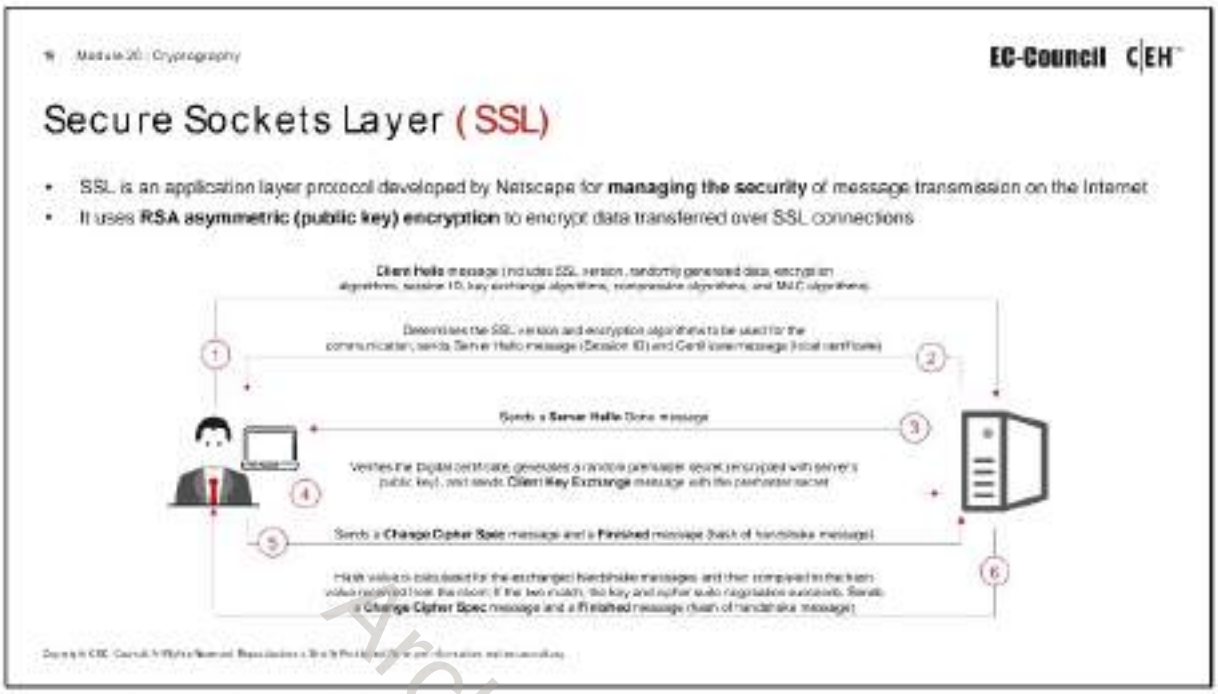


Figure 20.25: Using digital signature for email security



Secure Sockets Layer (SSL)

The Secure Sockets Layer (SSL) protocol is an application layer protocol developed by Netscape for managing the security of message transmission on the Internet. It is used to provide a secure authentication mechanism between two communicating applications, such as a client and a server. SSL requires a reliable transport protocol, such as TCP, for data transmission and reception. It uses RSA asymmetric (public-key) encryption to encrypt data transferred over SSL connections.

Any application-layer protocol that is higher than SSL, such as HTTP, FTP, and telnet, can form a transparent layer over SSL. SSL acts as an arbitrator between the encryption algorithm and the session key; it also verifies the destination server prior to the transmission and reception of data. SSL encrypts the complete data of the application protocol to ensure security.

SSL also offers "**channelsecurity**" with three basic properties:

- **Private channel** – All the messages are encrypted after a simple handshake is used to define a secret key.
- **Authenticated channel** – The server endpoint of the conversation is always encrypted, whereas the client endpoint is optionally authenticated.
- **Reliable channel** – Message transfer has an integrity check.

SSL uses both asymmetric and symmetric authentication mechanisms. Public-key encryption verifies the identities of the server, the client, or both. Once authentication has occurred, the client and server can create symmetric keys, allowing them to communicate and transfer data rapidly. An SSL session is responsible for carrying out the SSL handshake protocol to organize the states of the server and clients, thus ensuring consistency of the protocol.

SSL Handshake Protocol Flow

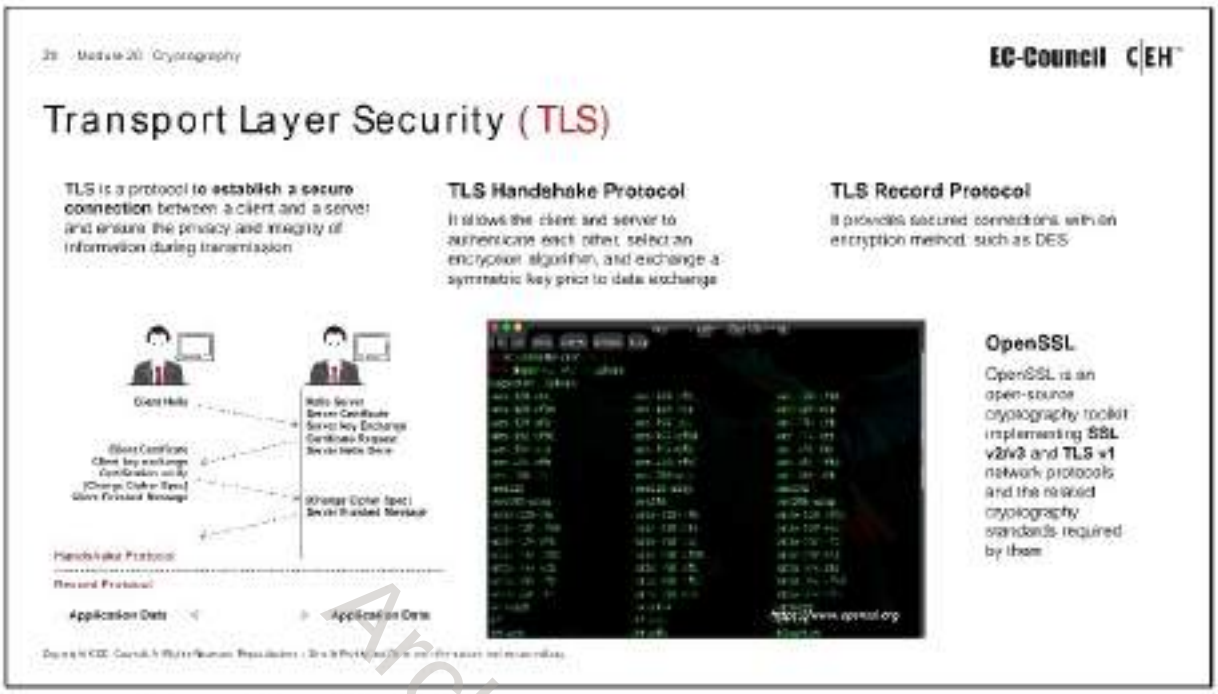
The SSL handshake protocol works on top of the SSL record layer. The processes executed in the three-way handshake protocol are as follows:

1. The client sends a hello message to the server, to which the server must respond with a hello message, or the connection will fail due to the occurrence of a fatal error. The attributes established due to the server and client hello are protocol version, session ID, cipher suite, and compression method.
2. After the connection is established, the server sends a certificate to the client for authentication. In addition, the server might send a server-key exchange message. On authentication of the server, it may ask the client for the certificate (if appropriate for the cipher suite selected).
3. The server sends a "hello done" message to inform the client that the handshake phase is complete and waits for the client's response.
4. If the client receives a certificate-request message, the client must respond to the message by sending a certificate message or "no certificate" alert. The server sends the client key-exchange message. The content of the message depends on the public-key algorithm between the server hello and the client hello. If the certificate sent by the client has signing ability, a digitally signed certificate verifies the message, and the client transmits it.
5. The client transmits the changed cipher-spec message and copies the pending cipher spec into the current cipher spec. The client sends a message to initiate the completion of the message under the new algorithm, keys, and secrets.
6. In response, the server replies by sending its own changed cipher-spec message, transfers the pending cipher spec to the current cipher spec, and initiates the completion of the message under the new cipher spec. At this point, the handshake is complete and the server starts exchanging the application-layer data.

Figure 20.26: SSL handshake protocol flow

The resumption of a previous session or the replication of an existing session proceeds as follows:

- The client initiates the communication by sending a hello message with the session ID of the session that is to be resumed.
- If the server finds a match, it re-establishes the session under the specified session state with the same session ID.
- At this point, both the server and the client exchange the changed spec messages and proceed directly to the finished messages.
- After re-establishment, the server and client exchange data at the application layer.
- If the session ID does not exist, the server creates a new session ID. The SSL client and server then carry out a complete handshake.



Transport Layer Security (TLS)

The Transport Layer Security (TLS) protocol is used to establish a secure connection between a client and a server and ensure the privacy and integrity of information during transmission. It uses a symmetric key for bulk encryption, asymmetric key for authentication and key exchange, and message authentication codes for message integrity. It uses the RSA algorithm with strengths of 1024 and 2048 bits. Using TLS, one can reduce security risks such as message tampering, message forgery, and message interception. An advantage of TLS is that it is independent of the application protocol. Higher-level protocols can lie on top of TLS transparently.

TLS consists of two layers: TLS Record Protocol and TLS Handshake Protocol.

1. **TLS Record Protocol**

   The TLS Record Protocol is a layered protocol. It provides secured connections with an encryption method such as DES. It secures application data using the keys generated during the handshake and verifies its integrity and origin.

   The TLS Record Protocol provides connection security with two basic properties:

   - **The connection is private**: Uses symmetric cryptography for data encryption (e.g., DES). The protocol generates unique keys for symmetric encryption for each connection, depending on a secret negotiated by another protocol (such as the TLS Handshake Protocol). One can use the TLS Record Protocol without encryption.
   - **The connection is reliable**: It provides a message integrity check at the time of message transport using a keyed MAC. Secure hash functions (e.g., SHA, MD5) help to perform MAC computations.

The TLS Record Protocol does the following:

- Fragments outgoing data into manageable blocks and reassembles incoming data
- Optionally compresses outgoing data and decompresses incoming data
- Applies MAC to the outgoing data and uses MAC to verify the incoming data
- Encrypts outgoing data and decrypts incoming data

The TLS Record Protocol sends the outgoing encrypted data to the TCP layer for transport.

2. **TLS Handshake Protocol**

The TLS Handshake Protocol allows the client and server to authenticate each other and select an encryption algorithm and cryptographic keys prior to data exchange by the application protocol.

It provides connection security with three basic properties:

- The peer's identity can be authenticated using asymmetric cryptography. This can be made optional but is mostly required for at least one of the peers.
- The negotiation of a shared secret is secure.
- The negotiation is reliable.

The TLS Handshake Protocol operates on top of the TLS Record Protocol and is responsible for producing cryptographic parameters of the session state. At the start of communication, the TLS client and server agree on a protocol version, select cryptographic algorithms, optionally authenticate each other, and use asymmetric cryptography techniques to create shared secrets.

The steps involved in the TLS Handshake Protocol are as follows:

- Initially, the client sends a "Client hello" message accompanied by the client's random value and supported cipher suites to the server.
- The server responds to the client by sending a "Server hello" message accompanied by the server's random value.
- The server sends its certificate to the client for authentication and may request the client's certificate. The server sends the "Server hello done" message.
- The client sends its certificate to the server, if requested.
- The client generates a random pre-master secret and encrypts it with the server's public key; then, it sends the encrypted pre-master secret to the server.
- The server receives the pre-master secret. Thereafter, the client and server each create the master secret and session keys based on the pre-master secret.
- The client sends "Change cipher spec" to the server to indicate that it will start using the new session keys for hashing and encrypting messages. The client also sends "Clientfinished".

- The server receives "Changecipher spec" from the client and switches its record layer security state to symmetric encryption using the session keys. Then, the server sends "Serverfinished" to the client.
- Now, the client and server can exchange application data over the secure channel they have established, and all the messages exchanged between the client and server are encrypted using a session key.

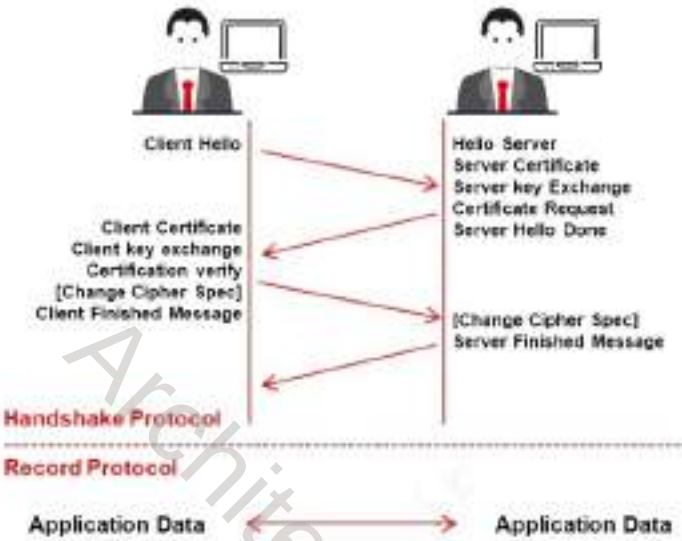


Figure 20.27: TLS handshake and record protocols

Cryptography Toolkits

Cryptography toolkits include cryptographic primitives, algorithms, and schemes used to provide security for various applications. Some cryptography toolkits are discussed below:

- **OpenSSL**

  Source: *https://www.openssl.org*

  OpenSSL is an open-source cryptography toolkit implementing the SSL and TLS network protocols and the related cryptography standards required by them. It is a command-line tool for using the various cryptography functions of OpenSSL's crypto-library from the shell. OpenSSL can be used for the creation and management of private keys, public keys, and parameters; public-key cryptographic operations; creation of X.509 certificates, CSRs, and CRLs; etc.

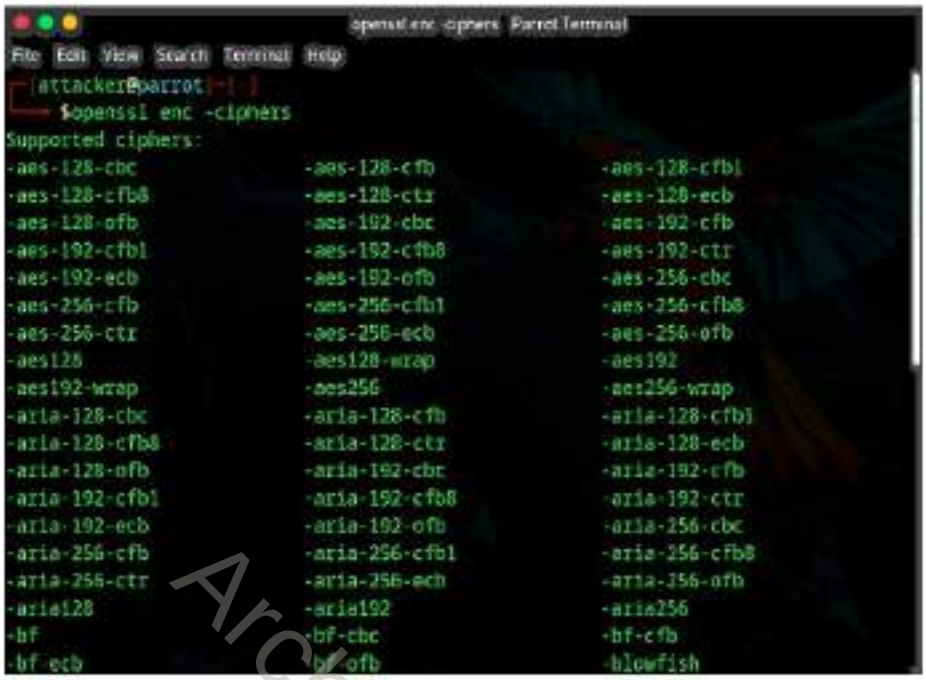
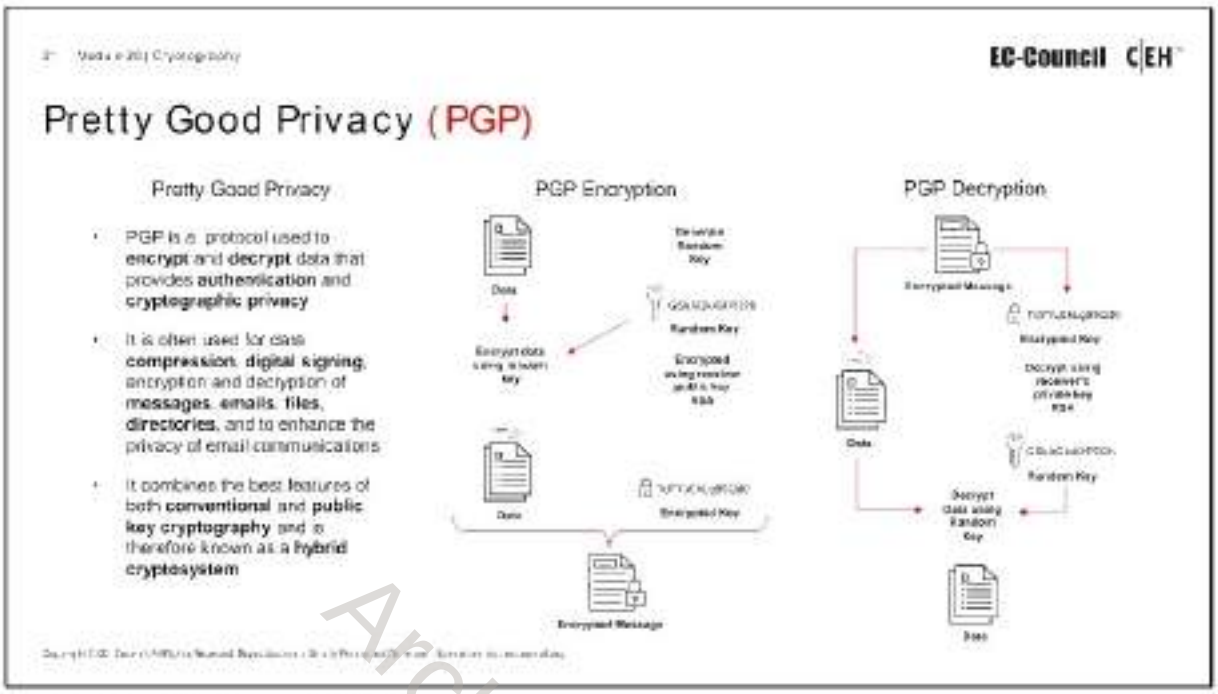


Figure 20.28: Screenshot of the OpenSSL command-line tool

Some additional cryptography toolkits are as follows:

- wolfSSL (*https://www.wolfssl.com*)
- AES Crypto Toolkit (*https://www.ni.com*)
- Libsodium (*https://github.com*)
- Crypto++ (*https://cryptopp.com*)
- PyCryptodome (*https://github.com*)



Pretty Good Privacy (PGP)

Pretty Good Privacy (PGP) is a protocol used to encrypt and decrypt data with authentication and cryptographic privacy. It is often used for data compression, digital signing, encryption and decryption of messages, emails, files, and directories, and to enhance the privacy of email communications. The algorithm used for message encryption is RSA for key transport and IDEA for bulk-message encryption. PGP uses RSA for computing digital signatures and MD5 for computing message digests.

It combines the best features of both conventional (around 1,000 times faster than public-key encryption) and public-key cryptography (solution to key distribution and data transmission issues) and is thereby, known as a hybrid cryptosystem.

PGP is used for:

- Encrypting a message or file prior to transmission so that only the recipient can decrypt and read it
- Clear signing of the plaintext message to ensure the authenticity of the sender
- Encrypting stored computer files so that no one besides the person who encrypted them can decrypt them
- Deleting files rather than just removing them from the directory or folder
- Data compression for storage or transmission

How PGP Works?

- **PGP Encryption**
  - When a user encrypts data with PGP, PGP first compresses the data.

    Compressing the data reduces patterns in the plaintext that could be exploited by most cryptanalysis techniques to crack the cipher, thereby increasing the resistance to cryptanalysis considerably.
  - PGP then creates a random key (GSkAQk49fPD2h) that is a one-time-only secret key.
  - PGP uses the random key generated to encrypt the plaintext, resulting in a ciphertext.
  - Once the data is encrypted, a random key is encrypted with the recipient's public key.
  - The public-key-encrypted random key (Td7YuEkLg99Qd0) is sent along with the ciphertext to the recipient.

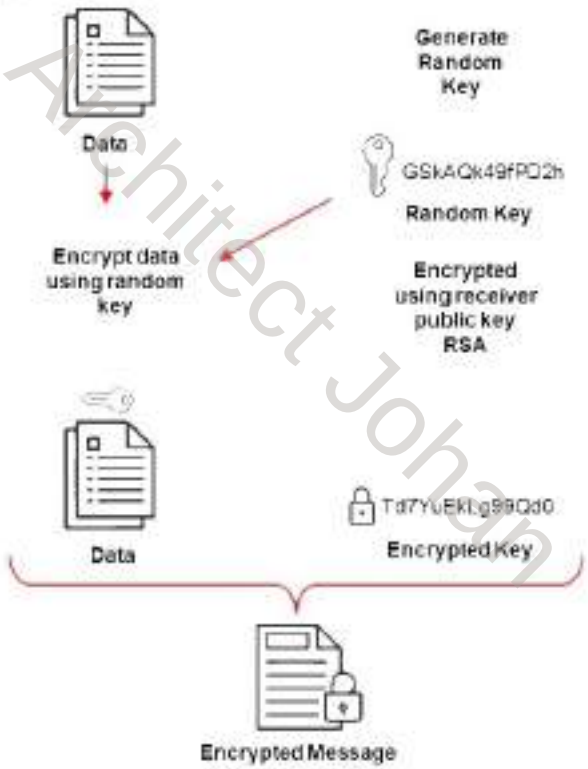


Figure 20.29: PGP Encryption

- **PGP Decryption**
  - Decryption works in reverse.
  - The recipient's copy of PGP uses his or her private key instead of the public key to recover the temporary random key.
  - PGP then uses the recovered random key to decrypt the conventionally encrypted ciphertext.

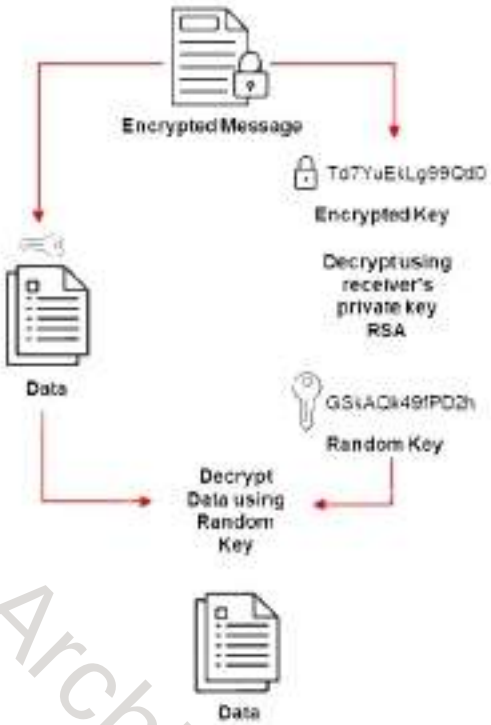
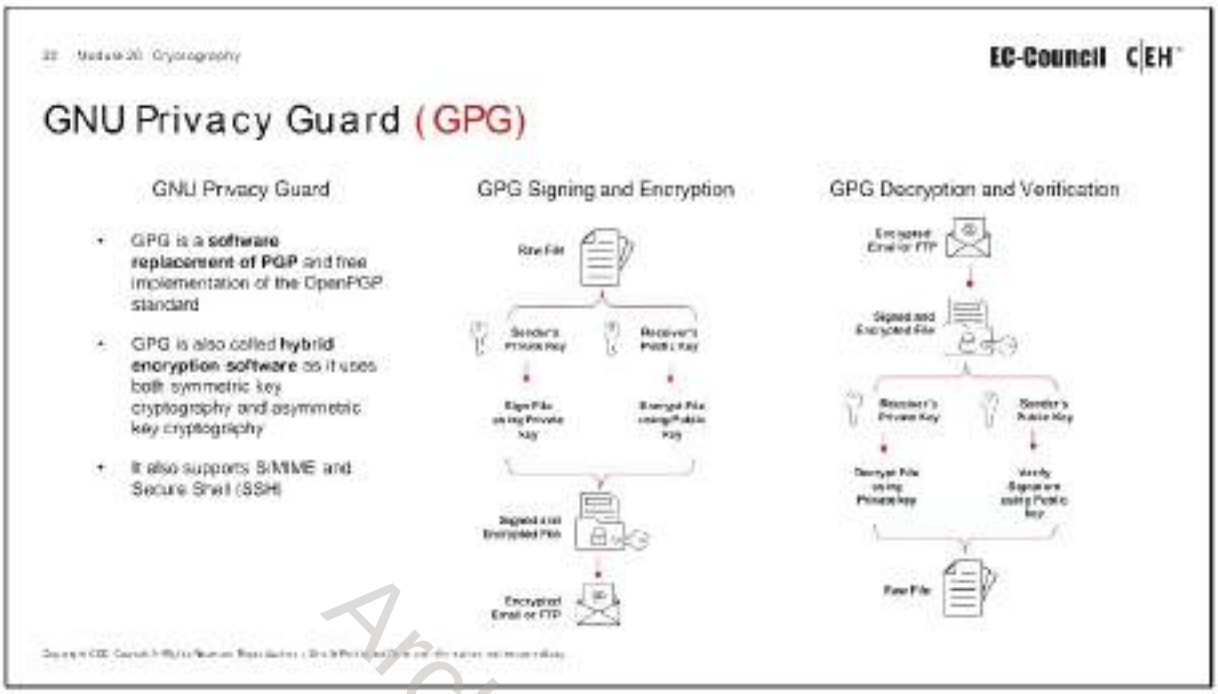


Figure 20.30: PGP Decryption

Note: Each step of the PGP encryption process (hashing, data compression, symmetric-key cryptography, and public-key cryptography) uses one of the various supported algorithms.



GNU Privacy Guard (GPG)

GNU Privacy Guard (GPG) is a software replacement of PGP and free implementation of the OpenPGP standard that is used to encrypt and decrypt data. GPG is also called a hybrid encryption software program, as it uses both symmetric-key cryptography and asymmetric-key cryptography for improved speed and secure key exchange, which is achieved using the receiver's public key for encrypting the session key.

GPG also supports S/MIME and Secure Shell (SSH). The latest version of GPG supports most cryptographic functions such as elliptic curve cryptography (ECDSA, ECDH, and EdDSA), and it also supports the cryptography library Libgcrypt.

GPG is used for the following:

- Proper key management of both private and public keys
- Creating new private keys and exporting or importing any key even though it is in some armored (e.g., ASCII) format
- Pushing the public key to the key server by signing code with the GPG key having a public signature
- Deleting a private key from local storage
- Encrypting and signing files using asymmetric keys for encrypting any file used for email or FTP
- Decrypting and verifying the encrypted file using asymmetric keys
- Detaching signatures where the signature file can be detached from the message file
- Managing and building the web of trust
- Automatically securing messages in messaging applications such as Psi and Fire

How GPG Works

- **GPG Encryption**
  - GPG encrypts messages individually by using asymmetric-key pairs.
  - The user sends the raw file, and GPG is used for signing the file using the sender's private key for confirming the file content at the time of signing.
  - Then, the file is encrypted using the receiver's public key. Now, the file can be decrypted only with the receiver's private key.
  - After encrypting the data, the encrypted file can be stored locally, distributed to the FTP servers, or sent to email recipients.

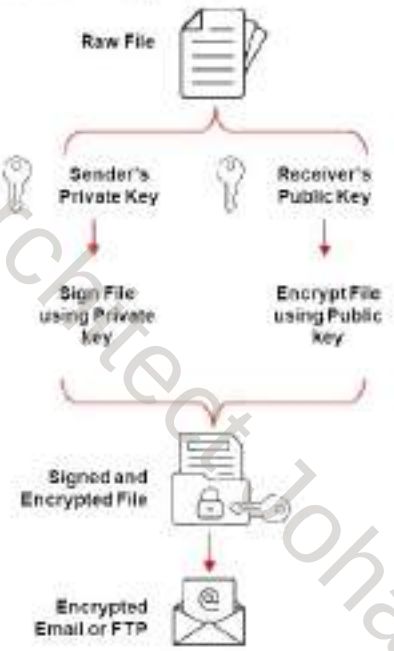
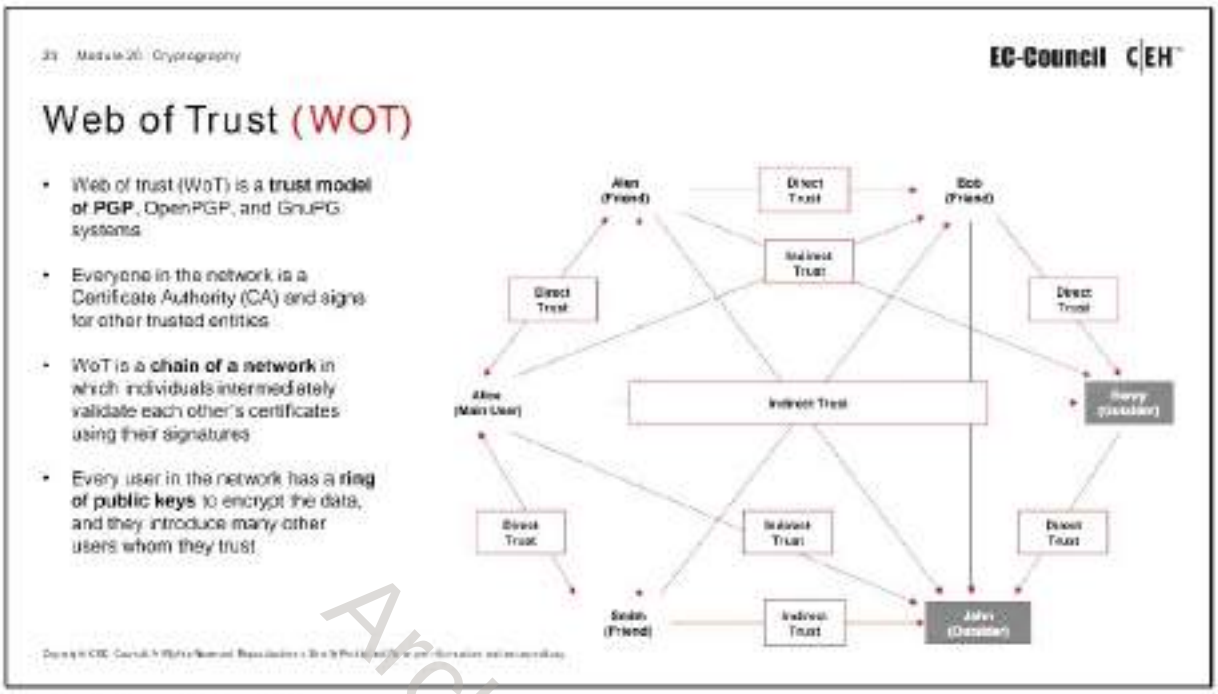


Figure 20.31: GPG Signing and Encryption

- **GPG Decryption**
  - GPG decryption is the reverse process of GPG encryption.
  - As the asymmetric-key pairs are used, GPG searches for the receiver's private key for decrypting the file.
  - Signature verification is done automatically by the GPG using the sender's public key after the decryption.

Figure 20.32: GPG Decryption and Verification



Web of Trust (WOT)

Web of trust (WoT) is a trust model of PGP, OpenPGP, and GnuPG accessible systems. It is an idea of decentralizing the key distribution among PGP users. In the PKI, only centralized power such as the CA signs certificates in the network, ensuring authenticity between the public key and its owner. In WoT, everyone in the network is a CA, and they can sign for other trusted entities. WoT is a network chain in which individuals intermediately validate each other's certificates using their signatures. These signatures verify the ownership of keys from various trust levels. There is a bunch of similar trust levels through direct or indirect references in WoT.

Working of WOT

In WOT, every PGP user in the network has a ring of public keys to encrypt the data, and they introduce many other users whom they trust. In this trust model, a user encodes the data with the receiver's public key that is decrypted only by the receiver's private key. Then, every user in this model digitally signs the data with their private keys; when the recipient is validating it against the user's public key, he/she can confirm the user's authenticity. This process will ensure that data are received from a valid user without being modified, and only the intended user can access the information, as only he/she holds the related private key.

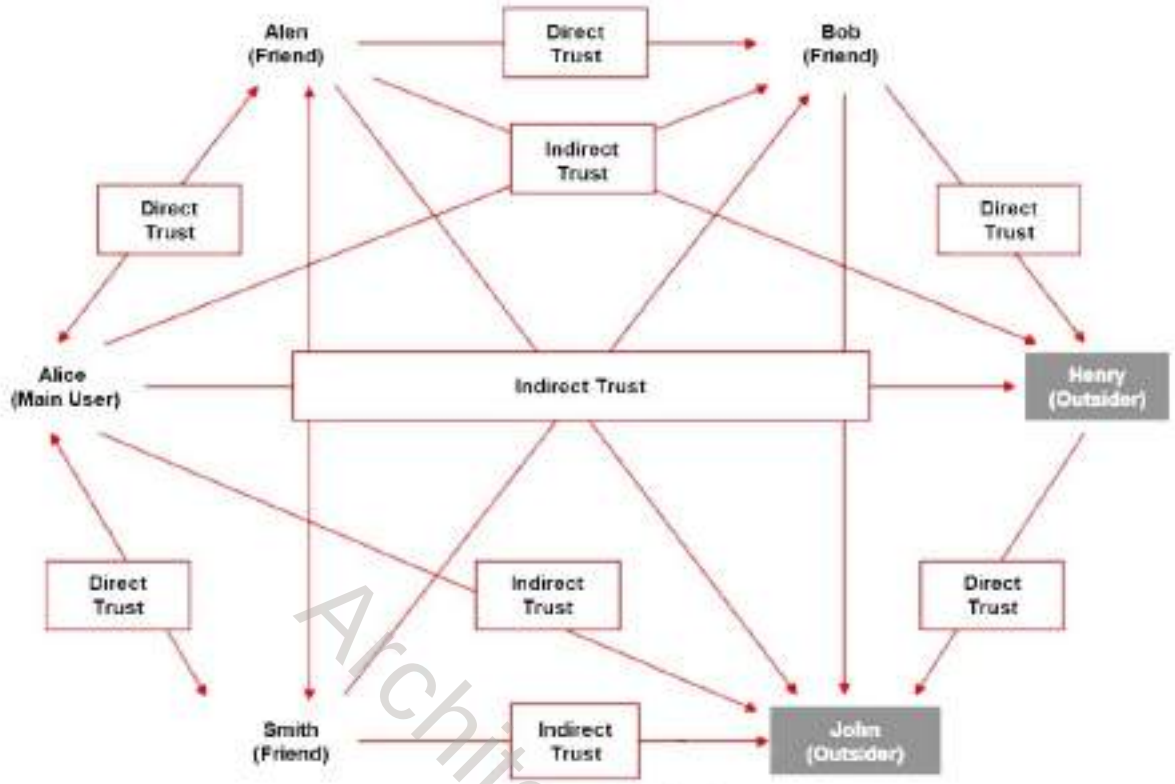
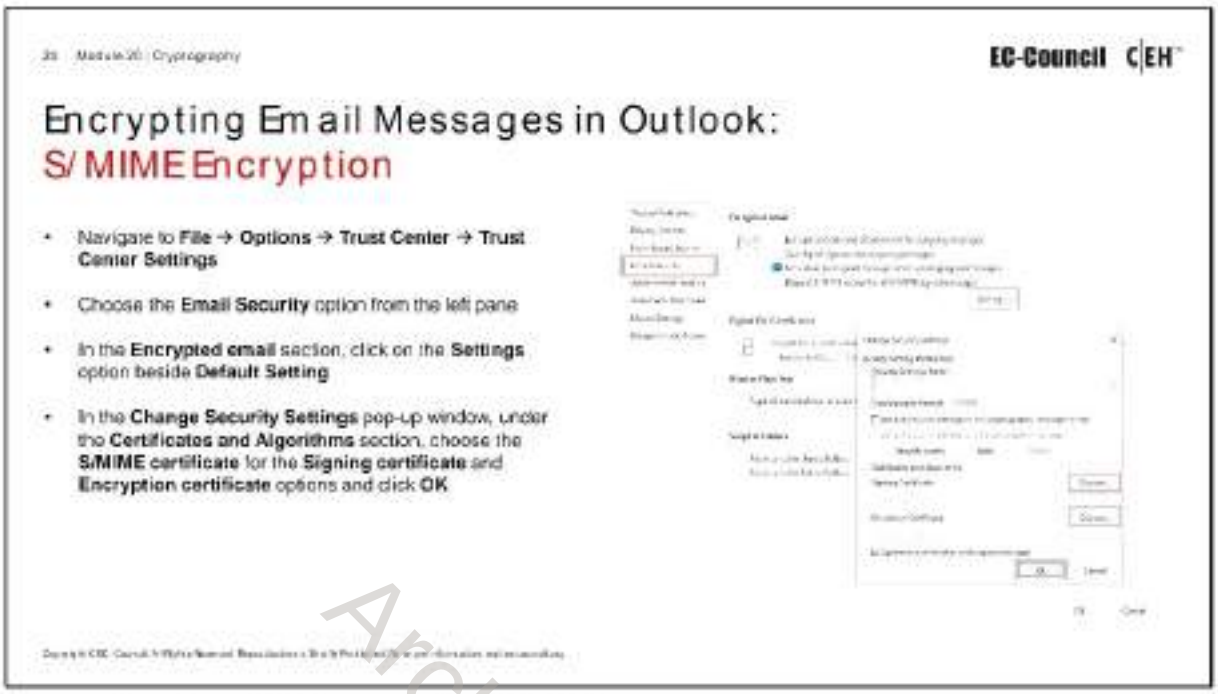


Figure 20.33: Working of WOT



Encrypting Email Messages in Outlook

Secure/Multipurpose Internet Mail Extensions (S/MIME) Encryption

S/MIME certification is a technique that allows users to encrypt their email messages. It is used for encrypting Outlook email messages so that senders and the designated receivers can access them without compromising the integrity of the message.

The steps to encrypt email messages using S/MIME encryption are discussed below.

To perform these steps, a signing certificate should be attached to the keychain.

- Select **File → Options → Trust Center → Trust Center Settings**.

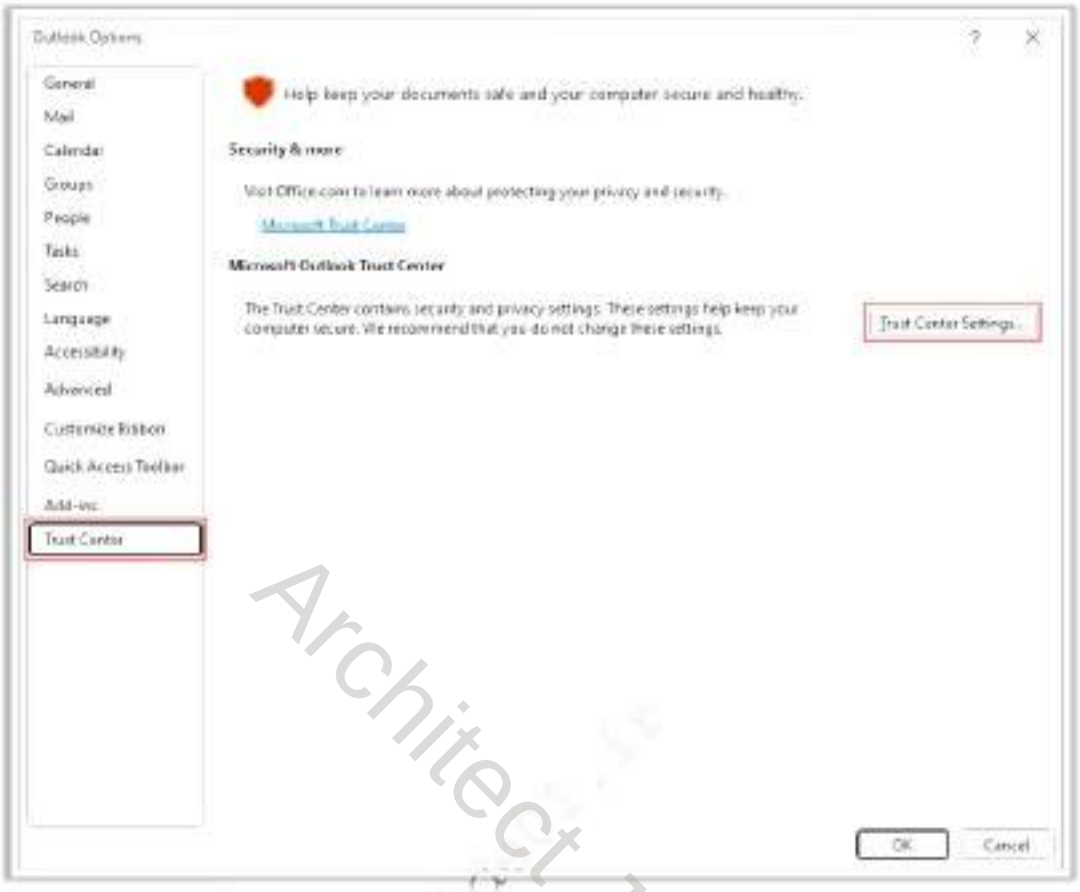


Figure 20.34: Screenshot showing Outlook Trust Center settings

- Choose the **Email Security** option from the left pane.
- In the **Encrypted email** section, click on the **Settings** option beside **Default Setting**.
- In the **Change Security Settings** pop-up window, under the **Certificates and Algorithms** section, choose the **S/MIME certificate** for the **Signing certificate** and **Encryption certificate** options and click **OK**.

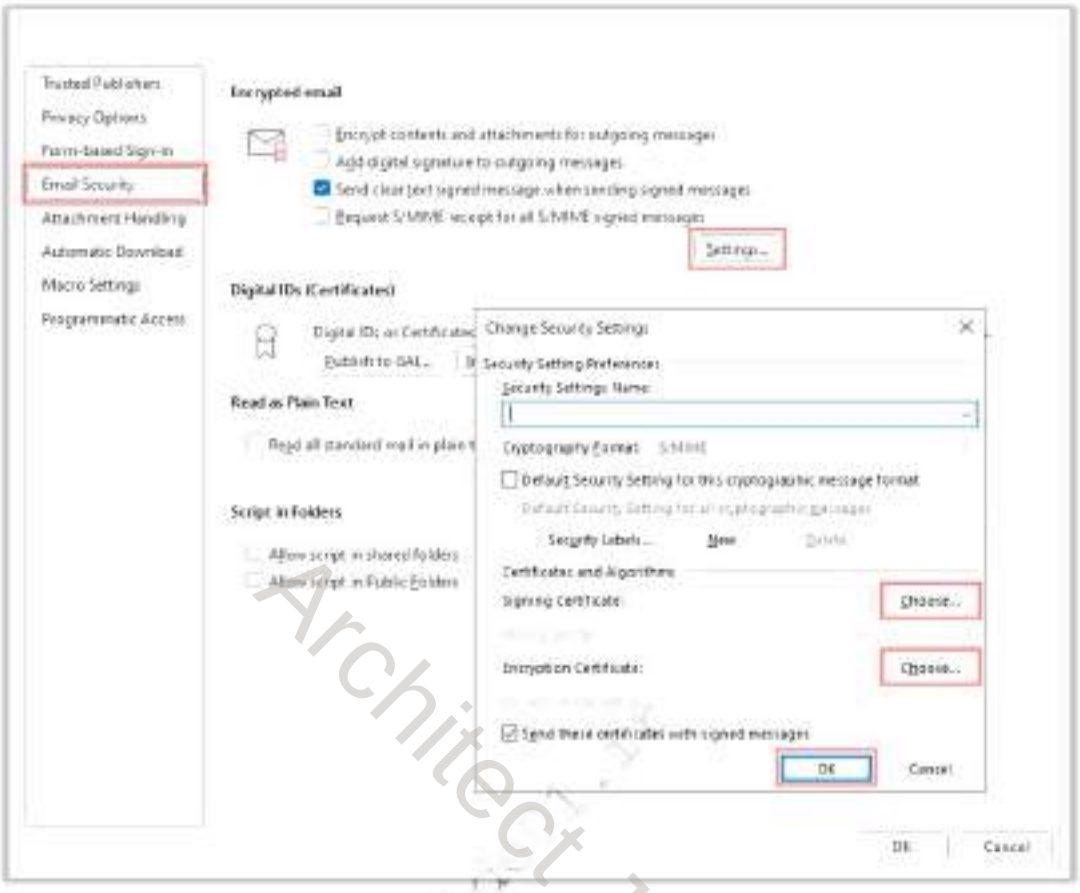


Figure 20.35: Screenshot of Outlook Email Security settings

Note: In this case, the Outlook version is already using S/MIME.

Microsoft 365 Message Encryption

Office 365 Message Encryption (OME) allows users to send an encrypted email message to any email address. OME requires receivers to login to a Microsoft Office 365 account or use a one-time password to authenticate themselves.

The steps to encrypt email messages using OME encryption are as follows.

- In an email message body, select the **Options** menu, go to **Encrypt**, and choose the encryption that includes the required constraints such as **Encrypt-Only** or **Do Not Forward**.

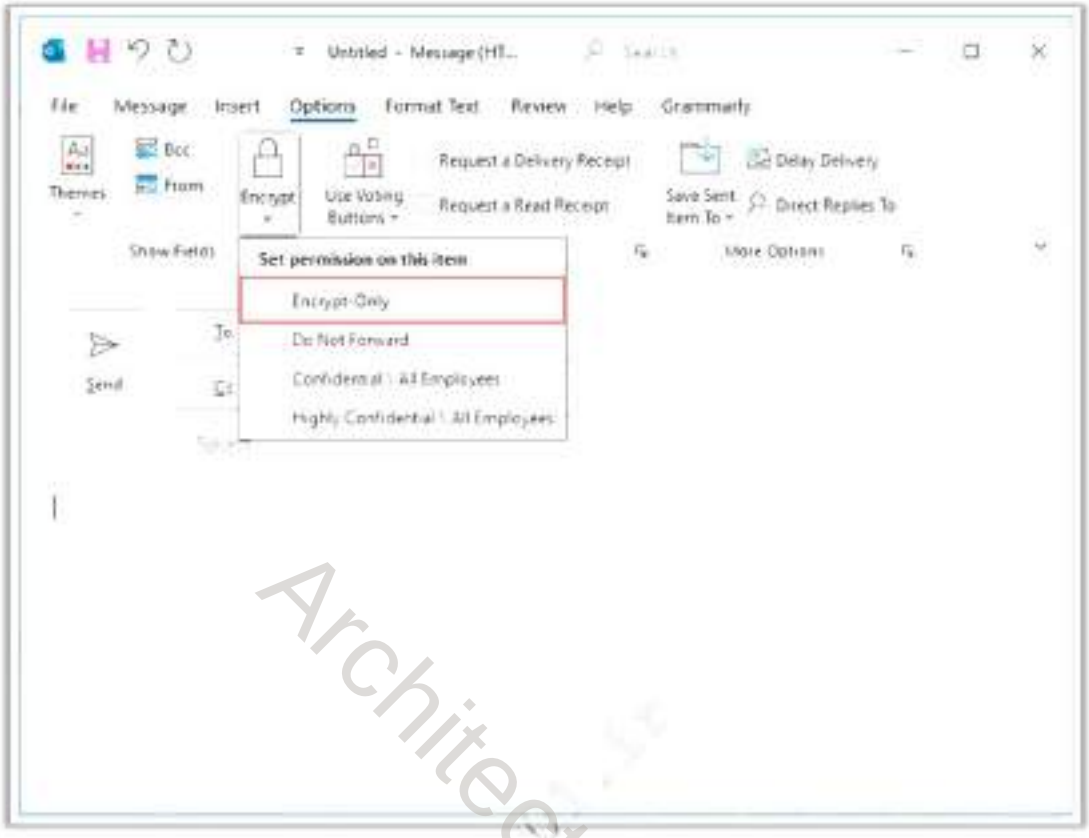


Figure 20.36: Screenshot of Outlook showing Encrypt options

Follow the steps below to encrypt a single message:

- Click **File** and then **Properties** in the email message body.
- In the **Properties** window, click on the **Security Settings** button in the **Security** section.
- In the **Security Properties** pop-up window, check the **Encrypt message contents and attachments** option and click **OK.**

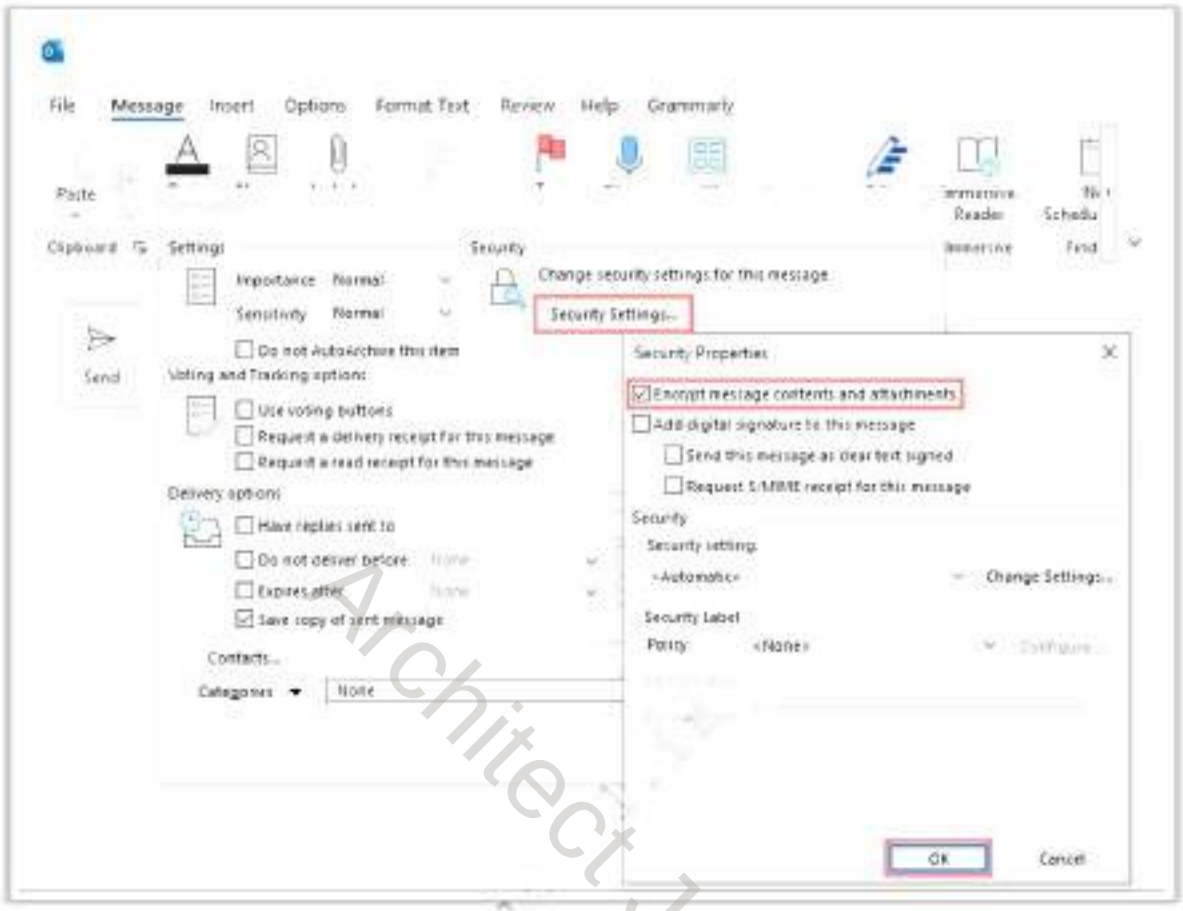


Figure 20.37: Screenshot of Outlook showing Security Properties

Follow the steps below to encrypt all outgoing messages:

- Select **File → Options → Trust Center → Trust Center Settings**.
- Choose the **Email Security** option from the left pane.
- In the **Encrypted email** section, check the **Encrypt contents and attachments for all outgoing messages** option and click **OK**.

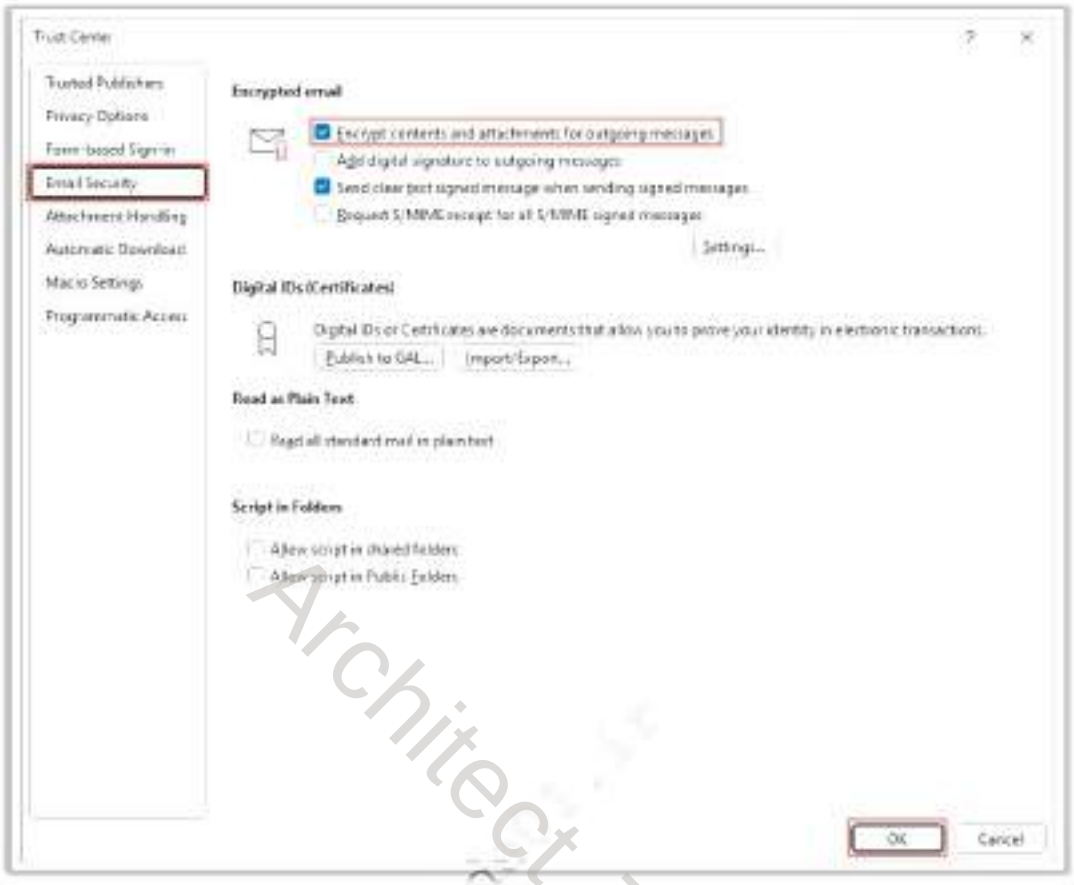


Figure 20.38: Screenshot of Outlook showing Email Security options

Signing/Encrypting Email Messages on Mac

Email security factors in Apple mail can be enhanced by deploying the encryption features called digital signature and message encryption provided by Apple mail service providers. A user is allowed to send a digitally signed and encrypted email from an Apple device.

Sending Digitally Signed and Encrypted Emails

- Access Apple mail ✉ on the Mac device and click on **File → New Message**.
- Place the cursor on the **From** field and then select the account holding the **personal certificate** in the keychain from the **pop-up menu**.

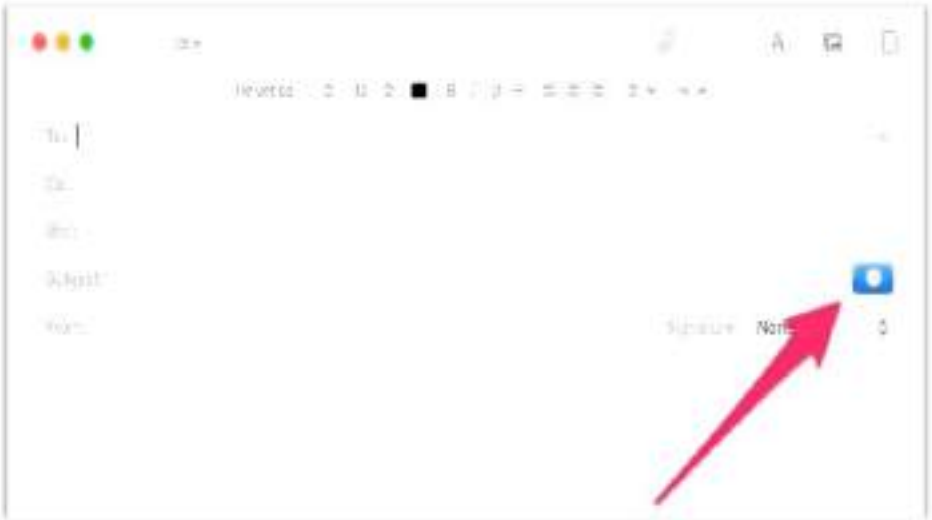


Figure 20.39: Screenshot of Secure Mailbox on Mac

Note: Upon enabling a personal certificate in the keychain, a blue tick (✓) is displayed in the mail, showing that an outgoing message can be digitally signed.

Receiving Digitally Signed and Encrypted Emails

- Email received with a signed tick icon (✓) indicates that the message is digitally signed by a legitimate sender. The recipient can access the sender's digital certificate by **clicking** on the ✓ icon.

  Any warning message appears indicates that the original email's data were tampered with and that the authenticator's identity cannot be identified.

- Email received with a closed lock icon (🔒) indicates that the message is encrypted. The recipient can access that email only by providing a valid private key for decryption.

Encrypting/Decrypting Email Messages Using OpenPGP

Although PGP has a good security factor, it is vulnerable to online attacks. Email with active OpenPGP (a hybrid of PGP) deployed on multiple environments such as Windows, macOS, Android, iOS, Linux, and browser plugins provides a stronger security factor. An email with active OpenPGP installed and configured with compatible browser extensions can enhance the security factors in the browser environment.

Users can use browser extension tools such as FlowCrypt, which can be configured along with OpenPGP, for secure email communication.

FlowCrypt (Gmail)

Source: *https://flowcrypt.com*

FlowCrypt is end-to-end email encryption software configured with OpenPGP for securing emails and attachments in Google mail (G Suite/Business/Enterprise). It allows the encryption/decryption of outgoing/incoming emails on user devices with private/public keys for accessing data.

Follow the steps below for sending and retrieving an encrypted mail (Gmail) using the FlowCrypt browser extension on Chrome or Firefox with OpenPGP:

- Open the browser and enable Open PGP. Install and configure the FlowCrypt browser extension.

At the Sender's End

- Log in to the Gmail account from *https://mail.google.com* using the same browser.

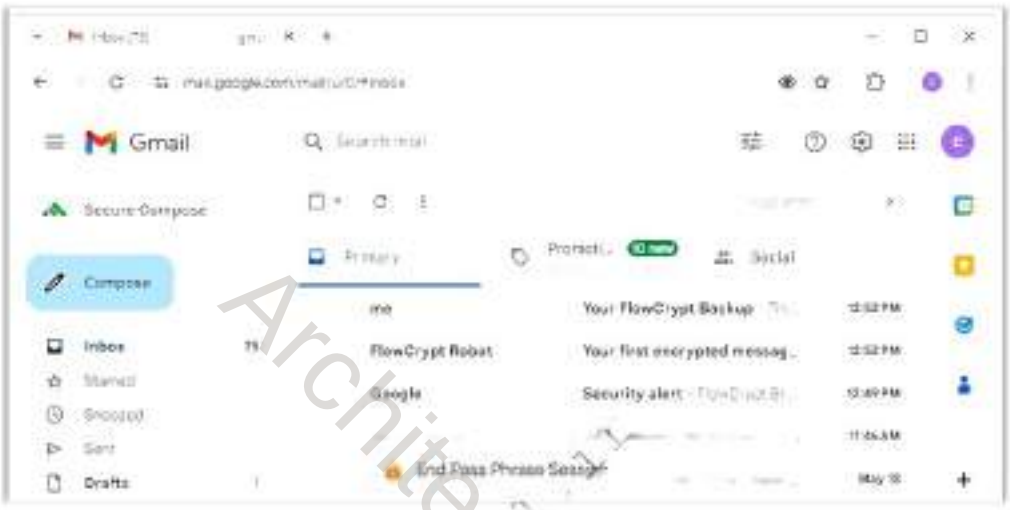


Figure 20.40: Screenshot of a Gmail inbox on a browser with FlowCrypt (OpenPGP)

- Select the **Secure Compose** icon in the left pane. A **New Secure Message** compose mailbox pops up, as shown in the screenshot below:

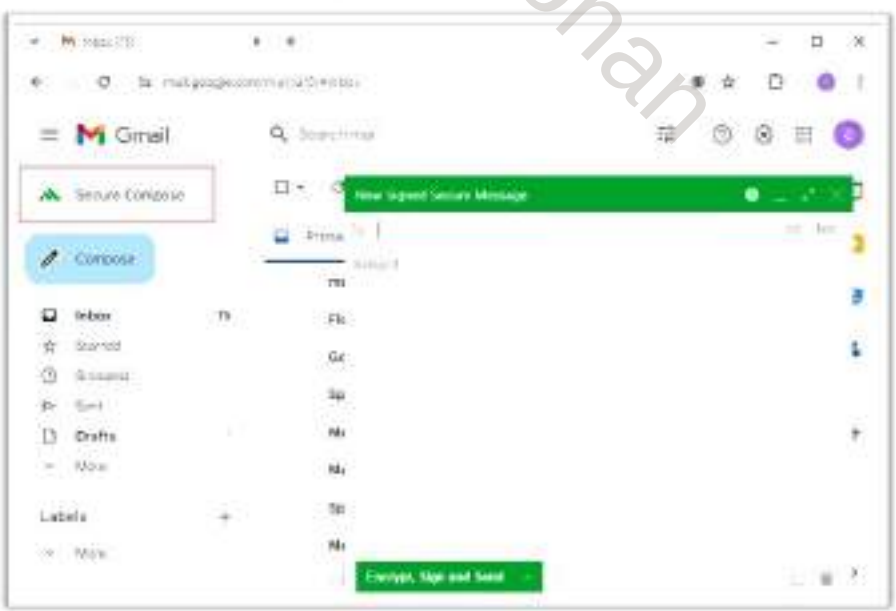


Figure 20.41: Screenshot of a New Secure Message pop-up

- Enter the recipient's email address in the **Add Recipient** field (**New Secure Message**).

Figure 20.42: Screenshot showing the addition of a recipient name

Note: If the recipient's name is displayed in green in the compose mailbox, the recipient also accesses their browser with FlowCrypt installed.

- Add a **Subject** and **Body** (add attachments, if any) for the email.

Figure 20.43: Screenshot showing the composed email

- Send the mail to the recipient address by clicking on **Encrypt, Sign and Send**.

At the Recipient End

- The recipient can access the email by clicking on the newly received email.

Figure 20.44: Screenshot of a received email in FlowCrypt

Note: Based on settings, the user can enter PGP keys for the decryption of emails.

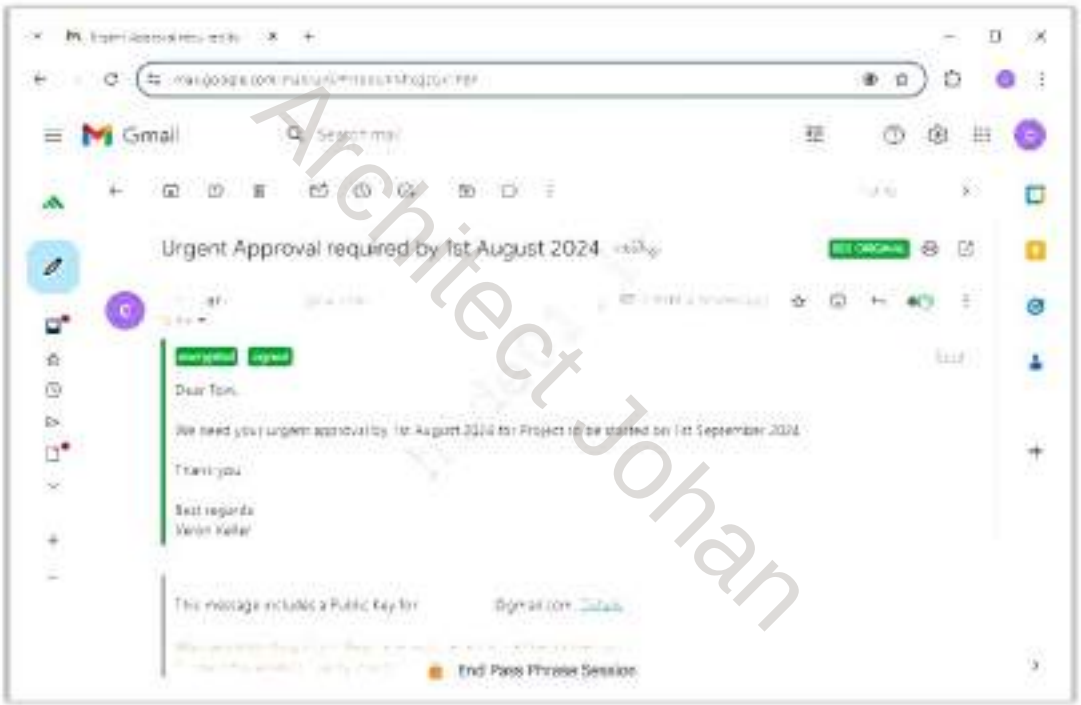
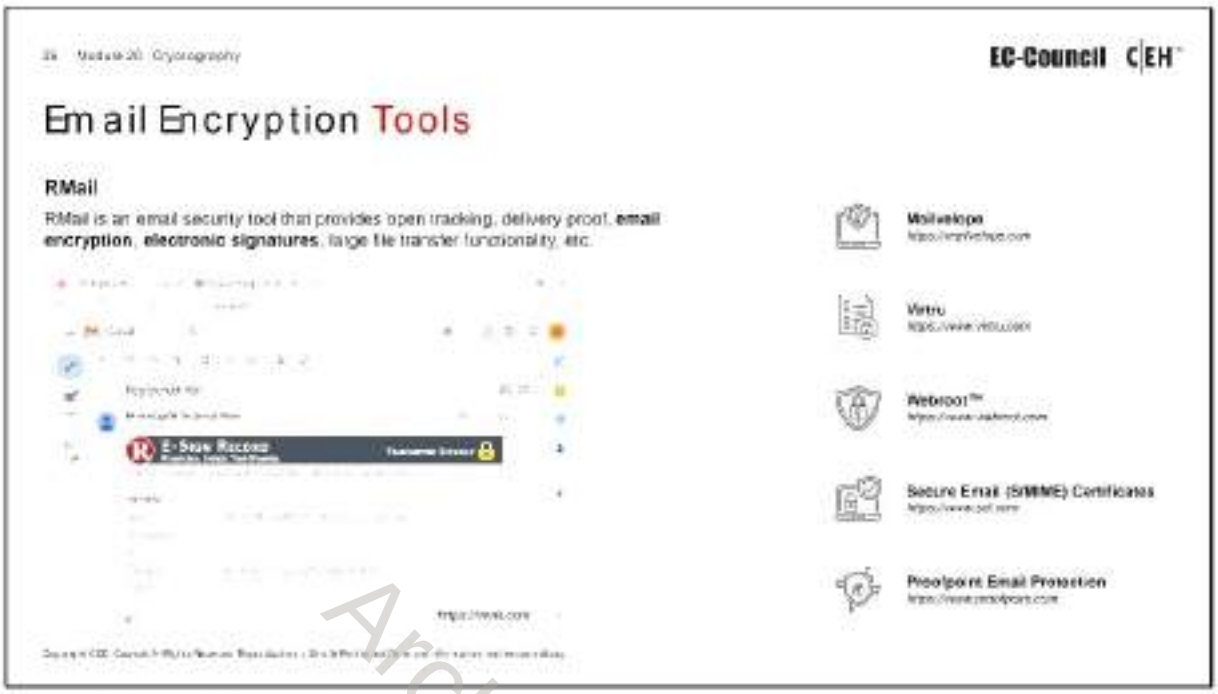


Figure 20.45: Screenshot showing an email accessed using FlowCrypt



Email Encryption Tools

Some important email encryption tools used to secure email messages are as follows:

- **RMail**

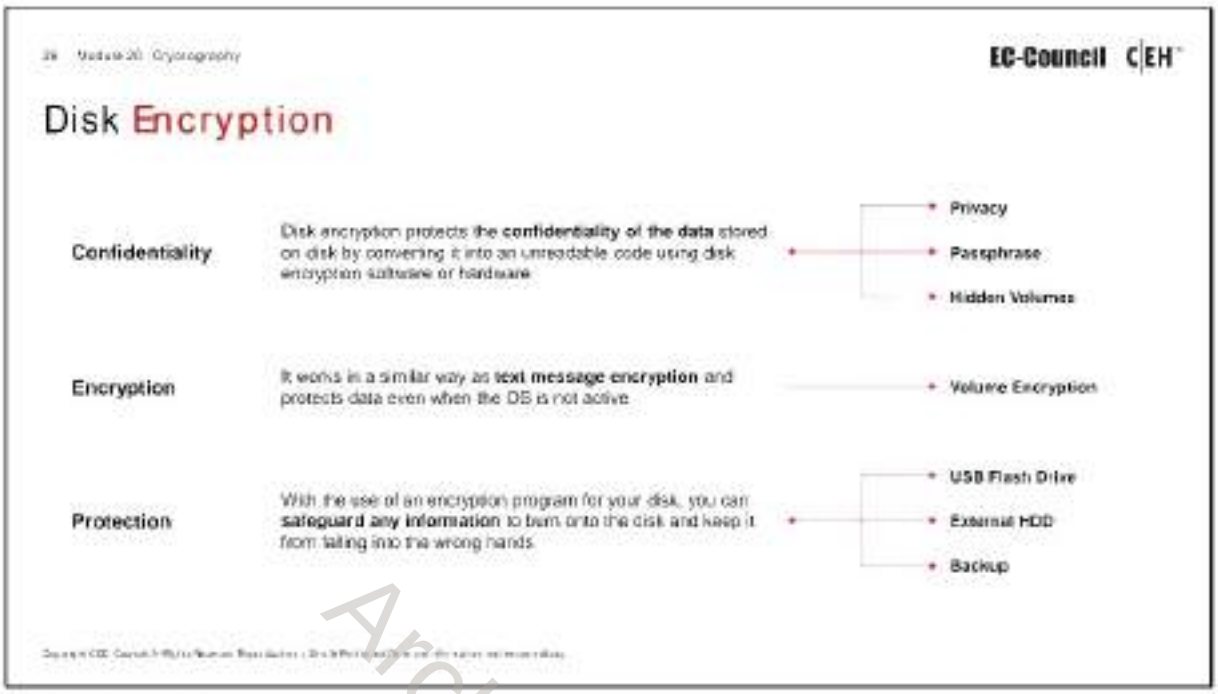
  Source: *https://rmail.com*

  RMail is an email security tool that provides open tracking, delivery proof, email encryption, electronic signatures, large file transfer functionality, etc. RMail works seamlessly with users' existing email platforms, including Microsoft Outlook, Gmail, etc. Using this tool, you can encrypt sensitive emails and attachments for security or legal compliance.

Figure 20.46: Screenshot of RMail

Some additional email encryption tools are as follows:

- Mailvelope (*https://mailvelope.com*)
- Virtru (*https://www.virtru.com*)
- Webroot™ (*https://www.webroot.com*)
- Secure Email (S/MIME) Certificates (*https://www.ssl.com*)
- Proofpoint Email Protection (*https://www.proofpoint.com*)
- Paubox (*https://www.paubox.com*)

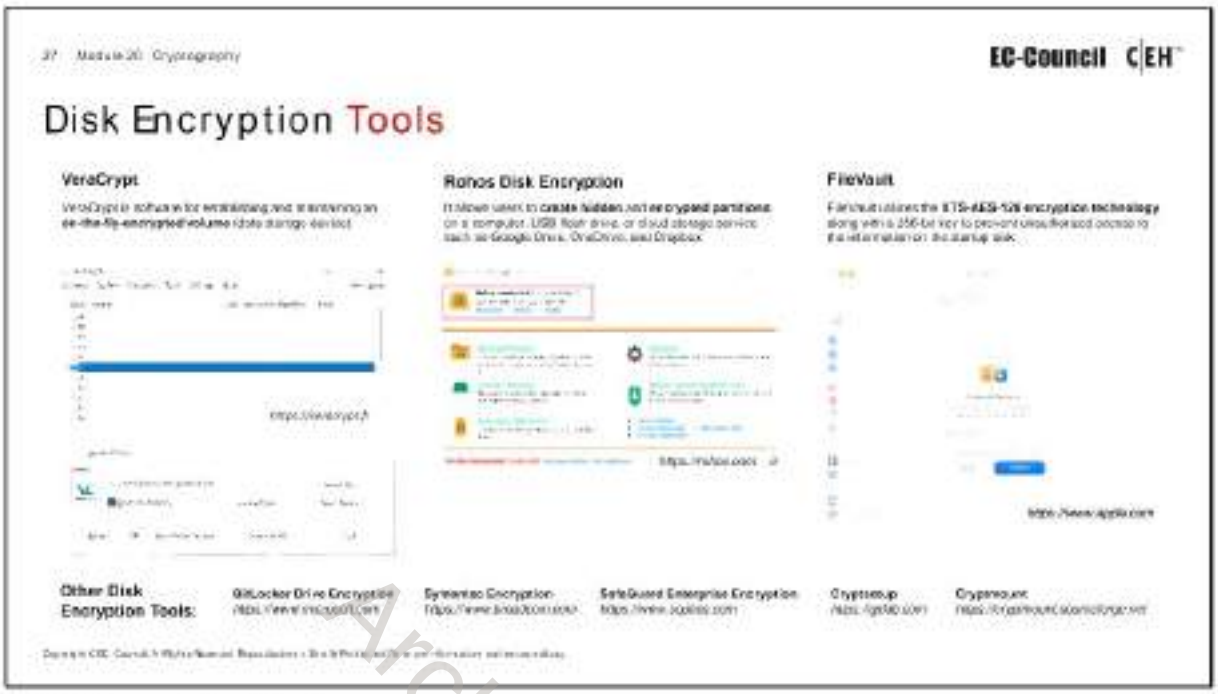


Disk Encryption

Disk encryption encrypts every bit of data stored on a disk or a disk volume, thus preventing illegal access to data storage. Disk encryption is a technology that protects the confidentiality of the data stored on a disk by converting it into an unreadable code using disk encryption software or hardware, thus preventing unauthorized users from accessing it. Disk encryption provides confidentiality and privacy using passphrases and hidden volumes.

Disk encryption works similarly to text-message encryption and protects data even when the OS is not active. By using an encryption program for one's disk (USB flash drive, external HDD, backup), one can safeguard any or all information on the disk and prevent it from falling into the wrong hands. Disk-encryption software scrambles the information on the disk into an illegible code. It is only after decryption of the disk information that one can read and use it.

Disk encryption is useful when the user needs to physically send sensitive information. In addition, disk encryption can protect the real-time exchange of information from compromising threats. When users exchange encrypted information, the chances of compromising the information are minimized. The only way an attacker can access the information is by decrypting the message. Furthermore, encryption software installed on a user's system ensures the security of the system. Install encryption software on any system that holds valuable information or systems that are exposed to unlimited data transfer.



Disk Encryption Tools

The common goal of disk encryption tools is to encrypt a disk partition to provide confidentiality to the information stored on it. Some disk encryption tools are discussed below.

- **VeraCrypt**

  Source: *https://veracrypt.fr*

  VeraCrypt is software for establishing and maintaining an on-the-fly-encrypted volume (data storage device). In on-the-fly encryption, data are automatically encrypted immediately before saving and decrypted immediately after loading, without any user intervention. No data stored on an encrypted volume can be read (decrypted) without using the correct password/keyfile(s) or correct encryption keys. The entire file system is encrypted (e.g., file names, folder names, free space, metadata, etc.).

  Files can be copied to and from a mounted VeraCrypt volume just like they are copied to/from any normal disk (e.g., by simple drag-and-drop operations). Files are automatically decrypted on the fly (in memory/RAM) while they are read or copied from an encrypted VeraCrypt volume. Similarly, files that are written or copied to the VeraCrypt volume are automatically encrypted on the fly (just before they are written to the disk) in RAM.

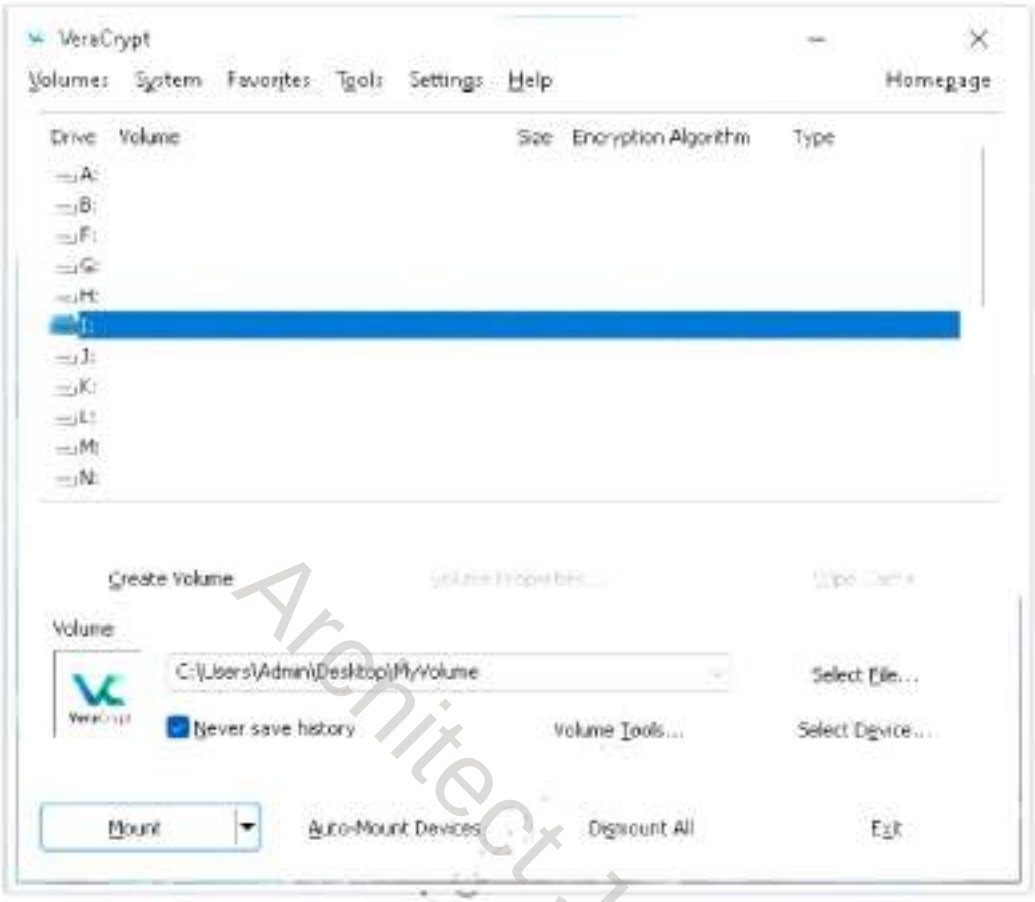


Figure 20.47: Screenshot of VeraCrypt

- **Rohos Disk Encryption**

  Source: *https://rohos.com*

  Rohos is a disk encryption tool that allows users to create hidden and encrypted partitions on a computer, USB flash drive, or cloud storage service such as Google Drive, OneDrive, and Dropbox. The tool uses the NIST-approved AES encryption algorithm and an encryption key length of 256 bits, which enables automatic encryption.

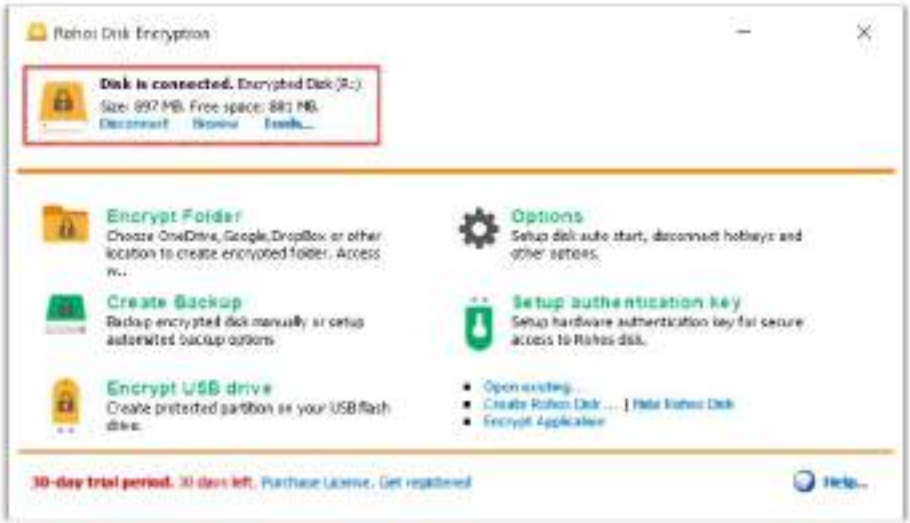


Figure 20.48: Screenshot of Rohos Disk Encryption

- **BitLocker Drive Encryption**

  Source: *https://www.microsoft.com*

  BitLocker provides offline-data and OS protection for your computer. It helps ensure that data that is stored on a computer that is running Windows® is not revealed if the computer is tampered with when the installed OS is offline. BitLocker uses a microchip that is called a Trusted Platform Module (TPM) to provide enhanced protection for your data and to preserve early boot-component integrity. The TPM can help protect your data from theft or unauthorized access by encrypting the entire Windows volume.

Figure 20.49: Screenshot of BitLocker Drive Encryption

Some additional disk encryption tools are as follows:

- Symantec Encryption (*https://www.broadcom.com*)
- SafeGuard Enterprise Encryption (*https://www.sophos.com*)
- GiliSoft Full Disk Encryption (*https://www.gilisoft.com*)
- Check Point Full Disk Encryption (*https://www.checkpoint.com*)
- DiskCryptor (*https://diskcryptor.org*)

Disk Encryption Tools for Linux

- **Cryptsetup**

  Source: *https://gitlab.com*

  Cryptsetup is a utility used to conveniently set up disk encryption based on the DMCrypt kernel module. It includes plain dm-crypt volumes, LUKS volumes, loop-AES, TrueCrypt (including the VeraCrypt extension), and BitLocker formats.

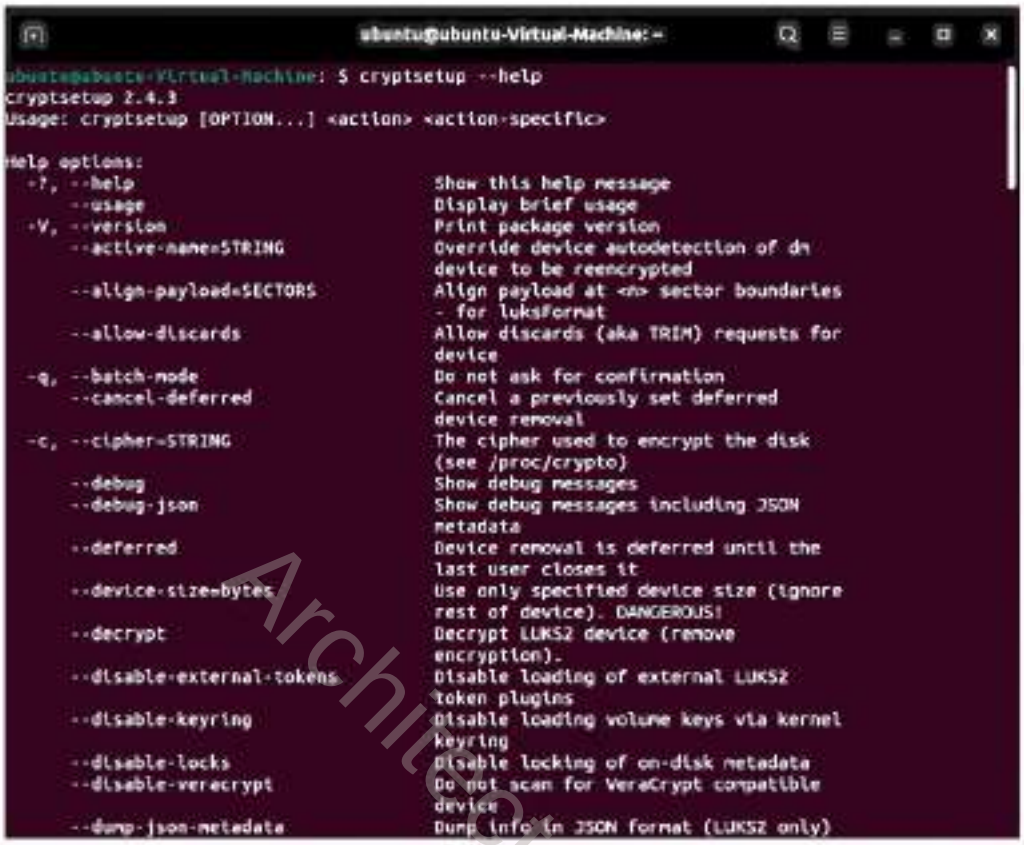


Figure 20.50: Screenshot of Cryptsetup

The following are some additional disk encryption tools for Linux:

- Cryptmount (*https://cryptmount.sourceforge.net*)
- Tomb (*https://dyne.org*)
- CryFS (*https://www.cryfs.org*)
- GnuPG (https://*www.gnupg.org*)
- Harmony Endpoint (*https://www.checkpoint.com*)

Disk Encryption Tools for macOS

- **FileVault**

  Source: *https://support.apple.com*

  FileVault utilizes the XTS-AES-128 encryption technology along with a 256-bit key to prevent unauthorized access to the information on the startup disk. FileVault is available for macOS Lion or higher versions. After turning on FileVault, the user must log in with their account. Soon after turning on FileVault, the encryption process is automatically initiated in the background. The files are encrypted as soon as they are created because they are stored in the startup disk.

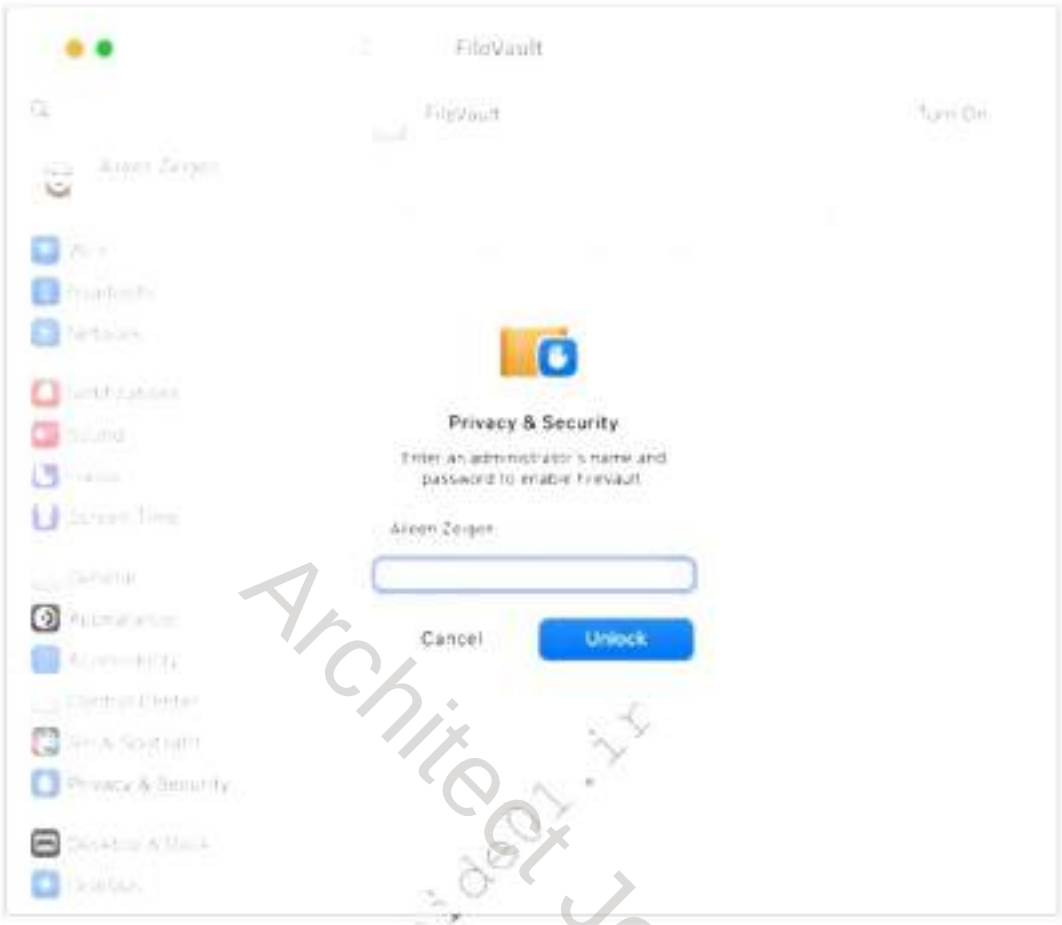
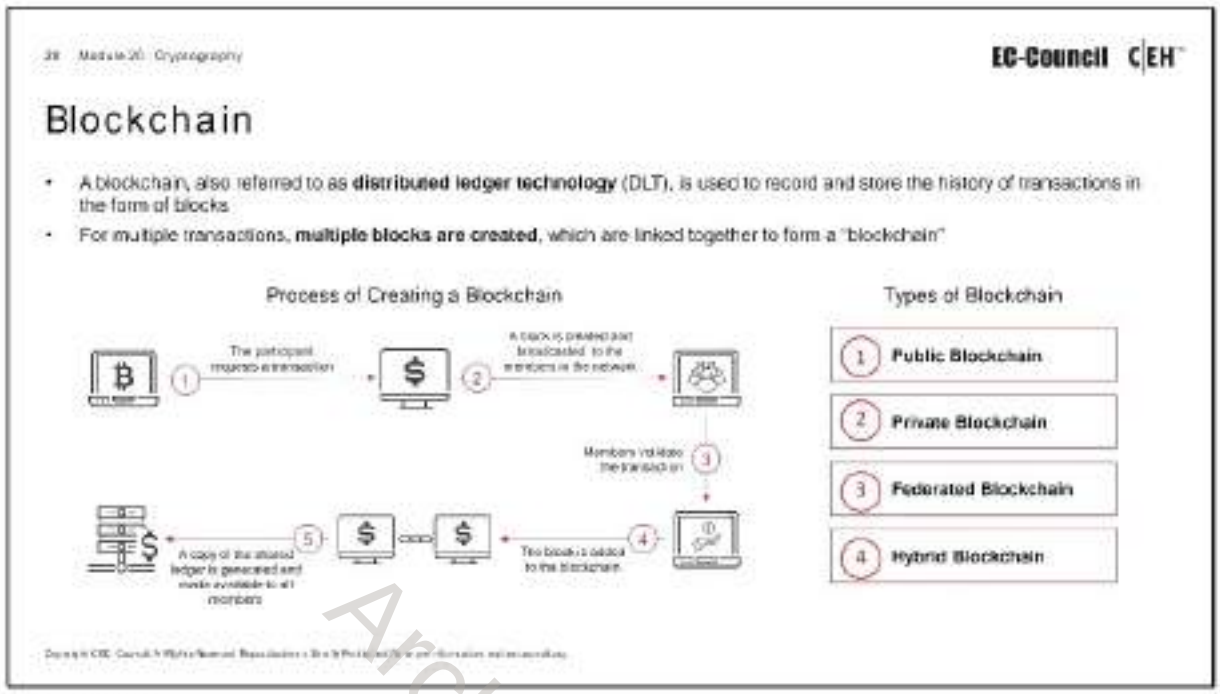


Figure 20.51: Screenshot of FileVault

The following are some additional disk encryption tools for macOS:

- VeraCrypt (*https://www.veracrypt.fr*)
- BestCrypt Volume Encryption (*https://www.jetico.com*)
- Dell Full Disk Encryption (*https://www.dell.com*)
- Comodo Disk Encryption (*https://www.comodo.com*)
- GravityZone Full Disk Encryption (*https://www.bitdefender.com*)



Blockchain

A blockchain is a type of distributed ledger technology (DLT) that is used to record and store the history of transactions securely in the form of blocks. Data recorded in blockchains is resistant to unwanted modifications, and account transparency is maintained through cryptographic techniques. For multiple transactions, multiple blocks are created, which are cryptographically linked together to form a "blockchain." This chain of records or blocks are known as ledgers, which are shared in the network to make other participants aware of all the transaction details and the number of bits owned by each member. The members in the network authenticate blocks using their hash values, and the hashes are further validated by crypto miners using complex cryptographic algorithms, following which, the blocks are approved to join the blockchain mechanism.

Blockchains are generally implemented using two mechanisms: hash functions (mostly SHA-256) and asymmetric key algorithms. The process of validating blocks is known as "proof of work," for which crypto miners are compensated, and the process of adding blocks to a blockchain after performing "proof of work" is referred to as "crypto mining."

Each block in a blockchain consists of three elements: data (transaction details), hash, and the hash of the previous block. Every time a new block is created using a new hash value, the hash value is shared with the next block. The first block in a blockchain referred to as the genesis and is represented by 0s. Once a block verifies its previous block's hash, it is allowed to join the blockchain.

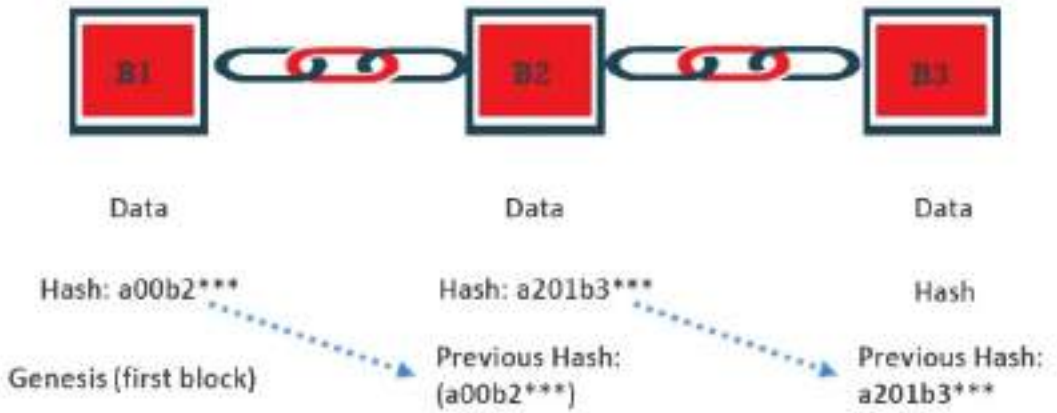


Figure 20.52: Blockchain

If a block is tampered with, the next block in the chain invalidates it because it does not match the previous hash value in the current block. However, a blockchain is not completely protected by merely generating hashes and comparing them with other blocks. Attackers can generate valid hashes for each block using many cryptographic techniques. The "proof of work" mechanism is used, as described above, to mitigate such risks. The security of the blockchain is ensured by both the effective usage of hashes and the "proof of work" by miners (on public ledgers).

Figure below illustrates the process of creating a blockchain.

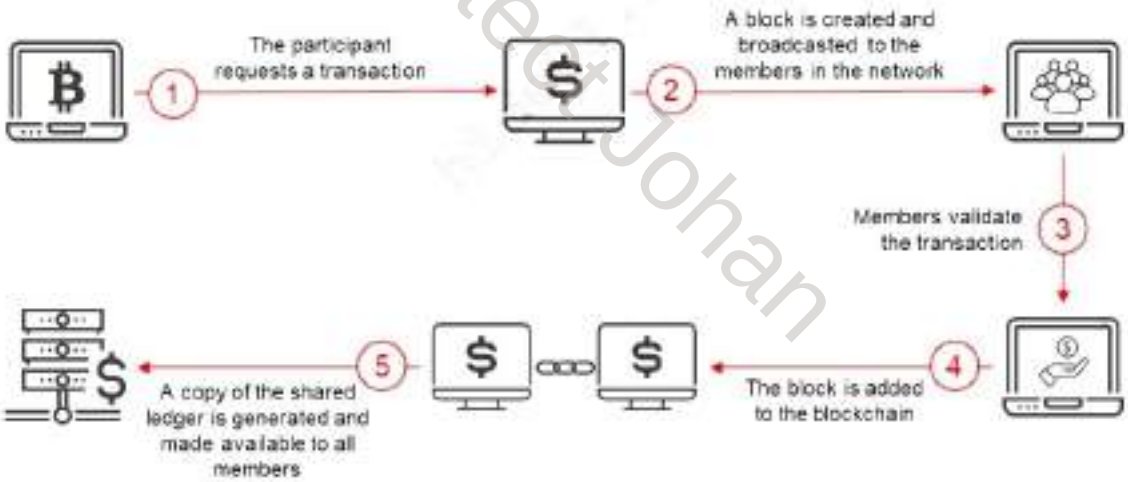


Figure 20.53: Creation of a blockchain

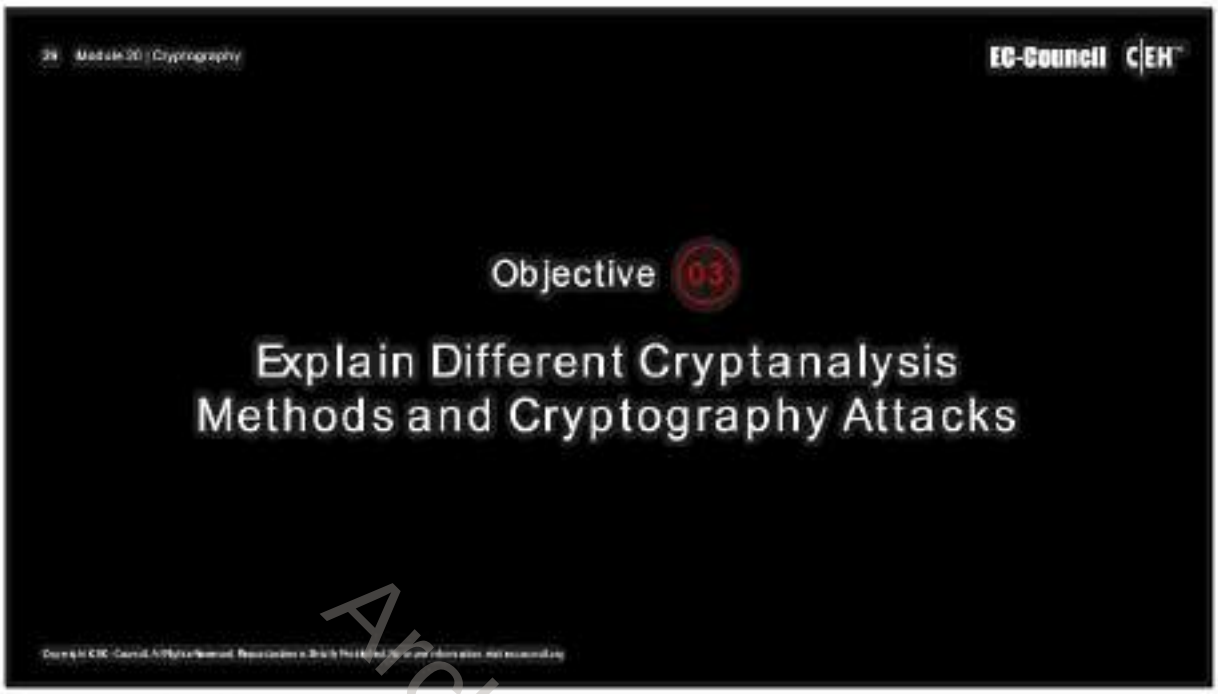
In figure, a block is created by a person involved in a transaction. This block is shared with all the members in the network. Each member validates the block using its hashes, following which the block is added to the blockchain. Thus, each member has the details of the new transaction. Blockchains can be created in four variants, each of which serves a different purpose.

- **Public ledger or public blockchain**: This type of blockchain has no central authority or administration to manage the blocks or ledgers. It is a decentralized and permission-less network in which anyone can join, create, and share blocks. Once the data on the blockchain have been validated, it is secure from modifications or alterations. Some

examples of public blockchains include Bitcoin and Ethereum. Each member in this blockchain can access a copy of other ledgers without any permissions. The following are the key aspects of public ledgers.

- Everyone in the network can participate in validation.
- Once a block is created, it cannot be modified or tampered with.
- Public ledgers can be employed in different sectors such as education and healthcare.
- Public ledgers are suitable for B2C services.

- **Private ledger or private blockchain:** In this type of blockchain, a supervisor or central authority decides who can join and participate in the blockchain network. In a private ledger, only the members involved in a transaction will have knowledge about the corresponding ledgers. Some examples of private blockchains are Hyperledger and Ripple (XRP). The following are the key aspects of private ledgers.
  - An administrator provides a certain level of access to participants.
  - Organizations can add and delete participants on demand.
  - Private ledgers can be employed in sectors such as defense and banking.
  - Private ledgers are suitable for B2B services.
- **Federated blockchain or consortium blockchain:** It is a partially decentralized blockchain in which a group of individuals or organizations, rather than a single entity as in private blockchains, create and manage separate blockchain networks. Control over the blockchain is provided to a group of predetermined or trusted nodes. Participants in a consortium blockchain are mostly from government organizations or central banks. This type of blockchain is extremely fast and scalable. EWF (Energy) and R3 (banks) are instances of federated blockchains.
- **Hybrid blockchain**: It is a combination of both private and public blockchain. In a hybrid blockchain, only a selected set of records or data from the blockchain can be publicly accessed; the remaining data are kept confidential in a private network. This type of blockchain enables organizations to select which data they wish to make public private. One important example of a hybrid blockchain is the IBM Food Trust.

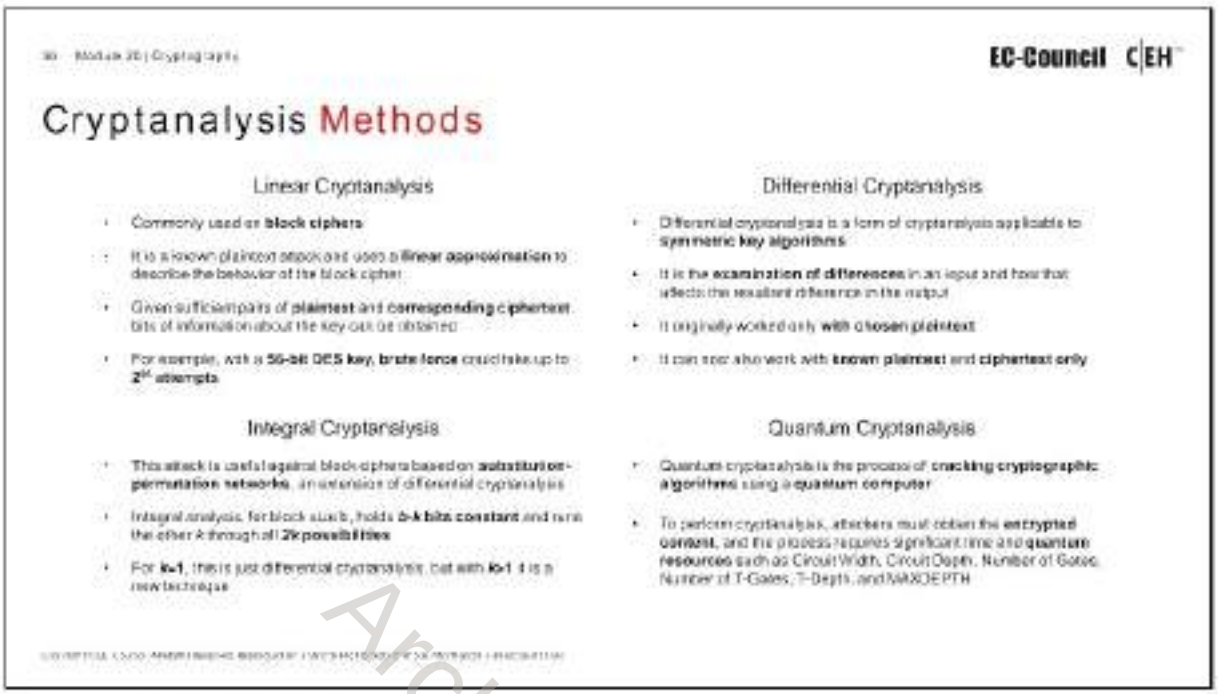


Cryptanalysis

Attackers may implement various cryptography attacks to evade the security of a cryptographic system by exploiting vulnerabilities in code, ciphers, cryptographic protocols, or key management schemes. This process is known as cryptanalysis.

Cryptanalysis is the study of ciphers, ciphertext, or cryptosystems with the ability to identify vulnerabilities in them and thus extract plaintext from ciphertext even if the cryptographic key or algorithm used to encrypt the plaintext is unknown.

This section deals with various cryptography attacks that an attacker performs to compromise cryptographic systems as well as various cryptanalysis techniques and tools that help in breaching cryptographic security.



Cryptanalysis Methods

- **Linear Cryptanalysis**

  Linear cryptanalysis is based on finding linear approximations to the action of a cipher, particularly block ciphers. This technique was invented by Mitsuru Matsui. It is a known plaintext attack that uses a linear approximation to describe the behavior of the block cipher. Given sufficient pairs of plaintext and corresponding ciphertext, bits of information about the key can be obtained. Obviously, the more pairs of plaintext and ciphertext one has, the greater the chances of success.

  Remember that cryptanalysis is an attempt to break cryptography. For example, with the 56-bit Data Encryption Standard (DES), brute-forcing the key could take up to 2^{56} attempts. Linear cryptanalysis, on the other hand, requires approximately 2^{43} known plaintexts to succeed. While this is better than brute-forcing, it is still impractical in most situations without sufficient data.

  The math may be complex for novice cryptographers, but let's look at the basics. In linear cryptanalysis, a linear equation expresses the equality of two expressions consisting of XORed binary variables. For example, the following equation XORs the sum of the first and third plaintext bits, and the first ciphertext bit is equal to the second bit of the key:

  $$P_1 \oplus P_3 \oplus C1 = K_2$$

  You can use this method to gradually recover the key that was used. After deriving such equations for each bit, you will have an equation of the form:

  $$P_{i1} \oplus P_{i2} \oplus \ldots \oplus C_{j1} \oplus C_{j2} \oplus \ldots = K_{k1} \oplus K_{k2} \oplus \ldots$$

Matsui's Algorithm 2 can then be applied, using known plaintext-ciphertext pairs, to guess the values of the key bits involved in the approximation. For each set of values of the key bits on the right-hand side (referred to as a partial key), count how many times the approximation holds true over all the known plaintext-ciphertext pairs; call this count T. The partial key whose T has the greatest absolute difference from half the number of plaintext-ciphertext pairs is designated as the most likely set of values for those key bits.

- **Differential Cryptanalysis**

  Differential cryptanalysis is a form of cryptanalysis applicable to symmetric-key algorithms. It was invented by Eli Biham and Adi Shamir. Essentially, it is the examination of differences in input and how that affects the resultant difference in the output. It originally worked only with chosen plaintext. It can also work with known **plaintext** and **ciphertext**.

- **Integral Cryptanalysis**

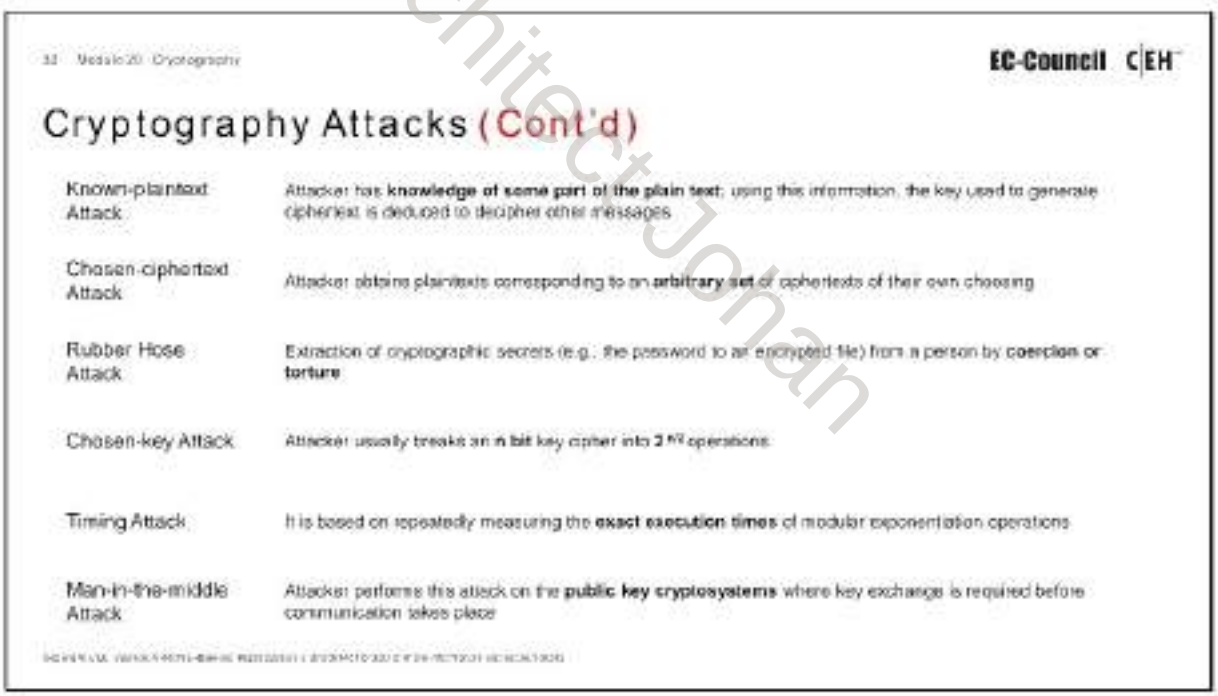
  Integral cryptanalysis was first described by Lars Knudsen. This attack is particularly useful against block ciphers based on substitution-permutation networks as an extension of differential cryptanalysis. The differential analysis looks at pairs of inputs that differ in only one bit position, with all other bits being identical. Integral analysis for block size b holds b-k bits constant and runs the other k bits through all 2k possibilities. For k = 1, this is just differential cryptanalysis, but with k > 1, it is a new technique.

- **Quantum Cryptanalysis**

  Quantum cryptanalysis is the process of cracking cryptographic algorithms using a quantum computer. Attackers can use Shor's quantum factoring algorithm on public-key cryptographic algorithms such as RSA and Elliptic Curve Diffie-Hellman (ECDH) to find the factors of large numbers in polynomial time and Grover's quantum search algorithm to make brute-force key search faster for block ciphers (AES) or hash functions (SHA).

  To perform cryptanalysis, attackers must obtain the encrypted content, and the process requires significant time and quantum resources. Given below are the resources required to conduct cryptanalysis.

  - **Circuit Width**: Specifies how many quantum bits or qubits are required in a time step
  - **Circuit Depth**: Specifies the time steps required for a circuit
  - **Number of Gates**: Specifies how many quantum gates are implemented in a circuit
  - **Number of T-Gates**: Specifies how many T-gates are implemented in a circuit
  - **T-Depth**: Specifies the time steps required for a T-gate
  - **MAXDEPTH**: Specifies the maximum depth of a circuit (e.g., 2^{40},2^{64}, or 2^{96})



Cryptography Attacks

Attackers conduct cryptography attacks by assuming that the cryptanalyst has access to the encrypted information. A cryptography attack or cryptanalysis involves the study of various principles and methods of decrypting the ciphertext back to the plaintext without knowledge of the key.

The various types of cryptography attacks are as follows:

- **Ciphertext-only Attack**

  Ciphertext-only is less effective but much more likely for the attacker. The attacker only has access to a collection of ciphertexts. This is much more likely than known plaintext but is also the most difficult. The attack is completely successful if the corresponding plaintexts (or even better, the key) can be deduced. The ability to obtain any information at all about the underlying plaintext is still considered a success. So what does the attacker do with the ciphertexts he/she has accumulated? You can analyze them for patterns, trying to find something that would give you a hint as to the key that was used to crack them. Often, the result of this attack is just a partial break and not a complete break.

- **Adaptive Chosen-plaintext Attack**

  In this type of attack, an attacker has complete access to the plaintext message including its encryption, and he/she can also modify the content of the message by making a series of interactive queries, choosing subsequent plaintext blocks based on the information from the previous encryption queries and functions. To perform this attack, an attacker needs to interact with the encryption device.

- **Chosen-plaintext Attack**

  A chosen plaintext attack is a highly effective type of cryptanalysis attack. In this attack, the attacker obtains the ciphertexts corresponding to a set of plaintexts of his/her own choosing. This allows the attacker to attempt to derive the key used and thus decrypt other messages encrypted with that key. Basically, since the attacker knows the plaintext and the resultant ciphertext, he/she gains many insights into the key used. This technique can be difficult but is not impossible.

- **Related-Key Attack**

  The related-key attack is similar to the chosen plaintext attack, except that the attacker can obtain ciphertexts encrypted under two different keys. This is actually a very useful attack if you can obtain the plaintext and matching ciphertext. The attack requires that the differing keys be closely related, e.g., in a wireless environment where subsequent keys might be derived from previous keys. Then, while the keys are different, they are close. Much like the ciphertext-only attack, this type of attack is most likely only going to yield a partial break.

- **Dictionary Attack**

  In this attack, the attacker constructs a dictionary of plaintext along with its corresponding ciphertext that he/she has analyzed and obtained for a certain period of time. After building the dictionary, if the attacker obtains the ciphertext, he/she uses the already built dictionary to find the corresponding plaintext. Attackers use this technique to decrypt keys, passwords, passphrases, and ciphertext.

- **Known-plaintext Attack**

  In this attack, the only information available to the attacker is some plaintext blocks along with the corresponding ciphertext and algorithm used to encrypt and decrypt the text. Using this information, the key used to generate the ciphertext is deduced so as to decipher other messages. This attack works on block ciphers and is an example of linear cryptanalysis. The known plaintext blocks are generated using a series of intelligent guesses and logic, and not by accessing the plaintext over a channel.

- **Chosen-ciphertext Attack**

  The attacker obtains the plaintexts corresponding to an arbitrary set of ciphertexts of his own choosing. Using this information, the attacker tries to recover the key used to encrypt the plaintext. To perform this attack, the attacker must have access to the communication channel between the sender and the receiver.

  There are two variants of this attack:

  - Lunchtime or Midnight Attack: In this attack, the attacker can have access to the system for only a limited amount of time or can access only a few plaintext-ciphertext pairs.
  - Adaptive Chosen-ciphertext Attack: In this attack, the attacker selects a series of ciphertexts and then observes the resulting plaintext blocks.

- **Rubber Hose Attack**

  Attackers extract cryptographic secrets (e.g., the password to an encrypted file) from a person by coercion or torture. In general, people under pressure cannot maintain security, and they will reveal secrets or hidden information. Attackers torture victims to reveal secret keys or passwords used to encrypt the information.

- **Chosen-key Attack**

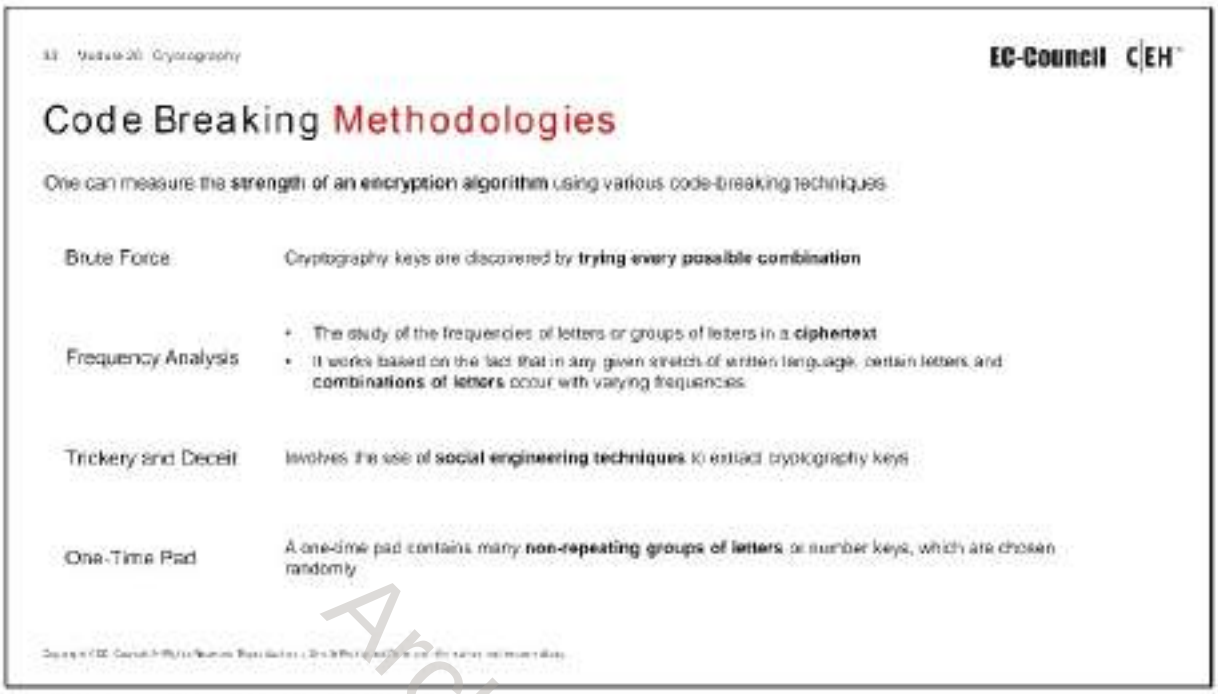
  In this type of attack, an attacker not only breaks a ciphertext but also breaks into a larger system, which is dependent of that ciphertext. The attacker usually breaks an n-bit key cipher into $2^{n/2}$ operations. Once an attacker breaks the cipher, he gets access to the system, and he can control the whole system, access confidential data, and perform further attacks.

- **Timing Attack**

  It is based on repeatedly measuring the exact execution times of modular exponentiation operations. The attacker tries to break the ciphertext by analyzing the time taken to execute the encryption and decryption algorithm for various inputs. In a computer, the time taken to execute a logical operation may vary based on the input given. An attacker tries to extract the plaintext by giving varying inputs.

- **Man-in-the-Middle Attack**

  This attack is performed against a cryptographic protocol. Here, an attacker intercepts the communication between a client and a server and negotiates the cryptographic parameters. Using this attack, an attacker can decrypt the encrypted content and obtain confidential information such as system passwords. An attacker can also inject commands that can modify the data in transit. The attacker usually performs an MITM attack on public-key cryptosystems where key exchange is required before communication takes place.



Code Breaking Methodologies

One can measure the strength of an encryption algorithm using various code breaking techniques, some of which are as follows:

- **Brute Force**

  Code breakers or cryptanalysts work to recover the plaintext of a message without knowing the required key in advance. They may first try to recover the key, or they may go after the message itself. A common cryptanalytic technique is a brute-force attack, or exhaustive search, in which the keys are determined by trying every possible combination of characters.

  The efficiency of a brute-force attack depends on the hardware configuration. The use of faster processors means that more keys will be tested per second. Cryptanalysts carried out a successful brute-force attack on a DES encryption method, which effectively rendered DES obsolete.

- **Frequency Analysis**

  Frequency analysis is the study of the frequency of letters or groups of letters in a ciphertext. Frequency analysis of letters and words is another method used to crack ciphers. It works on the principle that, in any given stretch of written language, certain letters and combinations of letters occur with varying frequencies. This technique examines the number of times that a particular symbol appears in a ciphertext. For example, the letter "**e**" is a common letter in the English language. If the letter "**k**" appears commonly in a ciphertext, it can be reasonably concluded that "**k**" in the encrypted language is equivalent to "**e**" in English.

Encrypted source code is more vulnerable to these types of attacks because words such as "**#define**," "**struct,**" "**else,**" and "**return**" are repeated frequently in code. Sophisticated cryptosystems are required to maintain the security of messages against frequency analysis.

- **Trickery and Deceit**

  Trickery and deceit require a high level of mathematical and cryptographic skills. It involves the use of social engineering techniques to extract cryptography keys.

  Example: It is fairly easy to decrypt an entire message if the user knows some of its content.

  An attacker can use social engineering techniques to trick or bribe someone to encrypt and send a known message, which, when intercepted, could then be easily decrypted using standard cryptanalysis techniques.

- **One-Time Pad**

  One can crack any cipher if provided with sufficient time and resources. However, there is an exception called a one-time pad, which users assume to be unbreakable even with infinite resources.

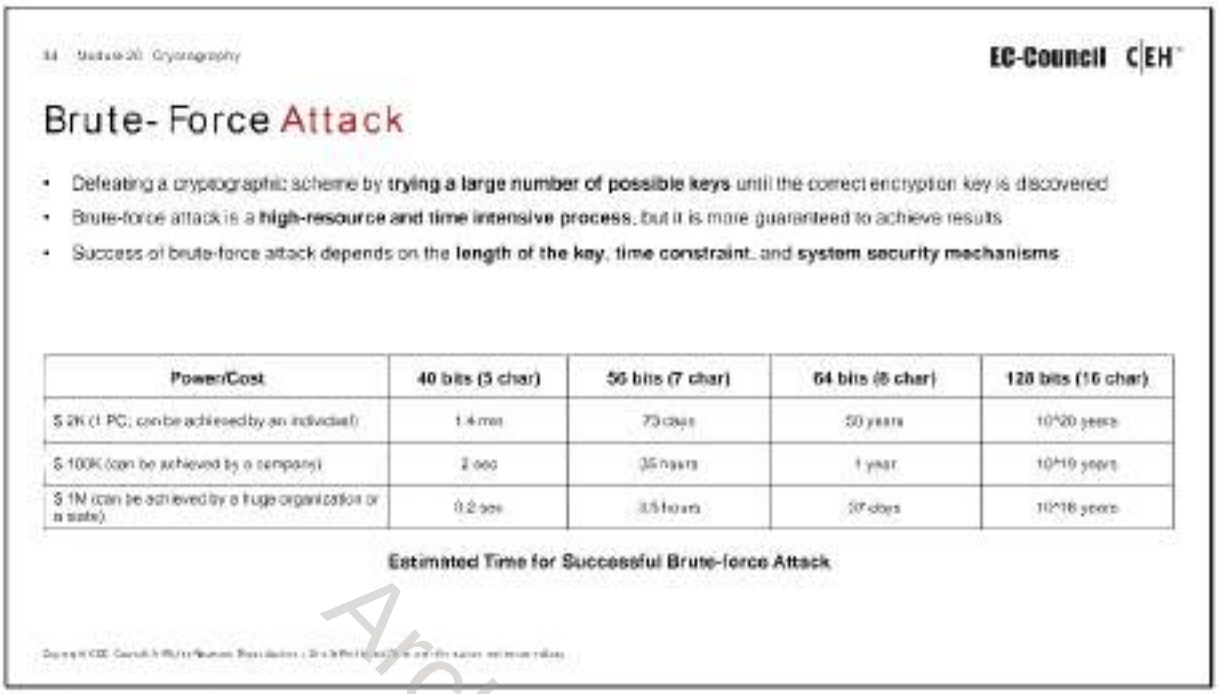
  A one-time pad mostly contains a non-repeating set of letters or numbers, which the system chooses randomly. The user writes them on small sheets of paper and then pastes them together in a pad.

  Example of One-time pad usage:

  The sender encrypts only one plaintext character using each key letter on the pad, and the receiver decrypts each letter of the ciphertext using an identical pad. Once the letter uses a page, he or she tears it off the pad and securely discards it; hence, the name one-time pad.

  Drawback:

  The key length is the same as that of the message, thus making it impossible to encrypt and send large messages.



Brute-Force Attack

It is extremely difficult to crack cryptographic systems, as they have no practical weaknesses to exploit; however, it is not impossible. Cryptographic systems use cryptographic algorithms to encrypt a message. These cryptographic algorithms use a key to encrypt or decrypt messages. In cryptography, this key is the important parameter that specifies the transformation of plaintext to ciphertext and vice versa. If you are able to guess or find the key used for decryption, then you can decrypt the messages and read them in clear text. 128-bit keys are common and considered strong. From a security perspective, to avoid guessing the key, cryptographic systems use randomly generated keys. This makes you devote considerable effort toward guessing the key. However, you still have a choice to determine the key used for encryption or decryption.

You can attempt to decrypt a message using all possible keys until you discover the key used for encryption. This method of discovering a key is called a brute-force attack. However, doing so requires a massive amount of processing power. It is a resource-intensive and time-intensive process. For any non-flawed protocol, the average time needed to find the key in a brute-force attack depends on the length of the key. If the key length is short, then it will take less time to find the key; if it is long, it will take more time. A brute-force attack will be successful if and only if the attacker has enough time to discover the key. However, the time required is relative to the length of the key.

The difficulty of a brute-force attack depends on various factors, such as

- The length of the key
- The number of possible values each component of the key can have
- The time it takes to attempt each key

- If there is any mechanism that locks the attacker out after a certain number of failed attempts

For example, if a system could brute-force a DES 56-bit key in one second, then for an AES 128-bit key, it takes approximately 149 trillion years. To perform a brute-force attack, the attacker needs double the time for every additional bit of key length; the reason is that the number of keys doubles with an increase of one bit.

However, a brute-force attack is more likely to achieve results.

Power/Cost	40 bits (5 char)	56 bits (7 char)	64 bits (8 char)	128 bits (16 char)
$ 2K (1 PC. Can be achieved by an individual)	1.4 min	73 days	50 years	10^20 years
$ 100K (this can be achieved by a company)	2 sec	35 hours	1 year	10^19 years
$ 1M (Achieved by a huge organization or a state)	0.2 sec	3.5 hours	37 days	10^18 years

Table 20.6: Estimate time for a successful brute-force attack

Birthday Attack

A birthday attack refers to a class of brute-force attacks against cryptographic hashes that renders brute-forcing easier to performs. This attack depends on the birthday paradox, which is the probability of two or more people in a group of 23 sharing the same birthday is greater than 0.5.

Birthday Paradox

For example, how many people are needed to have a high likelihood that two will share the same birthday (i.e., same day and month, not year). There are 365 days a year, and therefore, you might think that at least half or 182 people share the same birthday, when it is actually only 23!

The basic idea is as follows: How many people would you need to have in a room to have a strong likelihood that two amongst them would have the same birthday (same day and month, but not year). Obviously, if you put 367 people in a room, at least two of them must have their birthdays on the same day and month since there are only 365 days in a year, and an additional day in the case of a leap year. The paradox is not the number of people you need to guarantee a match, but the number of people you need to have a strong probability. Even with 23 people in a room, there is a 50% chance that two them will have their birthdays on the same day and month.

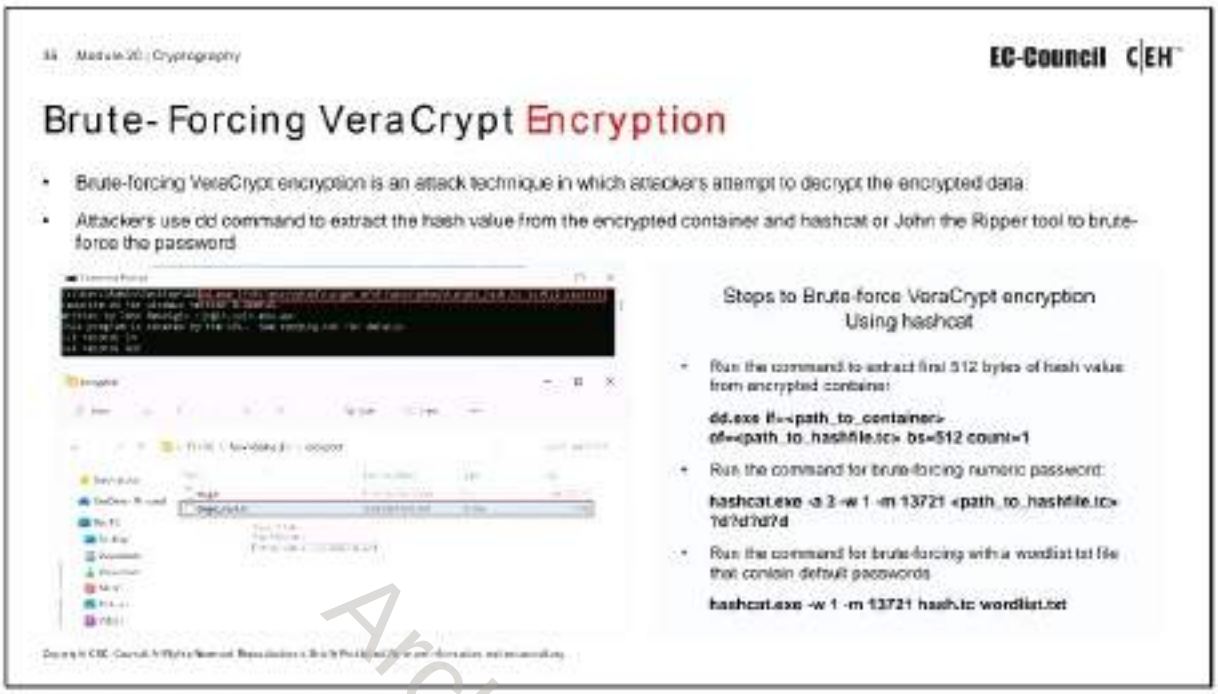
Birthday Paradox: Probability

The probability that the first person does not share a birthday with any previous person is 100% because there are no previous people in the set. This can be written as 365/365. The second person has only one preceding person, and the odds that the second person has a birthday different from the first are 364/365. The third person might share a birthday with two preceding people, so the odds of sharing a birthday with either of the two preceding people are 363/365.

Because each of these are independent, we can compute the probability as follows: 365/365 * 364/365 * 363/365 * 362/365 ... * 342/365 (342 is the probability of the 23rd person who shares a birthday with a preceding person). When we convert these to decimal values, it yields (truncating at the third decimal point) 1 * 0.997 * 0.994 * 0.991 * 0.989 * 0.986 * ... 0.936 = 0.49 or 49%. This is the probability that 23 people will not have any birthdays in common; thus, there is a 51% (better than even odds) chance that two of the 23 will have a birthday in common.

The idea behind the birthday attack is to attempt to find a collision for a given hash. Now, assume that the hash is MD5 with a 128-bit output. You would have to try 2^128 possible hashes to guarantee a collision, which is a very large number. In decimal notation, it is 3.40282366920938463463374607431768e+38

Now, from the birthday paradox, we need 1.174√2^128 or 21656477542535013597.184 hashes to guarantee a collision. Furthermore, this is still a very large number but many orders of magnitude smaller than the abovementioned value.



Brute-Forcing VeraCrypt Encryption

Brute-forcing VeraCrypt encryption is an attack technique in which attackers attempt to decrypt the encrypted data. Attackers use the dd command to extract the hash value from the encrypted container, and the hashcat or John the Ripper tool to brute-force the password by attempting different password combinations until the key used for encryption is discovered.

Steps to Brute-force VeraCrypt Encryption Using hashcat

- **Extract VeraCrypt Hash**

  Run the following command is run to extract the first 512 bytes of the hash value from the encrypted container and save it in a .tc extension file:

```
dd.exe  if=<path_to_container>  of=<path_to_hashfile.tc>  bs=512
count=1
```

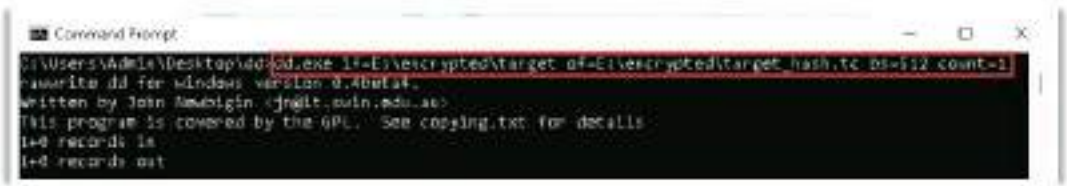


Figure 20.54: Screenshot showing dd command to extract hash value

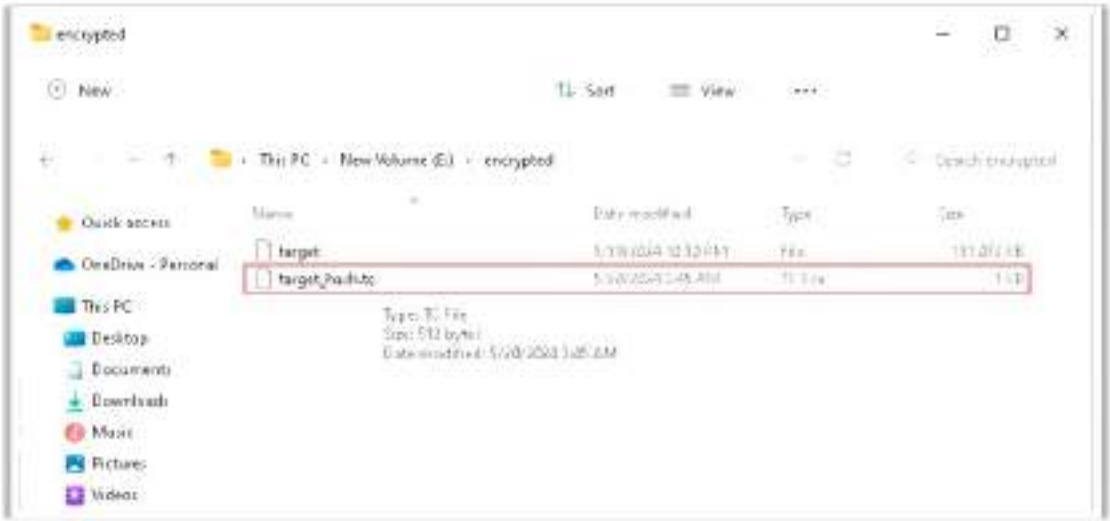


Figure 20.55: Screenshot showing the extracted hash

The above hash will be used for brute forcing.

- **Crack the Hash value using hashcat**

  Run the following command for brute-forcing a 4-digit numeric password:

  `hashcat.exe -a 3 -w 1 -m 13721 <path_to_hashfile.tc> ?d?d?d?d`

  - **a**: Brute-force attack mode
  - **-w**: Workload profile (low to high: 1-4)
  - **-m**: Decryption mode
  - **13721**: Decryption code
  - **?d?d?d?d**: Running hashcat with masks (each ?d indicates a numeric value 0-9)

The below screenshot shows the cracked password:

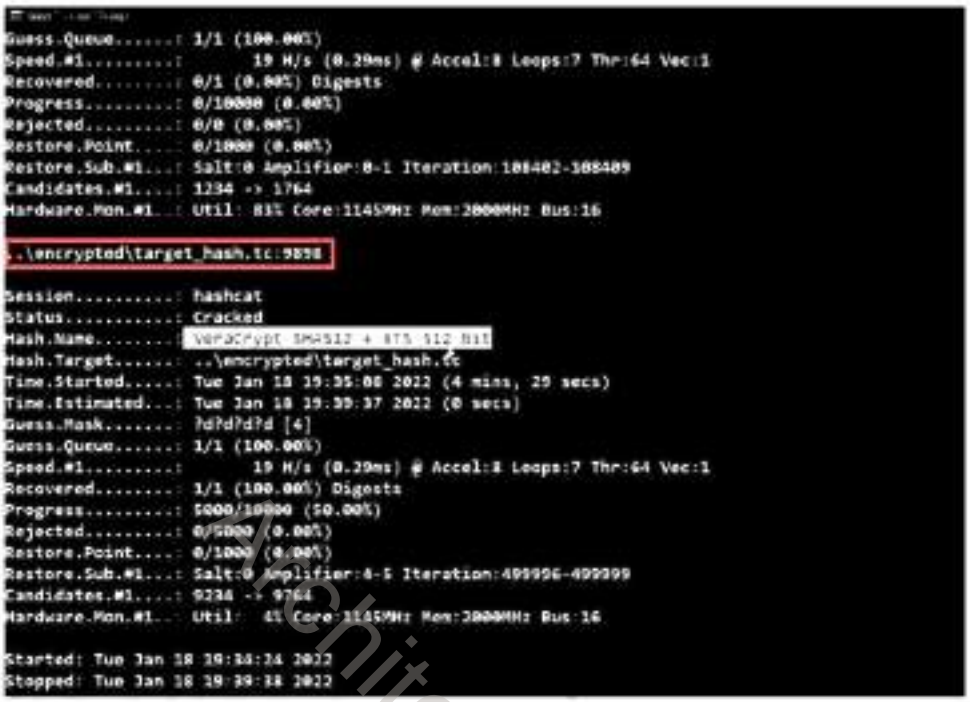
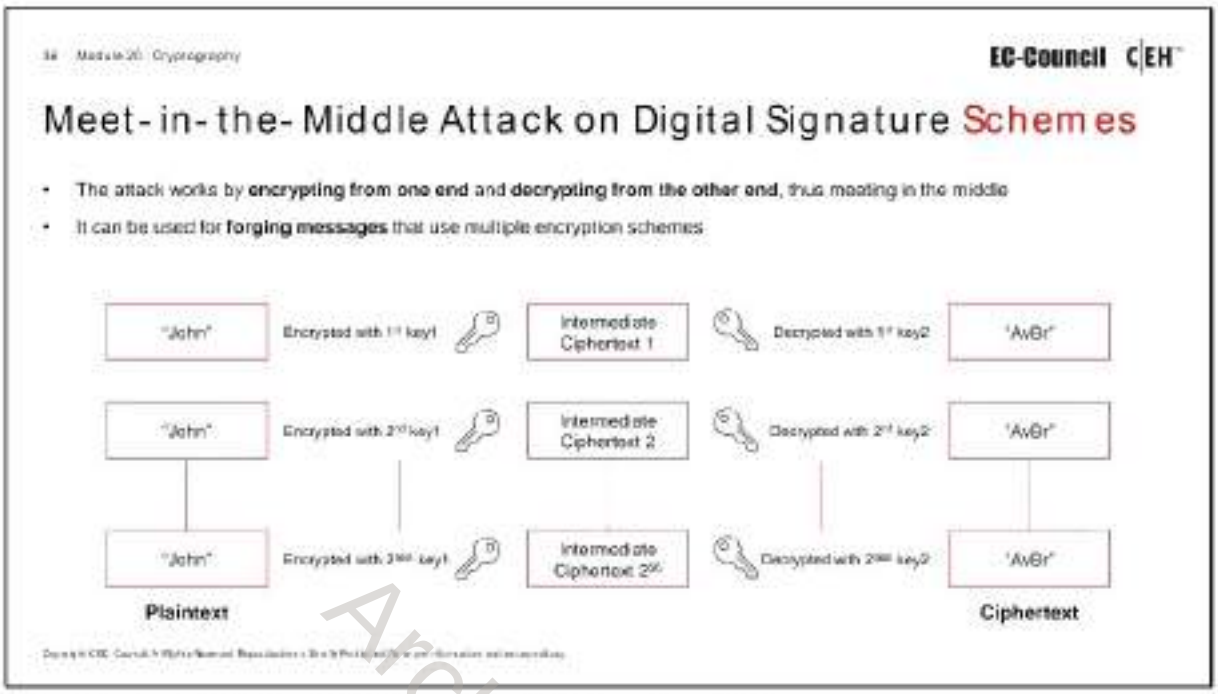


Figure 20.56: Screenshot showing the result of status, number of attempts, speed, and time elapsed

- Additionally, attackers can use the following command for brute forcing with a wordlist.txt file containing default passwords:

```
hashcat.exe -w 1 -m 13721 hash.tc wordlist.txt
```



Meet-in-the-Middle Attack on Digital Signature Schemes

A meet-in-the-middle attack is the best attack method for cryptographic algorithms using multiple keys for encryption. This attack reduces the number of brute-force permutations required to decode text encrypted by more than one key. A meet-in-the-middle attack uses space-time trade-off; it is also a type of birthday attack because it exploits the mathematics behind the birthday paradox, and the attack consumes less time than an exhaustive attack. It is called a meet-in-the-middle attack because it works by encrypting from one end and decrypting from the other end, thereby meeting "in the middle."

In the meet-in-the-middle attack, the attacker uses a known plaintext message. The attacker has access to both the plaintext as well as the respective encrypted text. This attack is performed by attackers for forging messages that use multiple encryption schemes.

Consider an example where the plaintext is "**John**," and the resulting double-DES-encrypted message is "**AvBr.**" To recover both the keys (i.e., key1 and key2) used for encryption, the attacker performs a brute-force attack on key1 using all the **2^{56} different single DES** possible keys to encrypt the plaintext of "John" and saves each key and the resulting intermediate ciphertext in a table. The attacker brute-forces key2 and decrypts "AvBr" up to **2^{56} times**. The attack is successful when the second brute-force attack gives the same result as the intermediate ciphertext present in the ciphertext table after the first brute-force attack. Once the attacker finds a match, he/she can determine both keys and complete the attack. At most, this attack takes 2^{56} or a maximum of **2^{57} total operations**. This enables the attacker to gain access to the data easily compared with double DES.

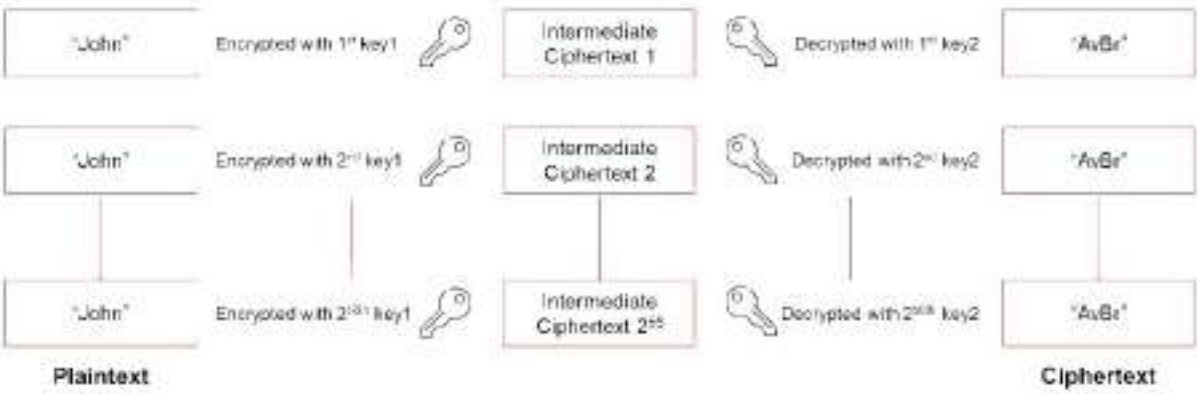
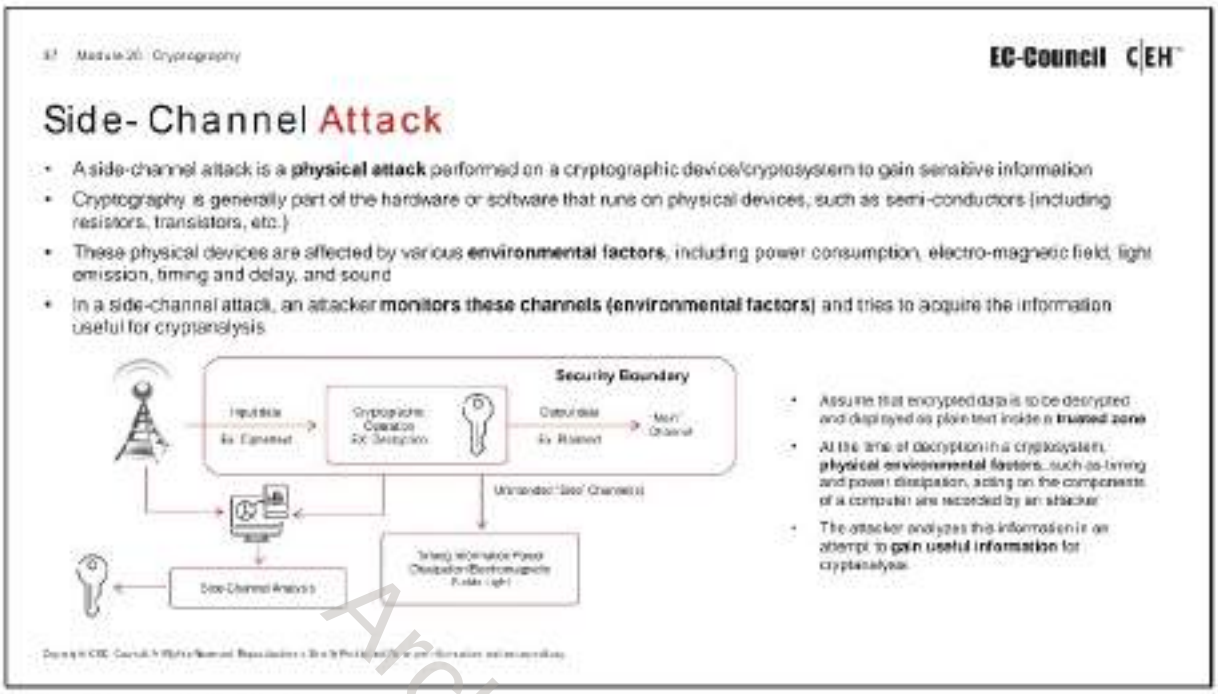


Figure 20.57: Meet-in-the-Middle attack on digital signature schemes



Side-Channel Attack

A side-channel attack is a physical attack performed on a cryptographic device/cryptosystem to gain sensitive information. Cryptography is generally part of the hardware or software that runs on physical devices such as semi-conductors (resistor, transistor, and so on) that interact with and affect various environmental factors as follows:

- **Power Consumption**

  Reveals operations that take place and parameters involved. It is applicable only to hardware cryptosystems. Power consumption analysis is of two types:

  - **Simple Power Analysis (SPA)**: Provides information regarding the instruction being executed at a certain time and the values of input and output
  - **Differential Power Analysis (DPA)**: It does not require the knowledge of the details of algorithm implementation; it exploits statistical methods

- **Electromagnetic Field**

  Computer components often generate electromagnetic radiation. By measuring the variations of the electromagnetic field over the chip surface, an attacker can predict its correlation to the underlying computation and data and may be able to deduce some valuable information about this computation and data.

- **Light Emission**

  Kuhn found that the average luminosity of a cathode ray tube (CRT) diffuse reflection of a wall is sufficient to reconstruct the signal displayed on the CRT. Thus, an attacker can gather ample information by reading the signals that a trusted computing platform's optical output channels emit.

According to Loughry and Umphress, one can deduce the data a computer is processing based on the optical radiation emitted from its LED (light-emitting diode) status indicators.

- **Timing and Delay**

  Systems often compute cryptographic algorithms without time consistency owing to performance optimizations. If such computations involves secret data, then the variations in time can be used to infer the secret information. Here, the attacker analyzes the time taken by a cryptographic device to process each message to discover the secret parameters.

- **Sound**

  Acoustic attacks exploit the sound produced during a computation. These acoustic emissions are from keyboards and computing components (e.g., CPU, memory)

In a side-channel attack, an attacker monitors these channels (environmental factors) and tries to acquire useful information for cryptanalysis. The information thus acquired is termed as side-channel information. Side-channel attacks are different from traditional/theoretical forms of attacks such as brute-force attacks. The side-channel attack depends on the way in which systems implement cryptographic algorithms rather than the algorithm itself.

Mitigation Techniques for Side-Channel Attacks

- Use differential power analysis (DPA) proof protocols with delimited side-channel leakage characteristics and update the keys before the leakage accumulation is significant.
- Use fixed-time algorithms (i.e., no data-dependent delays).
- Mask and blind algorithms using random nonces.
- Implement differential matching techniques to minimize net data-dependent leakage from logic-level transitions.
- Pre-charge registers and busses to remove leakage signatures from predictable data transitions.
- Add amplitude or temporal noise to reduce the attacker's signal-to-noise ratio.
- Employ security analysis software to detect attacks during the hardware design stage.
- Employ power-line conditioning and filtering to slow down the power monitoring analysis.
- Use displays with signal-attenuating materials to block electromagnetic radiation.
- Implement blinding techniques such as changing the algorithm's input or output to a random state.
- Implement hardware or software-based isolation mechanisms to compartmentalize sensitive operations and data.

- Design hardware components with built-in countermeasures against side-channel attacks, such as tamper-resistant enclosures, shielded cables, or secure elements that protect sensitive operations and data.
- Implement monitoring mechanisms to detect and respond to side-channel attacks in real-time, allowing for timely intervention and mitigation of potential security breaches.
- Adhere to established security standards and guidelines, such as those outlined by organizations like NIST, ISO, or IEC.
- Implement cryptographic algorithms and protocols in constant-time to eliminate timing variations that can be exploited by timing attacks.
- Design hardware components that offer built-in countermeasures such as dedicated hardware security modules (HSMs), secure enclaves, or physically unclonable functions (PUFs).
- Utilize multi-threading or multi-processing techniques to introduce variability in execution paths and timing.
- Employ algorithms that integrate randomization to increase the difficulty for attackers in correlating side-channel data with cryptographic operations.
- Employ cache partitioning or cache locking to isolate sensitive data from other processes.
- Integrate error detection and correction mechanisms to identify and mitigate induced faults.

Side-Channel Attack – Scenario

Assume that encrypted data are to be decrypted and displayed as plaintext inside a trusted zone. At the time of decryption in a cryptosystem, physical environmental factors, such as timing and power dissipation, acting on the components of a computer are recorded by an attacker. The attacker then analyzes this information to gain useful information for cryptanalysis.

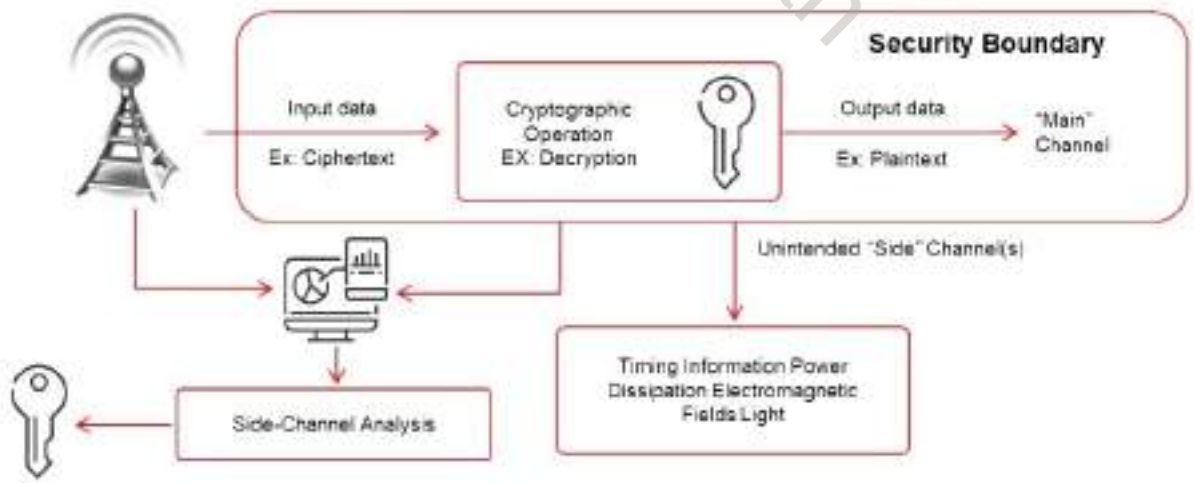
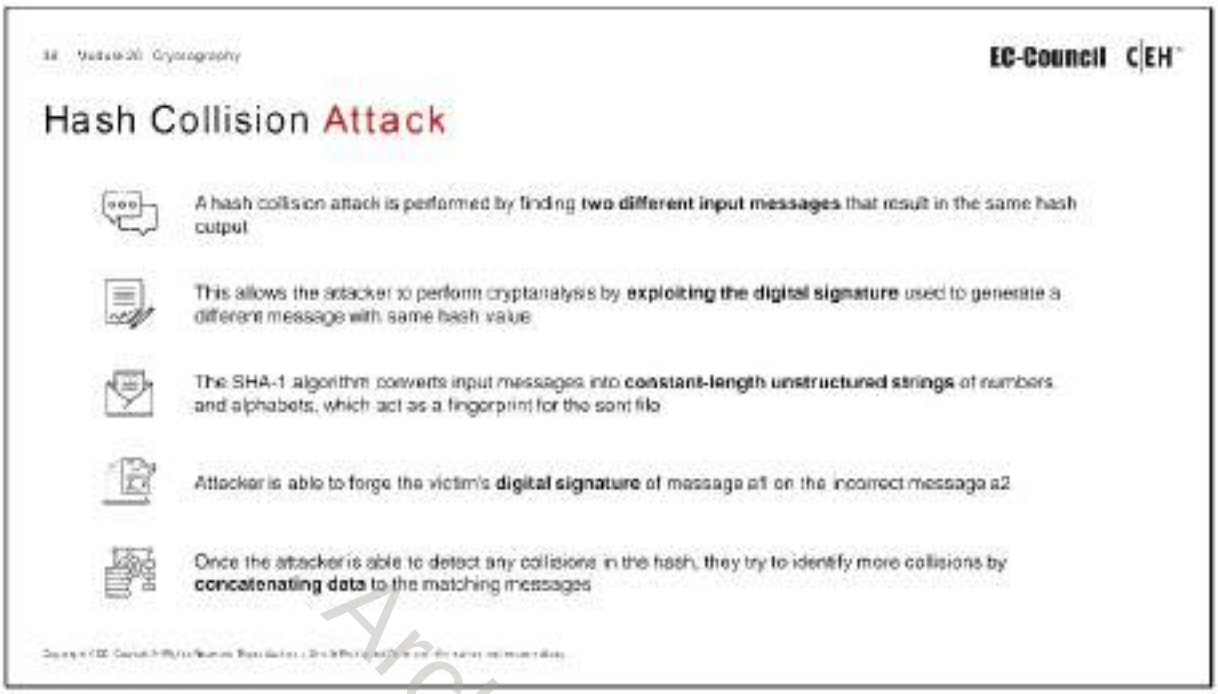


Figure 20.58: Side-Channel attack – scenario

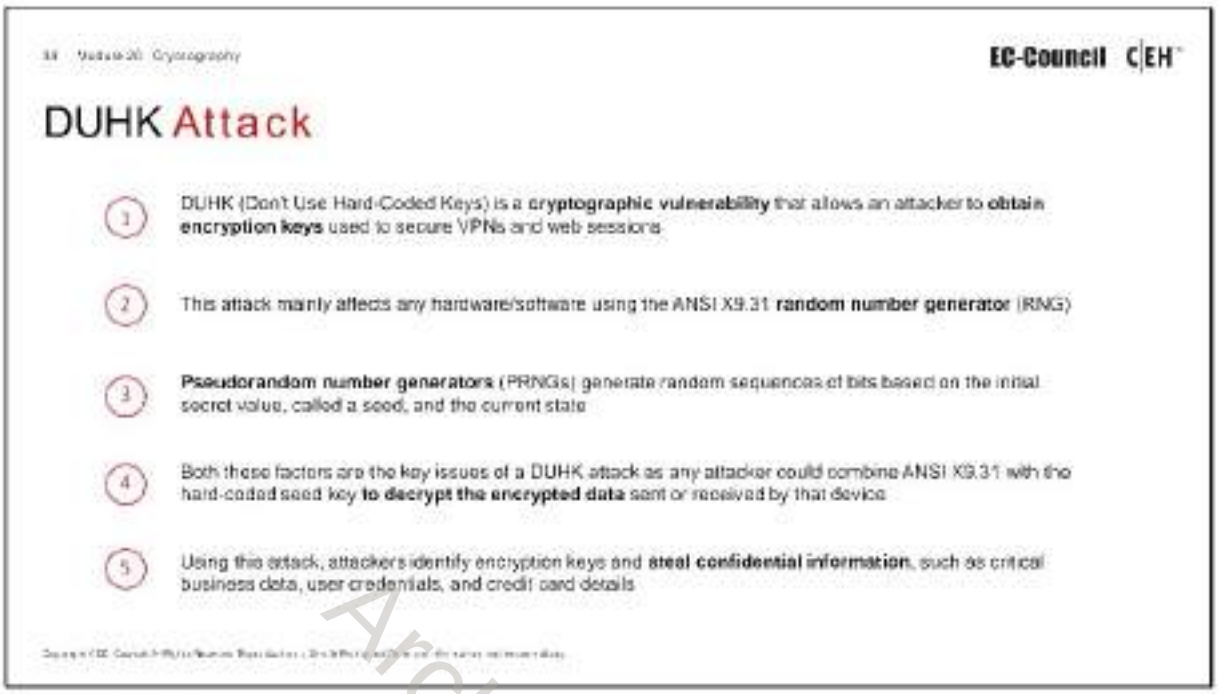


Hash Collision Attack

A hash collision attack is performed by finding two different input messages that result in the same hash output. For example, in a hash collision attack, "**hash(a1) = hash(a2)**", where a1 and a2 represent some random messages. Since the algorithm itself randomly selects these messages, attackers have no role in the content of these messages. This allows the attacker to perform cryptanalysis by exploiting the digital signature used to generate a different message with the same hash value.

One of the most popular hash functions is SHA-1, which is widely used as a digital signature algorithm. SHA-1 converts an input message into a constant length of unstructured strings of numbers and alphabets, which act as a fingerprint for the sent file. Therefore, the attacker tries to identify similar hashed output to get the digital signatures of the victim. This allows the attacker to forge the victim's digital signature of message a1 on message a2.

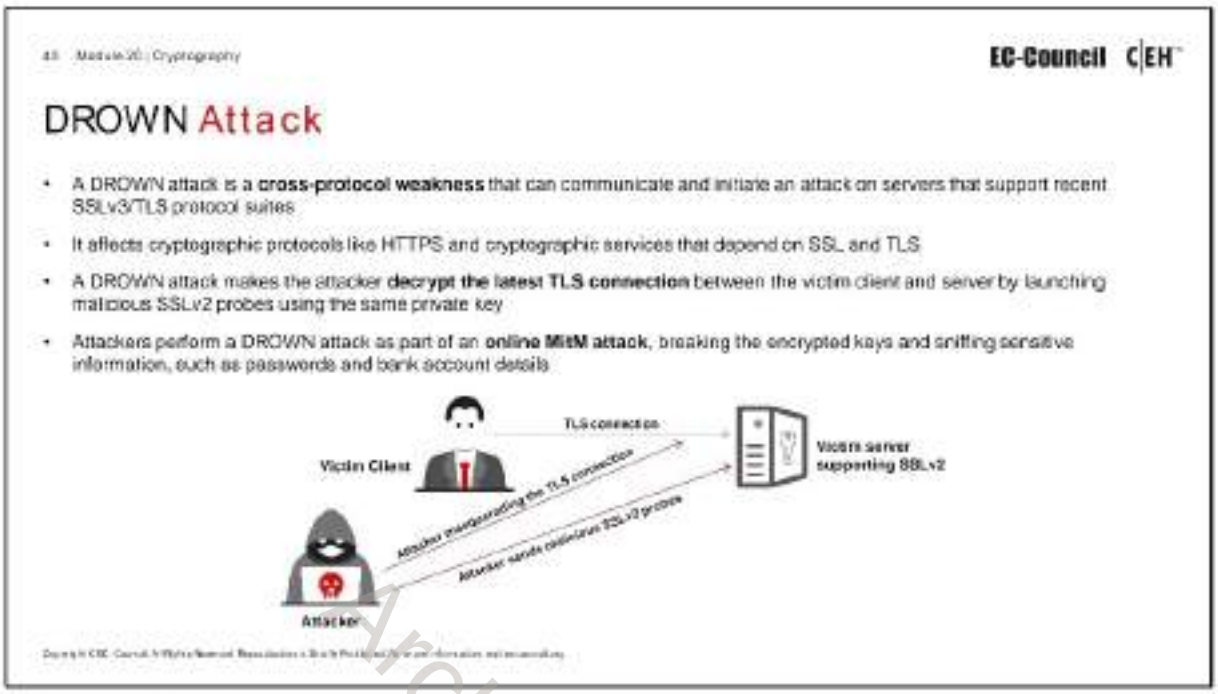
Once the attacker detects a collision in the hash, he/she can identify more collisions by concatenating the data to matching messages.



DUHK Attack

Don't Use Hard-Coded Keys (DUHK) is a cryptographic vulnerability that allows attackers to obtain encryption keys used to secure VPNs and web sessions. This attack mainly affects any hardware/software using the ANSI X9.31 Random Number Generator (RNG). Pseudorandom number generators (PRNGs) generate random sequences of bits based on the initial secret value, called seed, and the current state. The PRNG algorithm generates cryptographic keys that are used to establish a secure communication channel over the VPN. In some cases, the seed key is hardcoded into the implementation. Both the factors are key issues of the DUHK attack, as any attacker can combine ANSI X9.31 with the hard-coded seed key to decrypt the encrypted data sent or received by that device.

Man-in-the-middle attackers use the DUHK attack to learn the seed value, observe the current session, and obtain the current state value. Using this attack, attackers can identify encryption keys and steal confidential information such as critical business data, user credentials, and credit card details.



DROWN Attack

Decrypting RSA with Obsolete and Weakened eNcryption (DROWN) is a grave vulnerability that can affect important cryptographic protocols such as HTTPS and other cryptographic services that depend on SSL and TSL. The DROWN attack is a cross-protocol weakness that can communicate and initiate an attack on servers supporting recent SSLv3/TLS protocol suites. It is a new form of cross-protocol Bleichenbacher padding oracle attack.

The server is critically vulnerable to the DROWN attack if

- The server permits SSLv2 connection, which is mostly caused by a misconfiguration or incorrect default settings.
- The same private key certificate is used on a different server that allows SSLv2 connection, and it also makes the TLS server vulnerable, as the SSLv2 server can leak the key information.

The DROWN attack allows the attacker to decrypt the latest TLS connection between the victim client and the server by launching malicious SSLv2 probes using the same private key. Using this attack, the attacker can also force the victim client and server to use the RSA key exchange. Thus, the attacker can disrupt connections among the latest browsers and servers that favor the use of latest techniques, i.e., perfect-forward-secret key exchange, such as DHE and ECDH.

Attackers perform the DROWN attack as part of an online man-in-the-middle (MITM) attack, breaking encrypted keys, sniffing or stealing sensitive information such as passwords and bank account details, and accessing personal emails or messages. By performing this attack, the attacker can also masquerade as a secure website and thus seize or change the website contents.

Figure 20.59: DROWN attack

Rainbow Table Attack

A rainbow table attack is a type of cryptography attack whereby an attacker uses a rainbow table for reversing cryptographic hash functions. A rainbow table attack uses the cryptanalytic time-memory trade-off technique, which is less time consuming than other techniques. It uses already calculated information stored in memory for encryption. In the rainbow table attack, the attacker creates a table of all the possible passwords and their respective hash values, called a rainbow table, in advance.

A rainbow table contains word lists such as dictionary files and brute-force lists and their hash values. It is a lookup table particularly used for recovering a plaintext password from a ciphertext. The attacker uses this table to look for the password and tries to recover it from password hashes.

An attacker computes the hash for a list of possible passwords and compares it with the pre-computed hash table (rainbow table). If a match is found, then he/she can crack the password. It is easy to recover passwords by comparing the captured password hashes with pre-computed tables.

Related-Key Attack

Attackers launch a related-key attack by exploiting the mathematical relationship between the keys in a cipher and gain access to encryption and decryption functions. The attacker's motive for launching this attack is to find the related private/secret keys. To implement this attack, the attacker monitors the cipher operation where key values are initially unknown; then, the attacker captures the relation between those keys after a thorough examination. For instance, the attacker observes and finds that the last 80 bits of the keys are same at all times, but he/she does not initially know about these bits.

The failure in the WEP cryptogram, i.e., when used in wireless networks, is the best example of this attack. In this attack, each AP and user interface device uses the same key. The encryption used in WEP is a stream cipher known as RC4; it is important to note that the same keys should not be repeated in the stream cipher. To avoid this, WEP integrates a 24-bit initial vector (IV) in every packet transferred. The RC4 key for that particular packet is the IV associated with the WEP key. WEP keys need to be changed manually, however, this is rarely done. Hence, the attacker notices that the keys used for encryption are often the same. This drawback poses various risks on WEP, especially using the birthday paradox, because for every 4096 packets, two parties will share the same IV and hence the same RC4 key. This simple form of encryption allows the attacker to leverage weak RC4 keys, which ultimately forces the recovery of the WEP key.

Padding Oracle Attack

In a padding oracle attack, the attackers exploit the padding validation of an encrypted message to decipher the ciphertext. Such an attack is also known as a Vaudenay attack. In many cryptographic algorithms based on a block cipher, the messages are padded with additional random bits so that the length of the last block is of the required size. Padding oracle is a function of such encryption that verifies if a message was correctly padded. This attack is mainly performed on algorithms that operate in the CBC (Cipher Block Chaining) mode.

In this attack, the server (oracle) reveals information about whether the padding of an encrypted message was correctly done. In some cases, this information allows attackers to decrypt and optionally encrypt messages using the server's key (oracle's key) without having access to the corresponding encryption key.

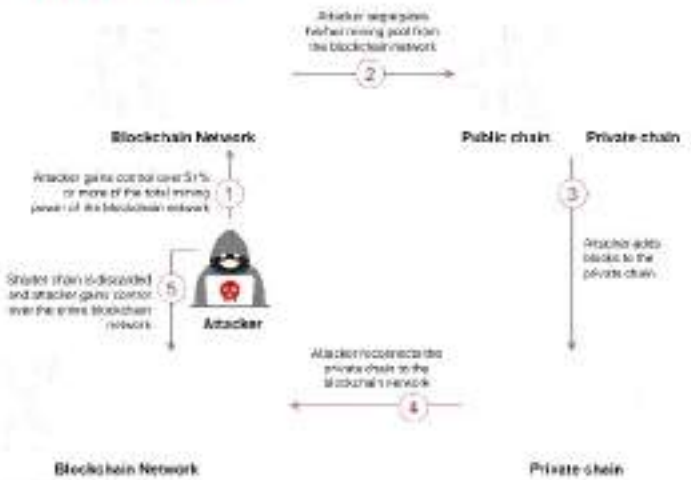
For example, consider a scenario with a standard implementation of CBC decryption. The server decrypts all the ciphertext blocks, verifies the padding, removes all the additional padding done during the encryption process, and then returns the original message to the application or user. If, for example, the server is unable to decrypt the message due to a padding error and returns an error message "Decryption failure: Invalid Padding" instead of a generic message "Decryption failed", this information can be exploited by an attacker. The attacker can use the server as a padding oracle to decipher the encrypted messages.

43 Module 20: Cryptography

EC-Council C|EH

Attacks on Blockchain: 51% Attack

- 51% attack, also known as majority attack, occurs when an attacker or group of attackers gains control of **more than 50% of the computational power** (hash rate) or staking power in a blockchain network
- This level of control allows them to manipulate the blockchain by conducting **double-spending attacks**, **denial-of-service** (DoS) attacks, **transaction reversals**, and other malicious activities

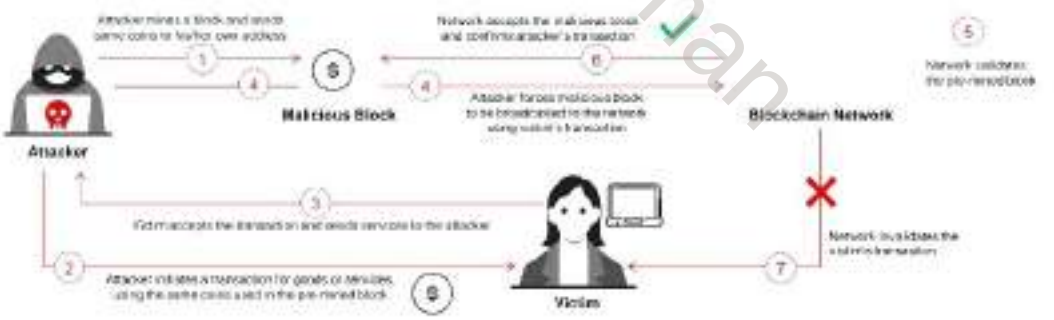


43 Module 20: Cryptography

EC-Council C|EH

Attacks on Blockchain: Finney Attack

- Finney attack is a type of blockchain attack that involves an attacker leveraging the time **delays between the broadcasting and the confirmation of transactions** in cryptocurrency networks to **reverse the transactions** before they are confirmed
- Attackers perform this attack to **double-spend the cryptocurrency**, effectively getting goods or services for free while retaining their coins

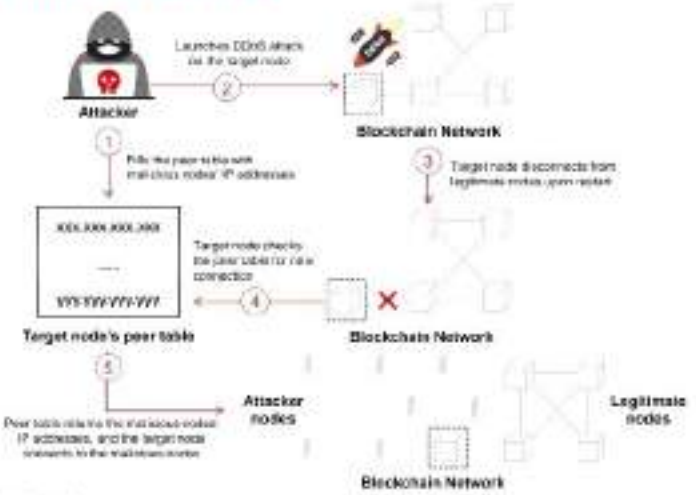


44 Module 20 | Cryptography

EC-Council C|EH

Attacks on Blockchain: Eclipse Attack

- Eclipse attack is a type of blockchain attack where the attacker **isolates a target node** from the rest of the network **by surrounding it with malicious nodes**, effectively controlling the node's view of the blockchain
- This attack allows the attacker to exploit the target node for various malicious purposes such as **disrupting transaction processing**, split mining power, facilitating **double-spending attacks** and so on

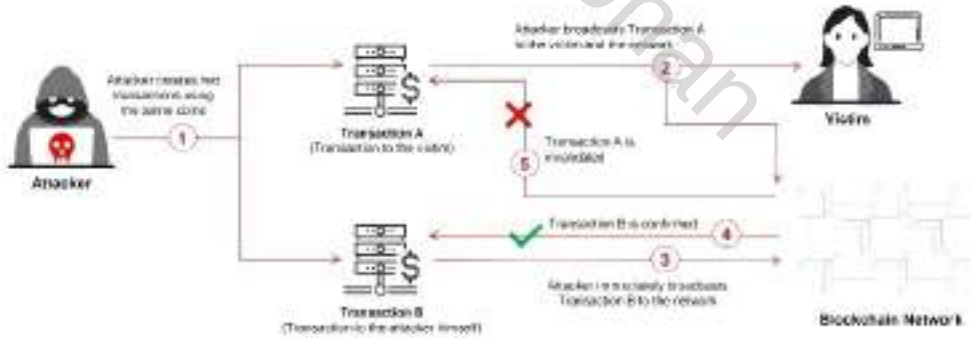


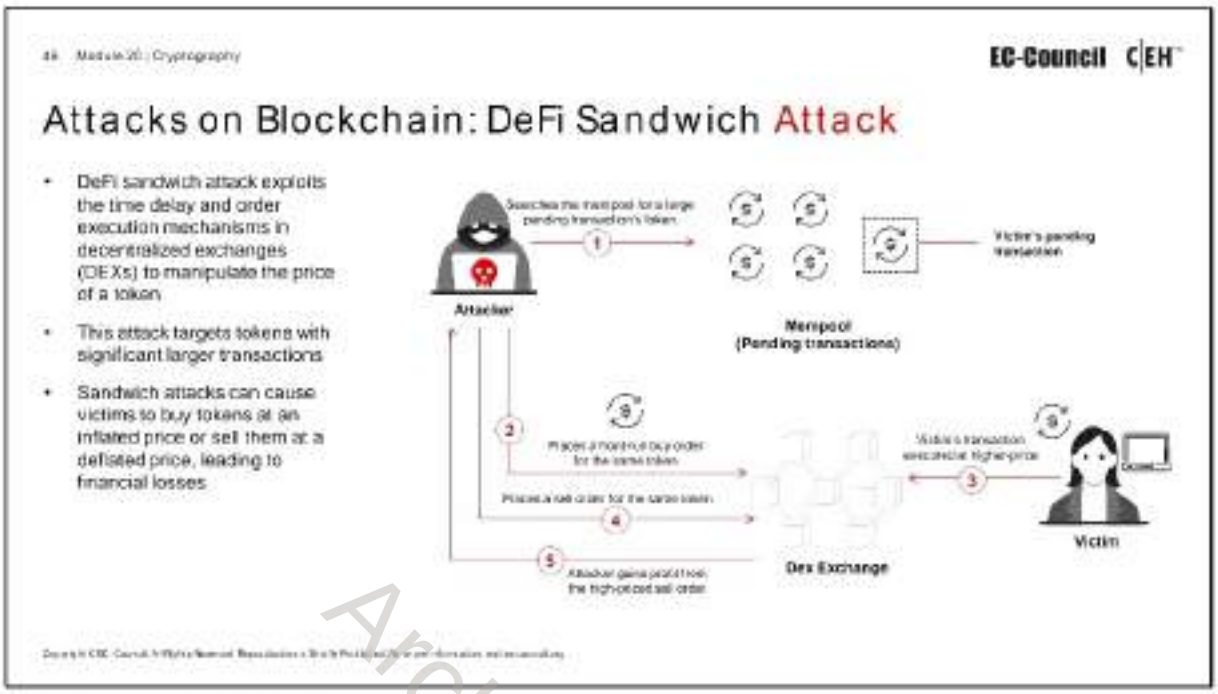
45 Module 20 | Cryptography

EC-Council C|EH

Attacks on Blockchain: Race Attack

- Race attack is a double-spending attack that exploits the **delay in transaction confirmation** in blockchain networks to obtain goods or services without actually paying for them
- Unlike the Finney attack, a race attack **does not require the attacker to pre-mine blocks** to reverse the victim's transaction but relies on the attacker's **ability to broadcast transactions quickly** and exploit the network's latency





Attacks on Blockchain

Attacks on Blockchain refer to various malicious activities aimed at exploiting vulnerabilities within a blockchain network. These attacks can take multiple forms, such as 51% attacks, in which a single entity gains majority control over the network's computing power, allowing it to double spend coins and reverse transactions.

Attackers can perform these actions to steal funds, disrupt services, or undermine trust in the system. The impact of blockchain attacks can be severe, leading to significant financial losses, reduced confidence in the blockchain technology, compromised data integrity, and potential legal and regulatory repercussions for affected entities. Therefore, attackers can perform different types of blockchain attacks, as discussed below:

- **51% Attack**

  A 51% attack, also known as a majority attack, occurs when an attacker or group of attackers gains control of more than 50% of the computational power (hash rate) or stacking power in a blockchain network. This level of control allows them to manipulate the blockchain by conducting double-spending attacks, denial-of-service (DoS) attacks, transaction reversals, and other malicious activities. Attackers often target less-secure or smaller blockchain networks, where achieving majority control is easier.

  The steps performed by an attacker to launch a 51% attack on a target blockchain network are as follows:

  - **Step 1**: The attacker obtains control of 51% or more of the total mining power of the blockchain network. This can be achieved by renting additional mining power, purchasing sophisticated hardware resources, or persuading a significant number of miners to join a pool managed by an attacker.

- **Step 2:** The attacker isolates the mining pool from the network and continues mining on a private blockchain, keeping their progress hidden from the main network.
- **Step 3:** Attacker extends their private blockchain by adding more blocks to it.
- **Step 4:** Once the private chain becomes longer than the public chain, the attacker releases it into the main network.
- **Step 5:** The main network recognizes the longer chain as a valid chain, which invalidates the transaction in the shorter public chain, allowing the attacker to maintain control over all transactions in the entire network.

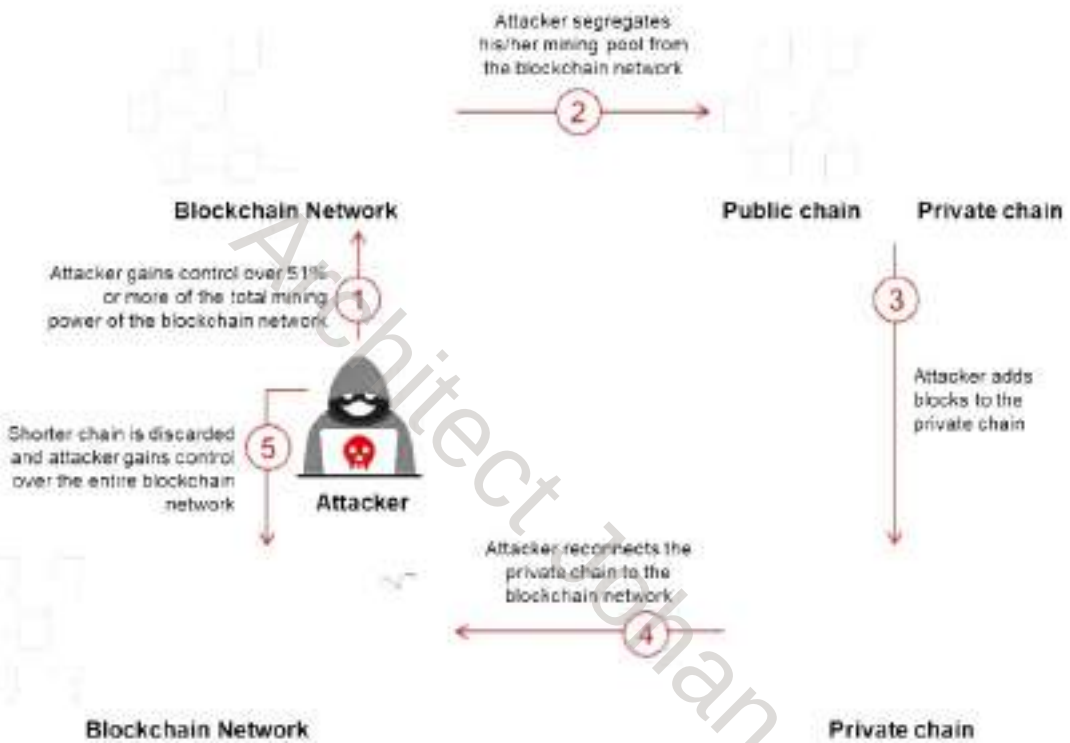


Figure 20.60: Illustration of 51% attack

- **Finney Attack**

A Finney attack is a type of blockchain attack that involves an attacker leveraging the time delays between the broadcasting and confirmation of transactions in cryptocurrency networks to reverse the transactions before they are confirmed. This type of attack primarily targets merchants and service providers who accept cryptocurrency payments, particularly those who do not wait for multiple confirmations before finalizing a transaction. Attackers perform this attack to double-spend cryptocurrency, effectively obtaining goods or services for free while retaining their coins.

The following are the various steps an attacker performs in a Finney attack:

- **Step 1:** The attacker premines a block that includes a transaction sending coins to themselves or another address they control, but does not immediately broadcast it to the network.
- **Step 2:** The attacker initiates a transaction with the victim for goods or services and sends them the same coins used in the pre-mined block.
- **Step 3:** Immediately after the victim accepts the transaction, the attacker broadcasts a predetermined block to the network.
- **Step 4:** Nodes in the network validate the predetermined block and its transactions. As this block contains the same coins sent to the attacker's address, it conflicts with the transaction sent to the victim.
- **Step 5:** If the pre-mined block is accepted before the victim's transaction is confirmed, the attacker's transaction becomes valid, thus reversing the victim's transaction. The attacker retains both the goods/services and coins, causing the victim to lose out.

Figure 20.61: Illustration of Finney attack

- **Eclipse Attack**

An Eclipse attack is a type of blockchain attack in which an attacker isolates a target node from the rest of the network by surrounding it with malicious nodes, thereby effectively controlling the node's view of the blockchain. This type of attack primarily targets the nodes that accept incoming connections. Isolating the target node from the blockchain network allows the attacker to manipulate the target node's perception of the network, which can be exploited for various malicious purposes such as disrupting transaction processing, splitting mining power, and facilitating double-spending attacks.

Various steps performed by an attacker in an Eclipse attack are given below:

- **Step 1**: The attacker fills the target node's peer tables with its own IP addresses so that the target node connects only to the attacker nodes.
- **Step 2**: Upon manipulating the peer tables, the attacker forces the target node to restart, typically by launching a DoS attack.

- **Step 3**: During the restart, the node disconnects from all legitimate nodes in the network.
- **Step 4**: Following the restart, the targeted node looks at its peer tables for new connections and unknowingly connects to the attacker nodes.
- **Step 5**: The target node is isolated from legitimate network participants and its inbound and outbound connections are redirected to the attacker nodes.

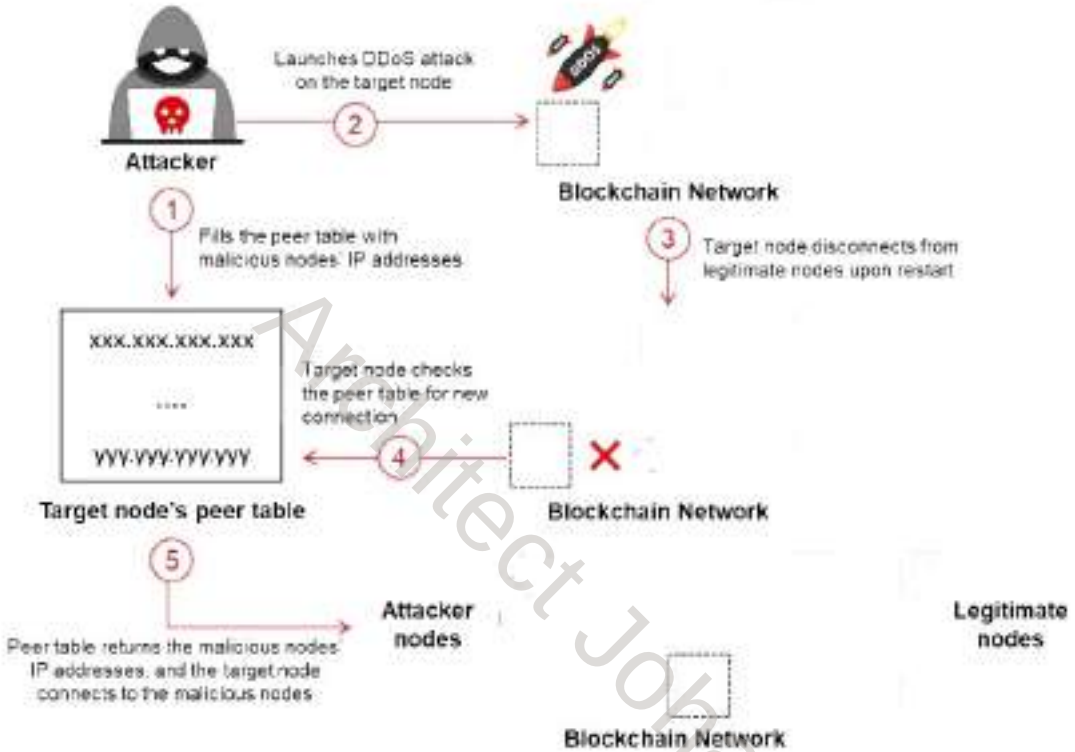


Figure 20.62: Illustration of Eclipse attack

- **Race Attack**

A race attack is a double-spending attack that exploits the delay in transaction confirmation in blockchain networks to obtain goods or services without actually paying for them and effectively spends the same coin twice. This attack is similar to the Finney attack, but does not require the attacker to pre-mine blocks to reverse the victim's transaction. It relies on an attacker's ability to broadcast transactions quickly and exploit a network's latency. Because mempool contains unconfirmed transactions, they can be manipulated by broadcasting multiple conflicting transactions in quick succession. Race attacks primarily target victims who accept zero-confirmation transactions.

Attackers perform the following steps to launch a race attack on a target service provider:

- **Step 1**: The attacker creates two transactions using the same coin: one for the victim (Transaction A) and one for his own address (Transaction B).

- **Step 2:** The attacker broadcasts Transaction A to the victim and the network, ensuring that the victim sees this transaction.

  (The victim generally accepts the transaction and provides goods or services based on a zero-confirmation transaction.)

- **Step 3:** Immediately, the attacker broadcasts Transaction B to the network in the hope that Transaction B will be confirmed before Transaction A.

  [Here, both transactions (Transaction A and Transaction B) will be in a race to be confirmed by the network. As both transactions use the same inputs, only one can be included in the blockchain.]

- **Step 4:** If Transaction B is confirmed first, Transaction A is invalidated, allowing the attacker to retain both goods/services and cryptocurrencies.

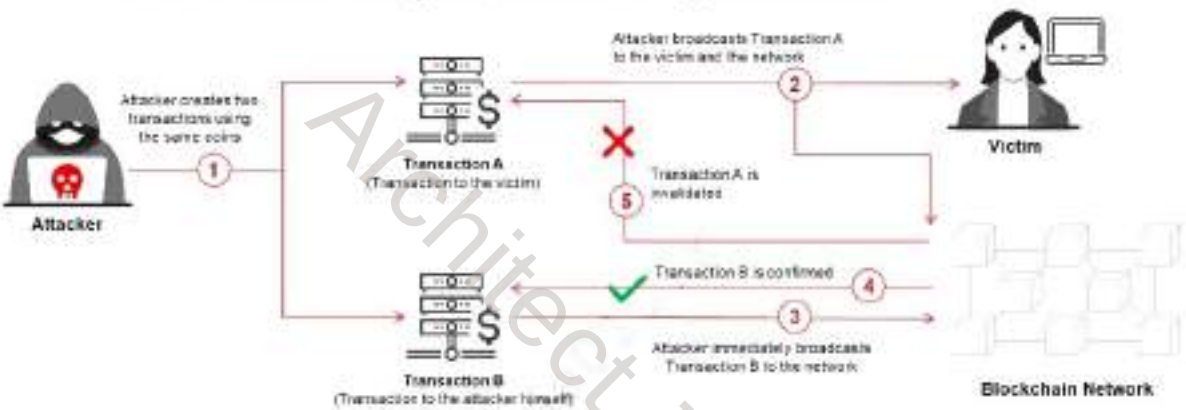


Figure 20.63: Race attack

- **DeFi Sandwich Attack**

  A decentralized finance (DeFi) sandwich attack is a blockchain attack targeting decentralized exchanges (DEXs) and automated market makers (AMMs) to manipulate market dynamics. In this attack, the attacker exploits the time delay and order execution mechanisms in the DEXs to manipulate the price of a token to their advantage. This attack targets tokens with significantly larger transactions. Sandwich attacks can cause victims to buy tokens at inflated prices or sell them at deflated prices, leading to financial losses.

  The following are the various steps an attacker performs in a DeFi sandwich attack:

  - **Step 1:** The attacker searches the mempool and finds a large pending transaction that is likely to raise the market price of the token.
  - **Step 2:** The attacker places a buy order for the same token immediately before the victim's transaction to increase the price of the token.
  - **Step 3:** Next, the victim's transaction is executed, buying the token at an inflated price caused by the attacker's front-running buy order.

- **Step 4:** Immediately after the victim's transaction is executed, the attacker places a sell order for the same token by exploiting the price increase caused by the victim's buy order.
- **Step 5:** The attacker profits by buying the token at a lower price before the victim's transaction and selling it at a higher price.

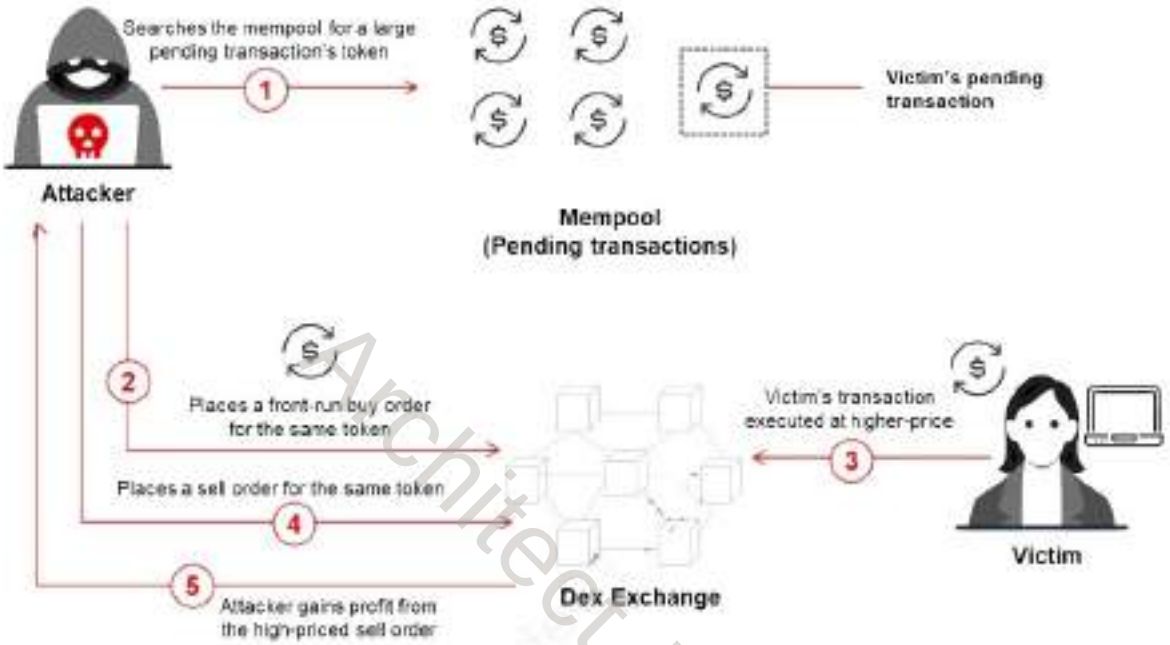


Figure 20.64: Illustration of DeFi sandwich attack

Quantum Computing Risks

The following are the key risks posed by quantum computing to current cryptographic systems:

- **Breaking Classical Cryptographic Systems**

  Quantum-computing algorithms can weaken the security of both symmetric and asymmetric cryptographic algorithms. For example, Shor's algorithm can quickly factorize large numbers and solve discrete logarithms, whereas Grover's algorithm can reduce the effective strength of the encryption keys. This threatens the security of widely used systems such as RSA, DSA, and ECC. Consequently, public-key infrastructure (PKIs) and digital signatures that rely on these algorithms can become vulnerable, compromising secure communication and digital identities.

- **Cryptographic Transition Challenges**

  Developing and implementing quantum-resistant algorithms is a proactive measure against quantum threats. However, the deployment of quantum-resistant algorithms across systems and networks is complex and requires significant time, effort, and resources. Vulnerability may persist during the transition period, leading to security breaches.

- **Data Harvesting for Future Decryption**

  Adversaries may collect and archive sensitive data encrypted with current cryptographic methods and plan to decrypt them later using quantum computers. This jeopardizes the long-term confidentiality of crucial information, especially for data that must remain secure over extended periods, such as government communications, financial records, and personal data.

- **Decrypting Secure Communications**

  For key exchange and authentication, several secure communication protocols (SSL/TLS and VPNs) depend on public-key cryptography. However, public-key cryptography is particularly vulnerable to quantum attacks, potentially compromising the security of these communication channels.

- **Threat of Quantum-Driven Exploits**

  Quantum computers can be used to develop new algorithms or techniques to undermine current security measures. These include advanced forms of side-channel attacks and new cryptographic methods.

- **Threat of Quantum-enhanced Eavesdropping**

  Quantum sensors can intercept encrypted communication with unprecedented precision, making it easier for attackers to capture and analyze data transmitted over secure channels. Quantum-enhanced eavesdropping can effectively undermine encryption methods currently used to protect the confidentiality of sensitive communications.

- **Undermining Blockchain Security**

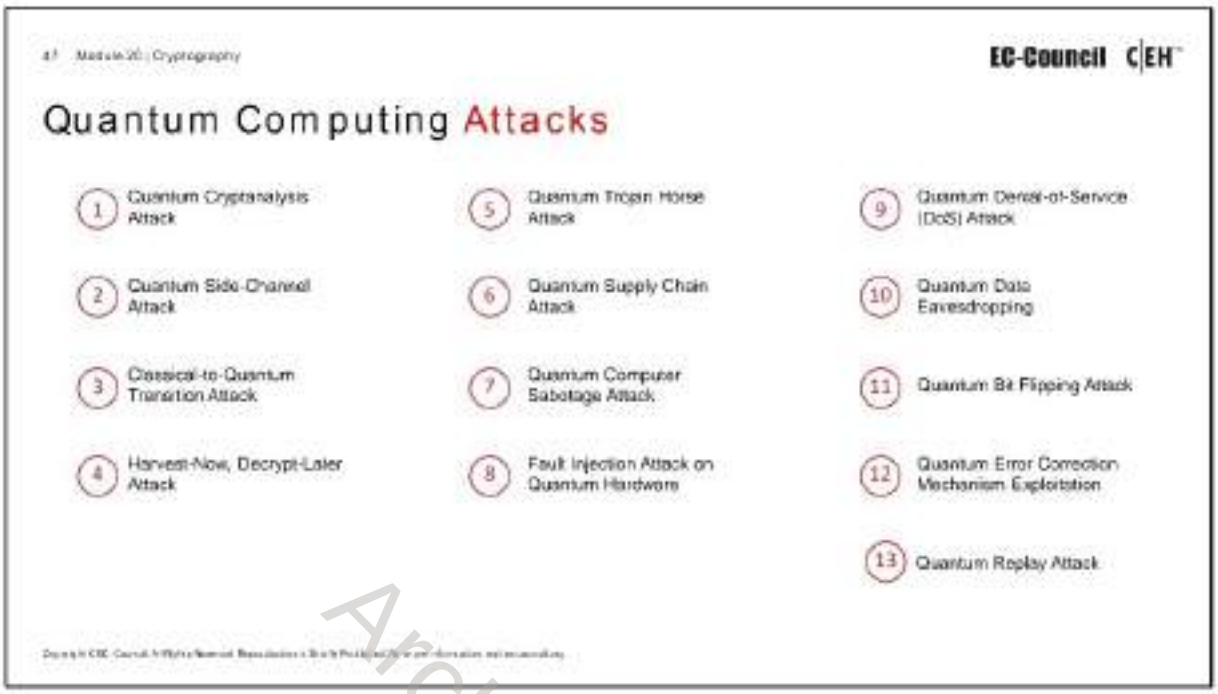
  Quantum computing can be used to derive private keys from public keys in blockchain systems and break the cryptographic hashes and digital signatures used in blockchain networks, potentially allowing attackers to alter blockchain records or perform double-spending attacks.

- **Threat to Secure Authentication Systems**

  Quantum computers can break public-key cryptographic algorithms that form the basis of secure authentication mechanisms, resulting in unauthorized access to systems and data. This puts the security of the authentication systems, including multifactor authentication and secure access protocols, at risk.

- **Risks of Quantum Malware**

  Quantum malware leverages quantum-computing algorithms to predict or reconstruct keys, bypass undetected encryption, and decrypt intercepted encrypted communications in real-time, allowing attackers to exfiltrate sensitive information as it is transmitted.



Quantum Computing Attacks

Various potential quantum-computing attacks and their implications for the security of modern cryptographic systems are discussed below:

1. **Quantum Cryptanalysis Attack**

   Quantum cryptanalysis attacks leverage quantum-computing techniques to compromise cryptographic systems that can withstand classical attacks. Quantum algorithms, such as Shor's and Grover's, allow attackers to break current encryption methods, jeopardizing the confidentiality and integrity of sensitive information. In addition, these algorithms allow attackers to forge digital signatures, thereby undermining the authenticity of electronic communications and transactions.

2. **Quantum Side-Channel Attack**

   Quantum side-channel attacks exploit information leakage from the physical implementation of quantum cryptographic systems to gather information from quantum computers without directly attacking the algorithms. By analyzing indirect vulnerabilities such as quantum noise, error rates, timing, power consumption, and electromagnetic emissions during quantum computations or key exchanges, attackers can infer sensitive information, including cryptographic keys.

3. **Classical-to-Quantum Transition Attack**

   Classical-to-quantum transition attacks exploit vulnerabilities that arise during the transition from classical cryptographic systems to quantum-resistant systems. During this period, hybrid systems that combine classical and quantum cryptography may exhibit weaknesses or interoperability issues. Attackers target these transitional systems to

compromise security, intercept communication, and extract sensitive information before robust quantum-resistant protocols are fully implemented and standardized.

4. **Harvest-Now, Decrypt-Later Attack**

   The Harvest-Now, Decrypt-Later attack involves intercepting and storing encrypted data today with the intention of breaking the encryption in the future once quantum computers become powerful enough to defeat current encryption algorithms. Attackers capture large volumes of sensitive encrypted data from the Internet or secure communications, expecting that future quantum computing capabilities will allow them to decrypt this information, thereby compromising long-term confidentiality and security.

5. **Quantum Trojan Horse Attack**

   Quantum Trojan horse attacks involve embedding malicious quantum devices or components in a quantum cryptographic system. These malicious elements can covertly gather and transmit sensitive information to attackers. By exploiting these hidden vulnerabilities, attackers can bypass the security of quantum cryptographic protocols, potentially compromising the integrity and confidentiality of the entire system without directly breaking quantum algorithms.

6. **Quantum Supply Chain Attack**

   Quantum supply chain attacks target the quantum-computing hardware and software supply chains. This type of attack involves tampering with components, software, or firmware at any point in the supply chain before deployment in a quantum system. The goal is to introduce vulnerabilities or backdoors that can be exploited later to compromise the security and integrity of a quantum system. These include the addition of malicious quantum circuits or states to quantum communication systems or algorithms to create exploitable weaknesses.

7. **Quantum-Computer Sabotage Attack**

   Quantum-computer sabotage attacks involve direct physical or cyberactions aimed at disrupting or destroying quantum-computer operations. This can include physically damaging the hardware to cause malfunctions, potentially affecting critical applications, and research that relies on advanced computational capabilities. Such attacks can significantly hinder the progress and deployment of quantum-computing technologies.

8. **Fault-Injection Attack on Quantum Hardware**

   Fault-injection attacks on quantum hardware involve deliberately embedding errors or faults into a quantum system to interrupt its workflow and compromise security. This type of attack is performed by manipulating environmental conditions, such as temperature and electromagnetic fields, or exploiting vulnerabilities in the quantum hardware design. By inducing faults, attackers can alter quantum states or computational processes, potentially leading to incorrect results or revealing sensitive information.

9. **Quantum Denial-of-Service (DoS) Attack**

   Quantum denial-of-service (DoS) attacks involve flooding a quantum computer or quantum network with a high volume of requests, overloading processing capabilities, or exploiting specific vulnerabilities in quantum algorithms and hardware. The goal is to render a quantum system inaccessible or to significantly degrade its performance, thereby preventing legitimate users from executing quantum computations.

10. **Quantum Data Eavesdropping**

    Quantum data eavesdropping attacks involve intercepting and analyzing the transmission of quantum data to gain unauthorized access to sensitive information. In this type of attack, adversaries exploit vulnerabilities in quantum communication channels such as those used in quantum key distribution (QKD). By measuring the transmitted quantum states, attackers can attempt to reconstruct the data or keys being exchanged, thereby posing a significant threat to the security of quantum communication.

11. **Quantum Bit-Flipping Attack**

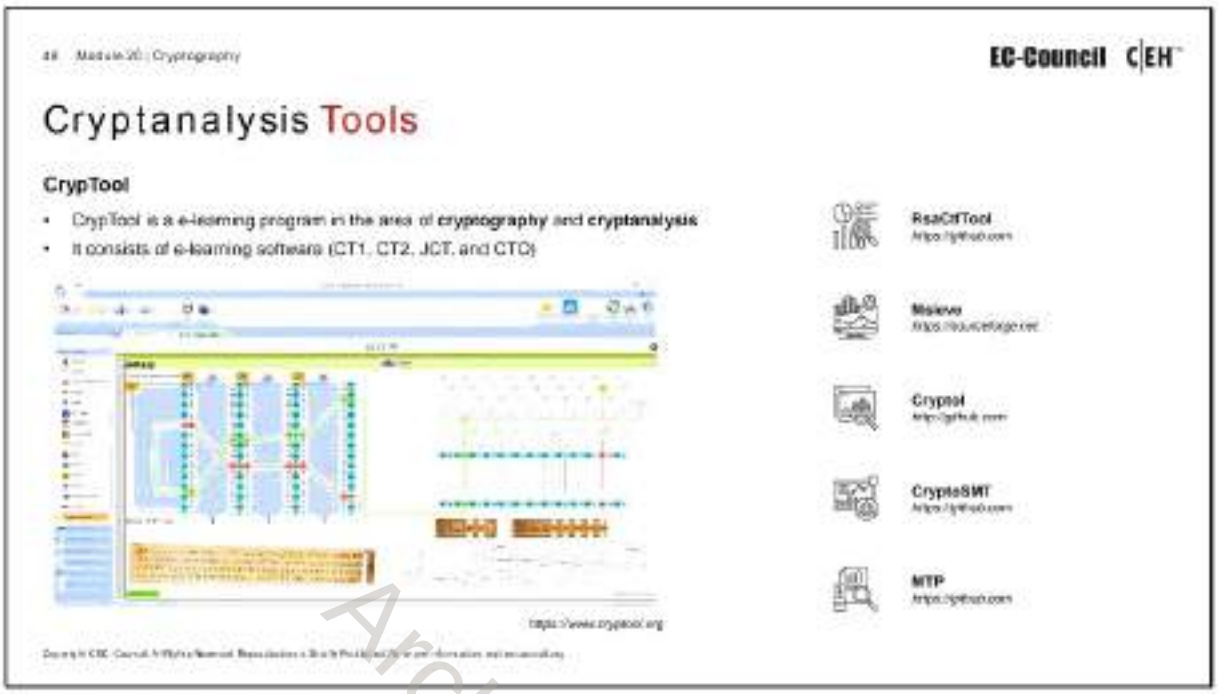
    Quantum bit-flipping attacks involve deliberately changing the state of qubits within a quantum system to disrupt its normal operation and compromise the integrity of the computations. In this attack, an adversary manipulates the qubits by flipping their states from 0 to 1, or vice versa. This can lead to incorrect results in quantum computations or can corrupt the information being processed, thereby posing a significant threat to the reliability and security of quantum systems.

12. **Quantum Error Correction Mechanism Exploitation**

    Quantum error correction mechanism exploitation involves attackers exploiting vulnerabilities in error-correction protocols in quantum computing systems to introduce undetected errors that compromise the reliability of quantum computations and potentially extract sensitive information. This can be achieved by sending tailored error patterns that error correction algorithms cannot effectively handle, leading to computational failures.

13. **Quantum Replay Attack**

    Quantum Replay Attack involves the interception and retransmission of quantum communication signals to deceive the receiver and accept them as legitimate. In such attacks, adversaries capture quantum data or quantum key distribution (QKD) transmissions and replay them to trick the system into thinking that the data are original and valid. This can lead to unauthorized access, data breaches, or compromises of the cryptographic keys.



Cryptanalysis Tools

Attackers use cryptanalysis tools to analyze and break ciphers. Some cryptanalysis tools are discussed as follows.

- **CrypTool**

  Source: *https://www.cryptool.org*

  The CrypTool offers a range of software tools and resources designed to carry out cryptanalysis using various cryptographic techniques and methods. It consists of various in-built projects such as CT1, CT2, JCT, and CTO.

  - **CrypTool 1 (CT1)** – It is written in C++ and is a Windows program. It supports classical and modern cryptographic algorithms (encryption and decryption, key generation, secure passwords, authentication, secure protocols, etc.). It is used to perform cryptanalysis of several algorithms (Vigenère, RSA, AES, etc.)
  - **CrypTool 2 (CT2)** – It supports visual programming GUI and execution of cascades of cryptographic procedures. It is also compatible for Windows OS.
  - **JCrypTool (JCT)** – It allows comprehensive cryptographic experimentation on Linux, macOS, and Windows. It also allows users to develop and extend its platform in various ways with their own crypto plug-ins.
  - **CrypTool-Online (CTO)** – It runs in a browser and provides a variety of encryption methods and analysis tools.

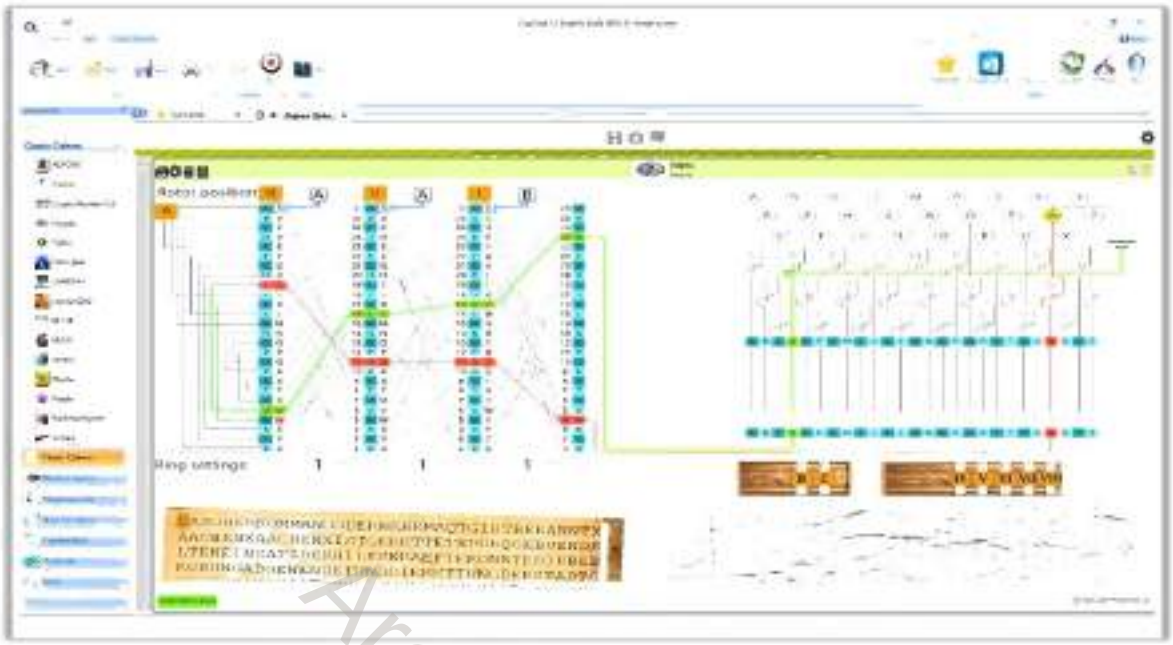


Figure 20.65: Screenshot of CrypTool

Some additional cryptanalysis tools are as follows:

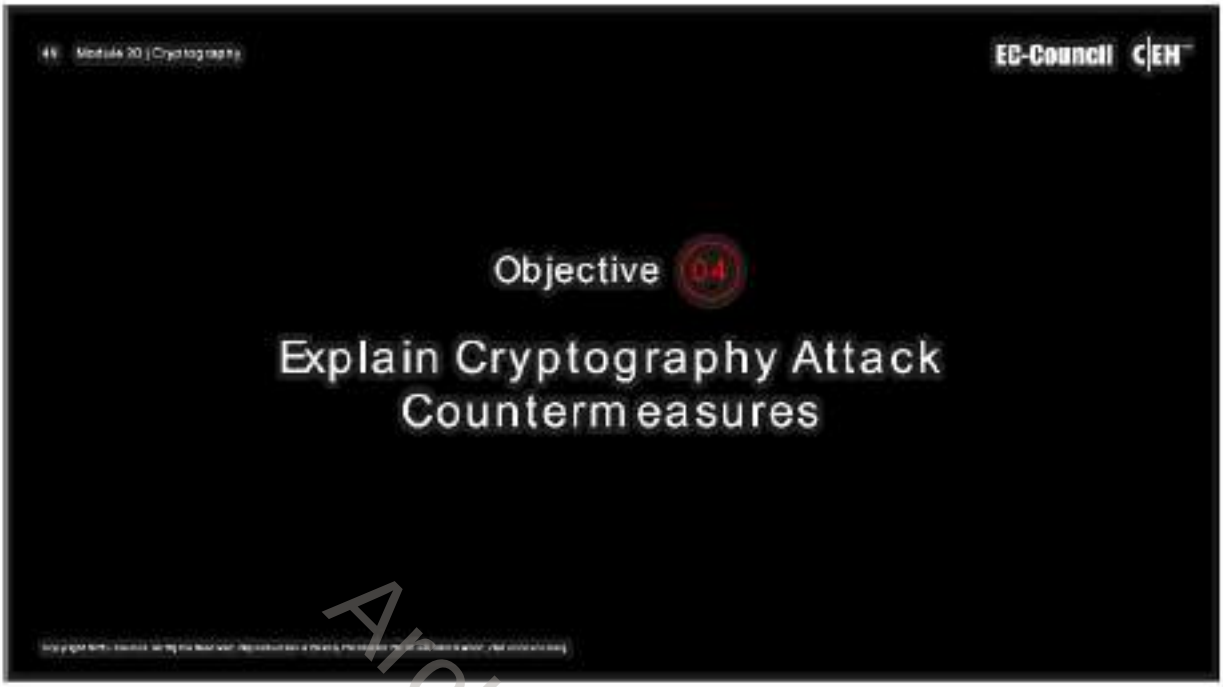
- RsaCtfTool (*https://github.com*)
- Msieve (*https://sourceforge.net*)
- Cryptol (*http://cryptol.net*)
- CryptoSMT (*https://github.com*)
- MTP (*https://github.com*)

Online MD5 Decryption Tools

Some online MD5 decryption tools that can be used to decrypt the MD5 hash value to discover the original message are as follows:

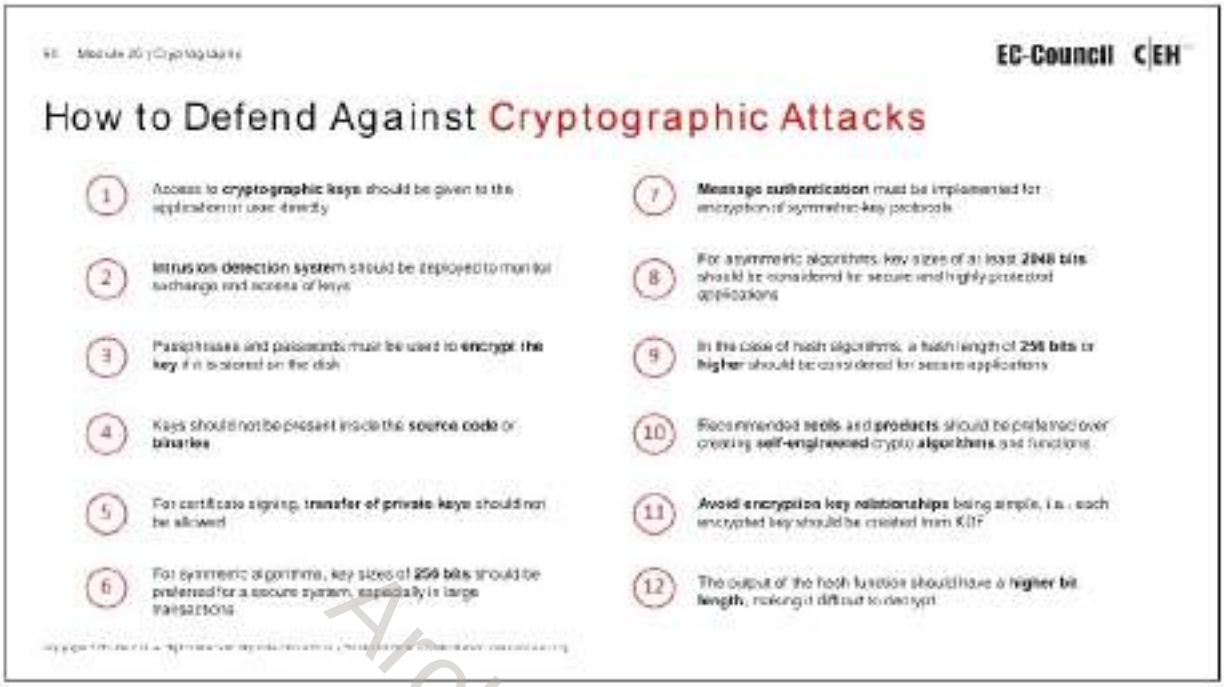
- MD5 Decrypter (*https://www.dcode.fr*)
- MD5 Decrypt (*https://iotools.cloud/tool*)
- Md5 Encrypt & Decrypt (*https://md5decrypt.net*)
- MD5Hashing.net (*https://md5hashing.net*)
- MD5 Encrypt/Decrypt (*https://10015.io*)
- MD5 Decryption (*https://www.md5online.org*)
- MD5Decrypter.com (*https://www.md5decrypter.com*)
- Online Hash Crack (*https://www.onlinehashcrack.com*)
- Md5.My-Addr.com (*https://md5.my-addr.com*)

- Cmd5 (*https://www.cmd5.org*)
- Hashes (*https://hashes.com*)
- Online MD5 Hashed Validator (*https://www.javainuse.com*)
- MD5 Hash Decode (*https://md5.web-max.ca*)
- MD5 Decrypt (*https://allinone.tools*)
- GettHIT.com (https://www.getthit.com)



Cryptography Attack Countermeasures

Attackers use various cryptanalysis methods and techniques to break cryptosystems and steal confidential information that is transmitted in the network. This section discusses some countermeasures that can be adopted to prevent such attacks.

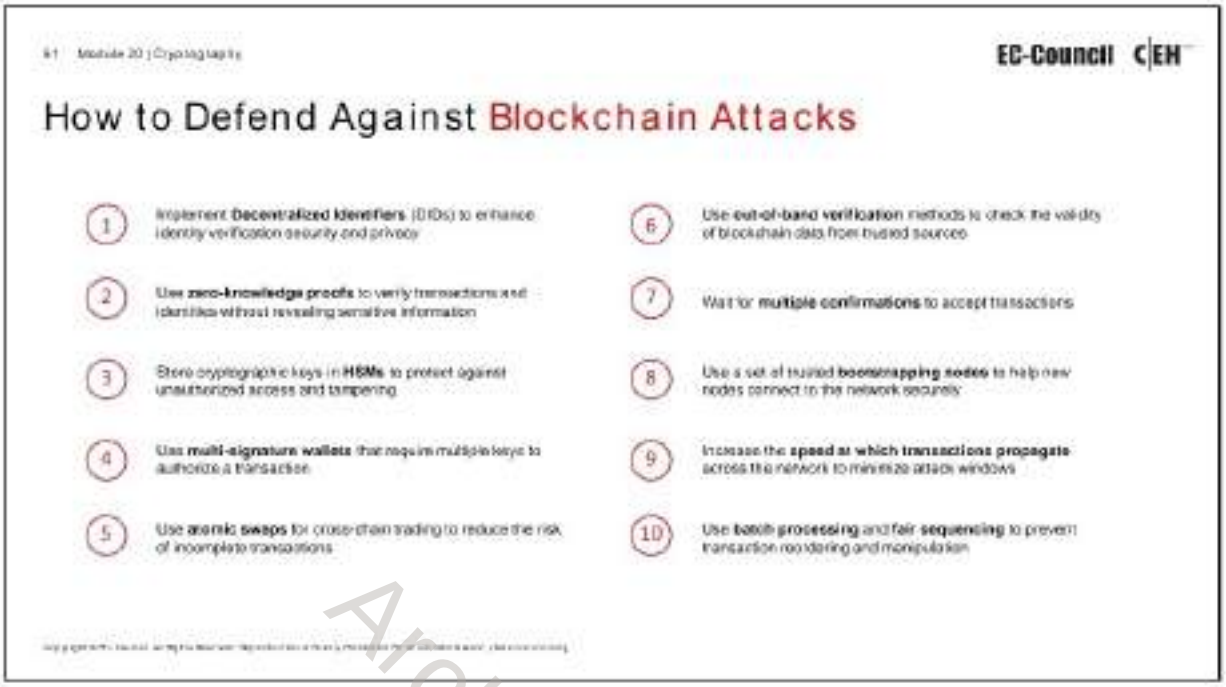


How to Defend Against Cryptographic Attacks

The following countermeasures can be adopted to prevent cryptographic attacks:

- Access to cryptographic keys should be given directly to the application or user.
- IDS should be deployed to monitor exchange and access of keys.
- Passphrases and passwords must be used to encrypt the key, if stored on the disk.
- Keys should not be present inside the source code or binaries.
- For certificate signing, the transfer of private keys should not be allowed.
- For symmetric algorithms, a key size of 256 bits should be preferred for a secure system, especially in the case of large transactions.
- Message authentication must be implemented for the encryption of symmetric-key protocols.
- For asymmetric algorithms, a key size of at least 2048 bits should be considered for secure and highly protected applications.
- In the case of hash algorithms, a hash length of 256 bits or higher should be considered for secure applications.
- Only recommended tools or products should be used rather than self-engineered crypto algorithms or functions.
- Impose a limit on the number of operations per key.
- The output of the hash function should have a larger bit length that makes it difficult to decrypt.

- Design applications and protocols that can avoid simple encryption key relationships, i.e., each encrypted key should be created from a key derivation function (KDF).
- Upgrade to the latest security standards.
- Use strong key schedules to mitigate the risks of related key attacks.
- Enforce hardware-backed security such as hardware security modules (HSMs) to enhance cryptographic key security.
- Do not use a single cryptographic key for multiple purposes.
- Use redundant cryptosystems to encrypt data multiple times.
- Implement regular key rotation to minimize the exposure of cryptographic keys to potential attacks.
- Use digital signatures to sign important messages or documents. Verify the signatures before accepting or processing data to prevent tampering or unauthorized modifications.
- Utilize hardware-based random number generators (RNGs) or collect entropy from diverse sources to generate cryptographic keys, nonces, and initialization vectors.
- Adopt quantum-resistant algorithms, such as lattice-based cryptography or hash-based signatures to provide security against quantum adversaries.
- Utilize zero-knowledge proof protocols such as zk-SNARKs for secure authentication and data integrity verification without exposing sensitive information.
- Prepare for the transition to post-quantum cryptography by evaluating and testing candidate algorithms recommended by standardization bodies such as NIST.
- Use Advanced Encryption Standard (AES) algorithms that are resistant to cryptanalysis and are reliable in protecting sensitive data.
- Employ techniques such as key stretching and salting to increase the computational cost of brute-force attacks against derived keys.
- Add a unique, random value (salt) to each password before hashing to prevent attackers from using precomputed hash dictionaries.
- Use protocols such as TLS to encrypt communication and verify the identity of both parties to prevent interception and tampering.
- Implement encryption schemes that do not produce predictable outputs for chosen inputs, such as using probabilistic encryption methods.
- Use encryption schemes to combine confidentiality and integrity, such as Galois/Counter Mode (GCM) or Encrypt-then-MAC, to ensure that ciphertexts cannot be manipulated.
- Change encryption keys periodically and derive keys from passwords using functions like PBKDF2, bcrypt, or Argon2 to strengthen key generation.
- Utilize hash functions resistant to collision attacks, such as SHA-256 or SHA-3, instead of weaker algorithms like MD5 or SHA-1.
- Adopt quantum-resistant algorithms to secure against quantum attacks, such as lattice-based, hash-based, or code-based cryptography.

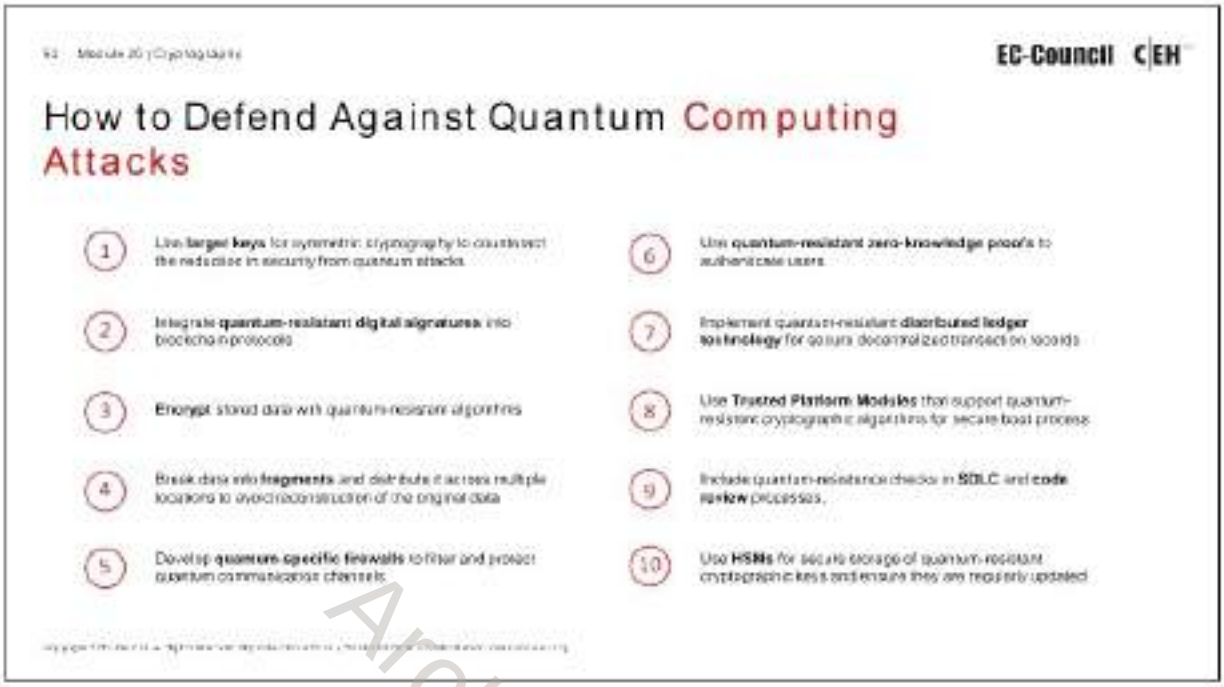


How to Defend Against Blockchain Attacks

The following countermeasures can be adopted to prevent blockchain attacks:

- Implement decentralized identifiers (DIDs) to enhance identity verification security and privacy.
- Use zero-knowledge proofs to verify transactions and identities without revealing sensitive information.
- Store cryptographic keys in HSMs to protect against unauthorized access and tampering.
- Use multi-signature wallets that require multiple keys to authorize a transaction.
- Implement real-time monitoring systems or use machine learning algorithms to detect abnormal transaction patterns that may indicate double-spending activities.
- Combine proof-of-work (PoW) with proof-of-stake (PoS) to mitigate the risks associated with high energy consumption in blockchain networks.
- Implement advanced DDoS protection mechanisms, such as decentralized DDoS mitigation networks, to defend against attacks that overwhelm network nodes.
- Employ formal verification methods to mathematically prove the correctness and security of smart contracts.
- Conduct regular and thorough code audits of blockchain and smart contract code.
- Implement secure interoperability protocols to protect against attacks that exploit cross-chain transactions.
- Use atomic swaps for cross-chain trading to reduce the risk of incomplete transactions.

- Boost mining pool surveillance.
- Avoid storing blockchain keys in unsecured computer files, such as Word documents, notepad files, or sticky notes.
- Make sure to use a trusted encryption program to store keys on a device.
- Implement randomized peer selection algorithms to prevent attackers from predicting the peers to which a node will connect.
- Implement timeouts for peer connections to force periodic reconnections.
- Maintain secondary trusted communication channels on which nodes can fall if they detect unusual network behaviors.
- Use out-of-band verification methods to check the validity of the blockchain data from trusted sources.
- Implement reputation systems that score peers based on their behavior and reliability, and prefer connecting to high-reputation peers.
- Use a set of trusted bootstrapping nodes to help new nodes securely connect to the network.
- Wait for multiple confirmations to accept transactions.
- Increase the speed at which transactions propagate across the network to minimize attack windows.
- Hide the details of pending transactions to prevent front-running attacks.
- Use batch processing and fair sequencing to prevent transaction reordering and manipulation.
- Develop secure consensus and order-matching algorithms to resist double spending and transaction manipulation.
- Use mechanisms that randomize the submission times of transactions to make it more difficult for attackers to predict and manipulate order executions.



How to Defend Against Quantum Computing Attacks

The following countermeasures can be adopted to prevent quantum computing attacks:

- Use quantum-resistant cryptographic algorithms such as lattice-based cryptography, hash-based cryptography, code-based cryptography, etc.
- Apply the principles of quantum mechanics to securely distribute cryptographic keys.
- Combine classical cryptographic methods with quantum-resistant algorithms to ensure security during transition periods.
- Use larger keys for symmetric cryptography to counteract the reduction in security from quantum attacks.
- Regularly change cryptographic keys to limit the time they are vulnerable.
- Implement protection against side-channel attacks such as power analysis and electromagnetic emissions.
- Develop virtual private networks that use quantum-resistant encryption methods.
- Develop and deploy authentication protocols that are secure against quantum attacks.
- Use digital certificates based on quantum-resistant cryptographic algorithms.
- Enhance MFA using quantum-resistant methods to ensure that even if one factor is compromised, the overall authentication remains secure.
- Design cryptographic systems in a modular fashion to allow for quick updates and replacement of cryptographic algorithms.

- Use software frameworks that support multiple cryptographic algorithms and can switch algorithms with minimal disruption.
- Integrate quantum-resistant digital signatures into blockchain protocols to ensure the integrity and authenticity of transactions.
- Encrypt the stored data with quantum-resistant algorithms to ensure that they remain secure even when quantum computers become more powerful.
- Break the data into fragments and distribute them across multiple locations to avoid the reconstruction of the original data, even if some fragments are compromised.
- Isolate critical systems from less-secure networks and implement multiple security layers to limit the impact of potential quantum attacks.
- Use cloud-based key management services that employ quantum-resistant algorithms.
- Employ secure multi-party computation (MPC) protocols in cloud environments.
- Develop quantum-specific firewalls to filter and protect quantum communication channels.
- Use quantum-resistant zero-knowledge proofs to authenticate users without revealing sensitive information.
- Implement quantum-resistant distributed ledger technology (DLT) to ensure secure and decentralized transaction records.
- Apply quantum-resistant threshold cryptography to require multiple parties to approve transactions.
- Ensure that random number generation used in cryptographic systems is secure against quantum threats.
- Use trusted platform modules (TPMs) that support quantum-resistant cryptographic algorithms to secure the boot process.
- Implement role-based access control (RBAC) and attribute-based access control (ABAC) with quantum-safe cryptographic protection.
- Include quantum-resistance checks in SDLC and code review processes.
- Integrate quantum-safe security measures into continuous integration/continuous deployment (CI/CD) pipelines.
- Use Hardware security modules (HSMs) for secure storage of quantum-resistant cryptographic keys and to ensure that they are regularly updated with quantum-safe firmware.

Key Stretching

Key stretching refers to processes used to make a weak key stronger, usually by making it longer. This technique helps in defending against brute-force attacks. In general, passwords or passphrases generated by end users are weak and predictable. Hence, key stretching helps security professionals/users to prevent such attacks by strengthening their passwords.

In the key stretching technique, the initial key is given as input to an algorithm that generates an enhanced key. The key must be sufficiently resistant to brute-force attacks. It is very difficult for the attackers to predict the enhanced key as they need to try every possible combination of the key or likely combinations of the enhanced key.

There are many functions and libraries that perform key stretching as part of their working:

- Password-Based Key Derivation Function 2 (PBKDF2) is part of PKCS #5 v. 2.01. It applies some functions (such as a hash or HMAC) to the password or passphrase along with Salt to produce a derived key.
- Bcrypt is used with passwords, and it essentially uses a variant of the Blowfish algorithm, converted to a hashing algorithm, to hash a password and add Salt to it.

Module Summary

This module discussed basic cryptography concepts used to protect confidential data as well as different types of cryptography. It also described ciphers and different encryption algorithms used to encrypt or decrypt data in detail. Furthermore, it illustrated various cryptography tools. It then highlighted the importance of PKI in encryption and discussed the email encryption protocols and tools in detail. It also discussed disk encryption along with various disk encryption tools. Moreover, it explained the various types of cryptanalysis methods and code breaking methodologies in use. Subsequently, it presented the various types of cryptanalysis attacks and cryptanalysis tools. Finally, it ended with an explanation of the countermeasures against various cryptography attacks.

Glossary

A

- **Availability:** Assurance that the systems responsible for delivering, storing, and processing information are accessible when required by the authorized users.
- **Authenticity:** Refers to the characteristic of a communication, document, or any data that ensures the quality of being genuine.
- **AI-Driven Ethical Hacking:** AI-driven ethical hacking is a modern approach to cybersecurity where artificial intelligence (AI) technologies are used to enhance the capabilities of ethical hackers.
- **AutoGPT:** AutoGPT is an AI-powered tool designed to automate task execution and data processing.
- **Active Attacks:** Active attacks tamper with the data in transit or disrupt communication or services between the systems to bypass or break into secured systems.
- **Adversary Behavioral Identification:** Adversary behavioral identification involves the identification of the common methods or techniques followed by an adversary to launch attacks on or to penetrate an organization's network.
- **Active Footprinting:** Active footprinting involves gathering information about the target with direct interaction.
- **ARP Ping Scan:** Attackers send ARP request probes to target hosts, and an ARP response indicates that the host is active.
- **ACK Flag Probe Scan:** Attackers send TCP probe packets set with an ACK flag to a remote device, and then analyze the header information (TTL and WINDOW field) of received RST packets to determine if the port is open or closed.
- **Anonymizer:** An anonymizer is an intermediate server placed between you as the end user and the website to access the website on your behalf and make your web surfing activities untraceable.
- **Application Vulnerability Scanning:** Tests and analyzes all elements of the web infrastructure for any misconfiguration, outdated content, or known vulnerabilities.
- **Automated Vulnerability Scanning:** Uses automated software tools such as Nessus, Qualys, and GFI LanGuard to systematically identify, evaluate, and report security vulnerabilities.
- **Audio Spyware:** Audio spyware is a sound surveillance program designed to record sound onto a computer.
- **Anti-Keyloggers:** Anti-keyloggers, also called anti-keystroke loggers, detect and disable keystroke logger software.
- **Application Flaws:** Application flaws are vulnerabilities in applications that are exploited by attackers.
- **Audio Steganography:** Audio steganography refers to hiding secret information in audio files such as .MP3, .RM, and .WAV.
- **Advanced Persistent Threats:** Advanced persistent threats (APTs) are defined as a type of network attack, where an attacker gains unauthorized access to a target network and remains undetected for a long period of time.
- **Antivirus Sensor System:** An antivirus sensor system is a collection of computer software that detects and analyzes malicious code threats such as viruses, worms, and Trojans.
- **Adware:** A software or a program that supports advertisements and generates unsolicited ads and pop-ups.

- **API Calls:** Application programming interfaces (APIs) are parts of the Windows OS that allow external applications to access OS information such as file systems, threads, errors, registry, kernel, buttons, mouse pointer, network services, web, and the Internet.
- **Anti-Trojan Software:** Anti-Trojan software is a tool or program that is designed to identify and prevent malicious Trojans or malware from infecting computer systems or electronic devices.
- **Angler Phishing:** Angler phishing is a cyber phishing fraud in which attackers target disgruntled users or customers over social media platforms.
- **Active Sniffing:** Active sniffing involves injecting Address Resolution Packets (ARP) into the network to flood the switch's Content Addressable Memory (CAM) table, which keeps track of host-port connections.
- **Address Resolution Protocol (ARP):** Address Resolution Protocol (ARP) is a stateless protocol used for resolving IP addresses to machine (MAC) addresses.
- **ARP Spoofing Attack:** ARP spoofing involves constructing many forged ARP request and reply packets to overload the switch.
- **Application-Level Hijacking:** Application-level hijacking refers to gaining control over the HTTP's user session by obtaining the session IDs.
- **Anomaly Detection:** It detects the intrusion based on the fixed behavioral characteristics of the users and components in a computer system.
- **Application-Level Firewall**: Application-level gateways (proxies) can filter packets at the application layer of the OSI model (or the application layer of TCP/IP).
- **Application Proxy:** An application-level proxy works as a proxy server and filters connections for specific services.
- **API DDoS Attack:** The DDoS attack involves saturating an API with a huge volume of traffic from multiple infected computers (botnet) to delay API services to legitimate users.
- **Automated Web Application Security Testing:** It is a technique employed for automating the testing process. These testing methods and procedures are incorporated into each stage of development to report feedback constantly.
- **Application Whitelisting:** Application whitelisting contains a list of application components such as software libraries, plugins, extensions, and configuration files, which can be permitted to execute in the system.
- **Application Blacklisting:** Application blacklisting contains a list of malicious applications or software that are not permitted to be executed in the system or the network.
- **Access point (AP):** An AP is used to connect wireless devices to a wireless/wired network.
- **Association:** It refers to the process of connecting a wireless device to an AP.
- **Agent Smith Attack:** Agent Smith attacks are carried out by luring victims into downloading and installing malicious apps designed and published by attackers in the form of games, photo editors, or other attractive tools from third-party app stores such as 9Apps.
- **Android Rooting:** Rooting process involves exploiting security vulnerabilities in the device firmware and copying the SU binary to a location in the current process's PATH (e.g., /system/xbin/su) and granting it executable permissions with the chmod command.
- **Anything-as-a-Service (XaaS):** Anything as a service or everything as a service (XaaS) is a cloud-computing and remote-access service that offers anything as a service over the Internet based on the user's demand.
- **AWS Cognito:** AWS Cognito is a service provided by Amazon Web Services that streamlines the authentication, authorization, and user management of web and mobile applications.

- **Asymmetric Encryption:** Asymmetric encryption (public-key) uses different encryption keys, which are called public and private keys for encryption and decryption, respectively.
- **Advanced Encryption Standard (AES):** The Advanced Encryption Standard (AES) is a National Institute of Standards and Technology (NIST) specification for the encryption of electronic data.

B

- **Behavioral Indicators:** Behavioral indicators of compromise are used to identify specific behavior related to malicious activities.
- **Black Hats:** Black hats are individuals who use their extraordinary computing skills for illegal or malicious purposes.
- **Blue Hat:** Blue hats are contract-based cybersecurity professionals hired by organizations to evaluate systems or software for vulnerabilities.
- **BugBountyGPT:** BugBountyGPT is tailored for bug bounty hunters and provides tools and insights for identifying and reporting security vulnerabilities.
- **BugHunterGPT:** BugHunterGPT assists security researchers in identifying and reporting bugs and vulnerabilities.
- **Brute-Force Attack:** In a brute-force attack, attackers try every combination of characters until the password is broken.
- **Buffer Overflow:** Buffer overflow or overrun is a common vulnerability in an applications or programs that accepts more data than the allocated buffer.
- **Backdoor Trojans:** A backdoor is a program that can bypass the standard system authentication or conventional system mechanisms such as IDS and firewalls, without being detected.
- **Botnet Trojans:** Attackers use botnet Trojans to infect a large number of computers throughout a large geographical area to create a network of bot that can achieve control via a command-and-control (C&C) center.
- **Baiting:** Baiting is a technique in which attackers offer end users something alluring in exchange for important information such as login details and other sensitive data.
- **Botnet:** A botnet is a huge network of compromised systems and can be used by an attacker to launch denial-of-service attacks.
- **Broken Access Control:** Broken access control is a method in which an attacker identifies a flaw related to access control and bypasses the authentication, which allows them to compromise the network.
- **Base64 Encoding:** The Base64 encoding scheme represents any binary data using only printable ASCII characters.
- **Blind/Inferential SQL Injection:** In blind SQL injection, an attacker poses a true or false question to the database to determine whether the application is vulnerable to SQL injection.
- **Blacklist Validation:** Blacklist validation rejects all the malicious inputs that have been disapproved for protected access.
- **Bandwidth:** It describes the amount of information that may be broadcast over a connection.
- **Basic service set identifier (BSSID):** It is the media access control (MAC) address of an access point (AP) or base station that has set up a basic service set (BSS).
- **Bluesnarfing:** Bluesnarfing is the theft of information from a wireless device through a Bluetooth connection, often between phones, desktops, laptops, PDAs, and other devices.

- **Bluebugging:** Bluebugging involves gaining remote access to a target Bluetooth-enabled device and using its features without the victim's knowledge or consent.
- **BYOD:** Bring your own device (BYOD) refers to a policy that allows an employee to bring their personal devices, such as laptops, smartphones, and tablets, to their workplace and use them to access the organization's resources by following the access privileges.
- **BlueBorne Attack:** A BlueBorne attack is performed on Bluetooth connections to gain access and take full control of the target device.
- **Business Network:** It comprises of a network of systems that offer information infrastructure to the business.
- **Basic Process Control System (BPCS):** A BPCS is responsible for process control and monitoring of the industrial infrastructure.
- **Blowfish:** Blowfish is a type of symmetric block cipher algorithm designed to replace DES or IDEA algorithms.
- **Blockchain:** A blockchain, also referred to as distributed ledger technology (DLT), is used to record and store the history of transactions in the form of blocks.

C

- **CEH Hacking Methodology (CHM):** EC-council's CEH hacking methodology (CHM) defines the step-by-step process to perform ethical hacking.
- **Confidentiality:** Assurance that the information is accessible only to those authorized to have access.
- **ChaosGPT:** ChaosGPT is an AI tool designed to simulate and understand chaotic and unpredictable behaviors.
- **CybGPT:** CybGPT is a comprehensive AI tool for cybersecurity professionals that offers a wide range of features for enhancing security operations.
- **Close-in Attacks:** Close-in attacks are performed when the attacker is in close physical proximity with the target system or network in order to gather, modify, or disrupt access to information.
- **Cyber Kill Chain Methodology:** The cyber kill chain methodology is a component of intelligence-driven defense for the identification and prevention of malicious intrusion activities.
- **Cyber Terrorists:** Cyber terrorists are individuals with a wide range of skills, motivated by religious or political beliefs, to create fear of large-scale disruption of computer networks.
- **Criminal Syndicates:** Groups of individuals that are involved in organized, planned, and prolonged criminal activities. They illegally embezzle money by performing sophisticated cyber-attacks.
- **Clearing Tracks:** Clearing tracks refers to the activities carried out by an attacker to hide malicious acts.
- **Cyber Threat Intelligence:** Cyber Threat Intelligence (CTI) is defined as the collection and analysis of information about threats and adversaries and the drawing of patterns that provide the ability to make knowledgeable decisions for preparedness, prevention, and response actions against various cyber-attacks.
- **Threat Intelligence Lifecycle:** The threat intelligence lifecycle is a continuous process of developing intelligence from raw data that supports organizations to develop defensive mechanisms to thwart emerging risks and threats.
- **Competitive Intelligence Gathering:** Competitive intelligence gathering is the process of identifying, gathering, analyzing, verifying, and using information about your competitors from resources, such as the Internet.
- **Common Vulnerability Scoring System (CVSS):** CVSS is a published standard that provides an open framework for communicating the characteristics and impacts of IT vulnerabilities.

- **Common Vulnerabilities and Exposures (CVE):** CVE® is a publicly available and free-to-use list or dictionary of standardized identifiers for common software vulnerabilities and exposures.
- **Common Weakness Enumeration (CWE):** Common Weakness Enumeration (CWE) is a category system for software vulnerabilities and weaknesses.
- **Credentialed/Authenticated Vulnerability Scanning:** Credentialed vulnerability scanning is a security testing method in which the scanner logs into the target system using valid credentials to perform a more thorough and comprehensive scan.
- **Cloud-based Vulnerability Scanning :** This type of assessment focuses on evaluating overall security of the cloud infrastructure according to the cloud service provider's best practices or guidelines.
- **Component Object Model (COM):** The Component Object Model (COM) is an interface module in Windows environments that enables a software component to interact with another software component's code without being aware of their actual implementation.
- **Child-Monitoring Spyware:** Child-monitoring spyware allows you to track and monitor what children are doing on the computer, both online and offline.
- **Combinator Attack:** Attackers combine the entries of the first dictionary with those of the second dictionary to generate a new wordlist to crack the password of the target system.
- **Crypter:** Software that protects malware from undergoing reverse engineering or analysis, thus making the task of the security mechanism harder in its detection.
- **Computer Worms:** Computer worms are malicious programs that independently replicate, execute, and spread across the network connections, thus consuming available computing resources without human interaction.
- **Consent Phishing:** Consent phishing is a type of social engineering attack that exploits the OAuth authentication protocol used by web services such as Google, Facebook, and Microsoft.
- **Chain Letters:** Emails that offer free gifts such as money and software on condition that the user forwards the mail to a specified number of people.
- **Catfishing Attack:** A catfishing attack is an online phishing scam in which attackers target a person on social media platforms and perform identity theft.
- **CRIME Attack:** Compression Ratio Info-Leak Made Easy (CRIME) is a client-side attack that exploits the vulnerabilities present in the data compression feature of protocols, such as SSL/TLS, SPDY, and HTTPS.
- **Circuit-Level Gateway Firewall:** Circuit-level gateways monitor requests to create sessions and determine if those sessions will be allowed.
- **Cross-Site Scripting (XSS) Attacks:** Cross-site scripting ('XSS' or 'CSS') attacks exploit vulnerabilities in dynamically generated web pages, enabling malicious attackers to inject client-side scripts into web pages viewed by other users.
- **Cross-Site Request Forgery (CSRF) Attack:** Cross-Site Request Forgery (CSRF) attacks exploit web page vulnerabilities that allow an attacker to force an unsuspecting user's browser to send malicious requests they did not intend.
- **Clickjacking Attack:** Attackers perform clickjacking attacks by tricking the victim into clicking on any malicious web page element that is placed transparently on the top of any trusted web page.
- **Cookie Poisoning:** It is a type of parameter tampering attack in which the attacker modifies the cookie contents to draw unauthorized information about a user and thus perform identity theft.
- **Code Analysis:** Code analysis or code review is the most effective technique for identifying vulnerabilities or flaws in the code.

- **Call Spoofing:** Call spoofing is a technique used by attackers to manipulate the caller ID information displayed on a recipient's phone when they receive a call.
- **CoAP:** Constrained Application Protocol (CoAP) is a web transfer protocol used to transfer messages between constrained nodes and IoT networks.
- **Cookie Sniffing:** It is a technique in which an attacker sniffs a cookie containing the session ID of the victim who has logged in to a target website and uses the cookie to bypass the authentication process and log in to the victim's account.
- **Cookie Replay:** It is a technique used to impersonate a legitimate user by replaying the session/cookie that contains the session ID of that user (as long as he/she remains logged in).
- **Camfecting Attack:** A camfecting attack is a webcam capturing attack that is performed to gain access to the camera of a target's computer or mobile device.
- **Critical Infrastructure:** A collection of physical or logical systems and assets that the failure or destruction of which will severely impact the security, safety, economy, or public health.
- **Cloud Computing:** Cloud computing is an on-demand delivery of IT capabilities where IT infrastructure and applications are provided to subscribers as a metered service over a network.
- **Container-as-a-Service (CaaS):** It provides services such as virtualization of container engines, management of containers, applications, and clusters through a web portal, or an API.
- **Community Cloud:** It is a multi-tenant infrastructure shared among organizations from a specific community with common computing concerns, such as security, regulatory compliance, performance requirements, and jurisdiction.
- **Cloud Consumer:** A person or organization that uses cloud computing services.
- **Cloud Provider:** A person or organization providing services to interested parties.
- **Cloud Carrier:** An intermediary for providing connectivity and transport services between cloud consumers and providers.
- **Cloud Auditor:** A party for making independent assessments of cloud service controls and taking an opinion thereon.
- **Cloud Broker:** An entity that manages cloud services in terms of use, performance, and delivery, and maintains the relationship between cloud providers and consumers.
- **Container:** A container is a package of an application/software including all its dependencies such as library files, configuration files, binaries, and other resources that run independently of other processes in the cloud environment.
- **Container Orchestration:** Container orchestration is an automated process of managing the lifecycles of software containers and their dynamic environments.
- **Cluster:** A cluster refers to a set of two or more connected nodes that run parallelly to complete a task.
- **Cloud Cryptojacking:** Cryptojacking is the unauthorized use of the victim's computer to stealthily mine digital currency.
- **Cloudborne Attack:** Cloudborne is a vulnerability residing in a bare-metal cloud server that enables the attackers to implant a malicious backdoor in its firmware.
- **Cache Poisoned Denial of Service (CPDoS):** In CPDoS, attackers create malformed or oversized HTTP requests to trick the origin web server into responding with malicious or error content, which is cached at the CDN servers.
- **Cloud Snooper Attack:** Cloud snooper attacks are triggered at AWS security groups (SGs) to compromise the target server and extract sensitive data stealthily.

- **Cloud Application Security:** It is a set of rules, processes, policies, controls, and techniques used to administer all the data exchange between collaborative cloud platforms.
- **Cloud Integration:** Cloud integration is the process of grouping multiple cloud environments together in the form of a public or hybrid cloud.
- **Cloud Auditing:** Cloud auditing is the process of analyzing the services offered by cloud providers and verifying the conformity to requirements for privacy, security, etc.
- **Cloud Security Alliance (CSA):** CSA is a nonprofit global organization that provides rising awareness and promotes best practices and security policies to help and secure the cloud environment.
- **CASB:** Cloud Access Security Brokers (CASBs) are on-premise or cloud-hosted solutions responsible for enforcing security, compliance, and governance policies for the cloud applications.
- **Cryptography:** Cryptography is the conversion of data into a scrambled code that is encrypted and sent across a private or public network.
- **Ciphers:** In cryptography, a cipher is an algorithm (a series of well-defined steps) for performing encryption and decryption.
- **CAST-128:** CAST-128, also called CAST5, is a symmetric-key block cipher having a classical 12- or 16-round Feistel network with a block size of 64 bits.
- **Camellia:** Camellia is a symmetric-key block cipher having either 18 rounds (for 128-bit keys) or 24 rounds (for 256-bit keys).
- **CHAP:** The Challenge-Handshake Authentication Protocol (CHAP) is an authentication mechanism used by Point-to-Point Protocol (PPP) servers to authenticate or validate the identity of remote clients or network hosts.
- **Cryptanalysis:** Cryptanalysis is the study of ciphers, ciphertext, or cryptosystems with the ability to identify vulnerabilities in them and thus extract plaintext from ciphertext even if the cryptographic key or algorithm used to encrypt the plaintext is unknown.

D

- **Distribution Attacks:** Distribution attacks occur when attackers tamper with hardware or software prior to installation.
- **Defense-in-Depth:** Defense-in-depth is a security strategy in which several protection layers are placed throughout an information system.
- **Diamond Model:** The Diamond Model offers a framework for identifying the clusters of events that are correlated on any of the systems in an organization.
- **Deep Web:** It consists of web pages and contents that are hidden and unindexed and cannot be located using traditional web browsers and search engines.
- **Dark Web or Darknet:** It is the subset of the deep web that enables anyone to navigate anonymously without being traced.
- **Dumpster Diving:** This uncouth technique, also known as trashing, involves the attacker rummaging for information in garbage bins.
- **DNS Cache Snooping:** DNS cache snooping is a DNS enumeration technique whereby an attacker queries the DNS server for a specific cached DNS record.
- **DNSSEC Zone Walking:** DNSSEC zone walking is a DNS enumeration technique where an attacker attempts to obtain internal records of the DNS server if the DNS zone is not properly configured.
- **Database Vulnerability Scanning:** A database scan focuses on testing databases for the presence of any misconfiguration or known vulnerabilities.

- **Dictionary Attack:** In this type of attack, a dictionary file is loaded into a cracking application that runs against user accounts.
- **Distributed Network Attack:** A Distributed Network Attack (DNA) technique is used for recovering passwords from hashes or password-protected files using the unused processing power of machines across the network.
- **DCSync Attack:** In a DCSync attack, an attacker initially compromises and obtains privileged account access with domain replication rights and activates replication protocols to create a virtual domain controller (DC) similar to the original AD.
- **Document Steganography:** Document steganography is the technique of hiding secret messages transferred in the form of documents.
- **Domain Dominance:** Domain dominance is a process of taking control over critical assets such as domain controllers on a target system and gaining access to other networked resources.
- **Data Protection API (DPAPI):** DPAPI is a unified location in Windows environments where all the cryptographically secured files, passwords of browsers, and other critical data are stored.
- **Downloader:** A type of Trojan that downloads other malware from the Internet on to the PC. Usually, attackers install downloader software when they first gain access to a system.
- **Dropper:** A type of Trojan that covertly installs other malware files on to the system.
- **Dynamic Malware Analysis:** It involves executing the malware code to know how it interacts with the host system and its impact on the system after infection.
- **DHCP Starvation Attack:** This is a denial-of-service (DoS) attack on the DHCP servers where the attacker broadcasts forged DHCP requests and tries to lease all the DHCP addresses available in the DHCP scope.
- **DNS Poisoning:** DNS poisoning is a technique that tricks a DNS server into believing that it has received authentic information when it has not received any
- **DNS Cache Poisoning:** DNS cache poisoning refers to altering or adding forged DNS records into the DNS resolver cache so that a DNS query is redirected to a malicious site.
- **Diversion Theft:** The attacker tricks a person responsible for making a genuine delivery into delivering the consignment to a location other than the intended location.
- **Deepfake Attack:** A deepfake attack is a type of phishing attack in which attackers create false media of a person they target using advanced technologies such as ML and AI.
- **DoS Attack:** Denial-of-Service (DoS) is an attack on a computer or network that reduces, restricts, or prevents accessibility of system resources to its legitimate users.
- **DDoS Attack:** Distributed denial-of-service (DDoS) is a coordinated attack that involves a multitude of compromised systems (Botnet) attacking a single target, thereby denying service to users of the targeted system.
- **Distributed Reflection Denial-of-Service (DRDoS) Attack:** A distributed reflected denial-of-service attack (DRDoS), also known as a spoofed attack, involves the use of multiple intermediary and secondary machines that contribute to the actual DDoS attack against the target machine or application.
- **DNS over HTTPS:** DNS over HTTPS (DoH) is an enhanced version of DNS protocol, which is used to prevent snooping of user's web activities or DNS queries during the DNS lookup process.
- **Demilitarized Zone (DMZ):** The demilitarized zone (DMZ) is an area that hosts computer(s) or a small sub-network placed as a neutral zone between a particular company's internal network and an untrusted external network to prevent outsider access to a company's private data.

- **Database Honeypots:** Database honeypots employ fake databases that are vulnerable to perform database-related attacks such as SQL injection and database enumeration.
- **DNS Server Hijacking:** Attacker compromises the DNS server and changes the DNS settings so that all the requests coming towards the target web server are redirected to his/her own malicious server.
- **Directory Traversal:** Directory traversal allows attackers to access restricted directories, including application source code, configuration, and critical system files to execute commands outside the web server's root application directory.
- **DNS Rebinding Attack:** Attackers use the DNS rebinding technique to bypass the same-origin policy's security constraints, allowing the malicious web page to communicate with or make arbitrary requests to local domains.
- **Dynamic Application Security Testing (DAST):** It is also known as a black-box testing approach and is performed directly on running code to identify issues related to interfaces, requests/responses, sessions, scripts, authentication processes, code injections, etc.
- **Direct-Sequence Spread Spectrum (DSSS):** DSSS is a spread spectrum technique that multiplies the original data signal with a pseudo-random noise-spreading code.
- **Directional Antenna:** A directional antenna can broadcast and receive radio waves from a single direction.
- **Dipole Antenna:** A dipole antenna is a straight electrical conductor measuring half a wavelength from end to end, and it is connected at the center of the radio frequency (RF) feed line.
- **Disassociation Attack:** In a disassociation attack, the attacker makes the victim unavailable to other wireless devices by destroying the connectivity between the AP and client.
- **De-authentication Attack:** In a de-authentication attack, the attacker floods station(s) with forged de-authenticates or disassociates to disconnect users from an AP.
- **Distributed Control System (DCS):** DCS is a highly engineered and large-scale control system that is often used to perform industry specific tasks.
- **Desktop-as-a-Service (DaaS):** This cloud computing service offers on-demand virtual desktops and apps to subscribers.
- **Docker:** Docker is an open-source technology used for developing, packaging, and running applications and all its dependencies in the form of containers, to ensure that the application works in a seamless environment.
- **Data Encryption Standard (DES):** DES is designed to encipher and decipher blocks of data consisting of 64 bits under control of a 56-bit key.
- **DSA:** The Digital Signature Algorithm (DSA) is a Federal Information Processing Standard for digital signatures.
- **Diffie–Hellman:** It is a cryptographic protocol that allows two parties to establish a shared key over an insecure channel.
- **Digital Signature:** Digital signature uses asymmetric cryptography to simulate the security properties of a signature in digital rather than written form.
- **DUHK Attack:** DUHK (Don't Use Hard-Coded Keys) is a cryptographic vulnerability that allows an attacker to obtain encryption keys used to secure VPNs and web sessions.
- **DROWN Attack:** A DROWN attack is a cross-protocol weakness that can communicate and initiate an attack on servers that support recent SSLv3/TLS protocol suites.
- **DeFI Sandwich Attack:** A decentralized finance (DeFi) sandwich attack is a blockchain attack targeting decentralized exchanges (DEXs) and automated market makers (AMMs) to manipulate market dynamics.

E

- **Email Indicators:** Email indicators are used to send malicious data to the target organization or individual.
- **Ethical Hacking:** Ethical hacking involves the use of hacking tools, tricks, and techniques to identify vulnerabilities and ensure system security.
- **Eavesdropping:** Eavesdropping is the act of secretly listening to the conversations of people over a phone or video conference without their consent.
- **Enumeration:** Enumeration is the process of extracting usernames, machine names, network resources, shares, and services from a system or network.
- **External Vulnerability Scanning:** External scanning examines the network from a hacker's point of view to identify exploits and vulnerabilities accessible to the outside world.
- **Exploit:** A malicious code that breaches the system security via software vulnerabilities to access information or install malware.
- **Exploit Chaining:** Exploit chaining, also referred to as vulnerability chaining, is a cyberattack that combines various exploits or vulnerabilities to infiltrate and compromise the target from its root level.
- **Email Spyware:** Email spyware is a program that monitors, records, and forwards all incoming and outgoing emails.
- **Exploit Kit:** An exploit kit or crimeware toolkit is a platform to deliver exploits and payloads such as Trojans, spywares, backdoors, bots, and buffer overflow scripts to the target system.
- **Elicitation:** Attackers extract information from the victim by engaging him/her in normal and disarming conversations.
- **E-wallet Phishing:** An attacker targets users of electronic wallets by sending a phishing email or messages to potential victims, posing as a legitimate e-wallet provider.
- **Egress Filtering:** Egress filtering scans the headers of IP packets leaving a network.
- **Email Honeypots:** Email honeypots are also called email traps. They are nothing but fake email addresses that are specifically used to attract fake and malicious emails from adversaries.
- **Error Based SQL Injection:** Error based SQL injection forces the database to perform some operation in which the result will be an error.
- **Evil Twin:** An evil twin is a wireless AP that pretends to be a legitimate AP by imitating its SSID.
- **Edge Computing:** Edge computing is a distributed decentralized computing model in which data processing is performed close to edge devices.
- **EC2 Instances:** Amazon EC2 (Elastic Compute Cloud) is a web service that provides resizable computing capacity in the cloud and is designed to make web-scale cloud computing easier for developers.
- **Elliptic Curve Cryptography:** ECC is a modern public-key cryptography developed to avoid larger cryptographic key usage.
- **Eclipse Attack:** An Eclipse attack is a type of blockchain attack in which an attacker isolates a target node from the rest of the network by surrounding it with malicious nodes, thereby effectively controlling the node's view of the blockchain.

F

- **FreedomGPT:** FreedomGPT is an AI tool designed to provide ethical hackers with unrestricted access to AI.
- **FraudGPT:** FraudGPT is an AI tool specifically designed to detect and prevent fraudulent activities.

- **Federal Information Security Management Act (FISMA):** The FISMA provides a comprehensive framework for ensuring the effectiveness of information security controls over information resources that support Federal operations and assets.
- **Footprinting:** Footprinting is the first step of any attack on information systems in which an attacker collects information about a target network to identify various ways to intrude into the system.
- **Fingerprint Attack:** Attackers break down the passphrase into fingerprints comprising single and multi-character combinations to crack complex passwords.
- **Folder Steganography:** In folder steganography, files are hidden and encrypted within a folder and do not appear to normal Windows applications, including Windows Explorer.
- **Fileless Malware:** Fileless malware, also known as non-malware, infects legitimate software, applications, and other protocols existing in the system to perform various malicious activities.
- **File Fingerprinting:** File fingerprinting is the process of computing the hash value for a given binary code.
- **Forbidden Attack:** A forbidden attack is a type of man-in-the-middle attack used to hijack HTTPS sessions.
- **Firewall:** Firewalls are hardware and/or software designed to prevent unauthorized access to or from a private network.
- **Flooding:** The attacker sends loads of unnecessary traffic to produce noise, and if the IDS does not analyze the noise traffic well, then the true attack traffic may go undetected.
- **Firewalking:** Firewalking is a technique that uses TTL values to determine gateway ACL filters and it maps networks by analyzing the IP packet responses.
- **Frontjacking Attack:** Front jacking is a type of web server attack in which an attacker injects or manipulates the front-end components of a web application, such as scripts or HTML elements, to hijack a user interface or user interactions.
- **File Injection Attack:** A file injection attack is a technique used to exploit "dynamic file include" mechanisms in web applications.
- **Frequency-Hopping Spread Spectrum (FHSS):** FHSS, also known as frequency-hopping code-division multiple access (FH-CDMA), is a method of transmitting radio signals by rapidly switching a carrier among many frequency channels.
- **Fault Injection Attacks:** Fault injection attacks, also known as Perturbation attacks, occur when a perpetrator injects any faulty or malicious program into the system to compromise the system security.
- **Function-as-a-Service (FaaS):** This cloud computing service provides a platform for developing, running, and managing application functionalities without the complexity of building and maintaining necessary infrastructure (serverless architecture).
- **Firewalls-as-a-Service (FWaaS):** This cloud computing service protects users and organizations from both internal and external threats by filtering the network traffic.
- **Fog Computing:** Fog computing is a distributed and independent digital environment in which applications and data storage are positioned between data sources (devices generating data) and a cloud service.
- **Finney Attack:** A Finney attack is a type of blockchain attack that involves an attacker leveraging the time delays between the broadcasting and confirmation of transactions in cryptocurrency networks to reverse the transactions before they are confirmed.
- **51% Attack:** A 51% attack, also known as a majority attack, occurs when an attacker or group of attackers gains control of more than 50% of the computational power (hash rate) or stacking power in a blockchain network.

G

- **Gray Hats:** Gray hats are the individuals who work both offensively and defensively at various times.
- **Green Hat Hackers:** Green hat hackers are individuals motivated by the desire to become skilled professionals in the field of cybersecurity.
- **Gaining Access:** Gaining access refers to the point where the attacker obtains access to the operating system or applications on the target computer or network.
- **Google Hacking Database:** The Google Hacking Database (GHDB) is an authoritative source for querying the ever-widening reach of the Google search engine.
- **Golden Ticket Attack:** A golden ticket attack is a post-exploitation technique implemented by attackers to gain complete control over the entire Active Directory (AD).
- **Ghostwriting:** Ghostwriting is a bypass technique that involves modifying the structure of the malware code without effecting its functionality.
- **Global System for Mobile Communications (GSM):** It is a universal system used for mobile data transmission in wireless networks worldwide.
- **Golden SAML Attack:** Golden SAML attacks are performed to target identity providers on cloud networks such as the ADFS, which utilizes the SAML protocol for the authentication and authorization of users.
- **Government Access to Keys (GAK):** Government Access to Keys (GAK) refers to the statutory obligation of individuals and organizations to disclose their cryptographic keys to government agencies.
- **GOST Block Cipher:** The GOST (Government Standard) block cipher, also called Magma, is a symmetric-key block cipher having a 32-round Feistel network working on 64-bit blocks with a 256-bit key length.
- **GNU Privacy Guard:** GPG is a software replacement of PGP and free implementation of the OpenPGP standard.

H

- **Hacker Teams:** A consortium of skilled hackers having their own resources and funding. They work together in synergy for researching the state-of-the-art technologies.
- **Host-Based Indicators**: Host-based indicators are found by performing an analysis of the infected system within the organizational network.
- **Hacking**: Hacking refers to exploiting system vulnerabilities and compromising security controls to gain unauthorized or inappropriate access to a system's resources.
- **Hacker:** A hacker is a person who breaks into a system or network without authorization to destroy, steal sensitive data, or perform malicious attacks.
- **Hacktivist:** Individuals who promote a political agenda by hacking, especially by defacing or disabling websites.
- **HackerGPT**: HackerGPT is an AI-driven tool designed to assist ethical hackers in identifying vulnerabilities.
- **Health Insurance Portability and Accountability Act (HIPAA):** The HIPAA Privacy Rule provides federal protections for the individually identifiable health information held by covered entities and their business associates and gives patients an array of rights to that information.
- **Host-based Vulnerability Scanning**: Conducts a configuration-level check to identify system configurations, user directories, file systems, registry settings, etc., to evaluate possibility of compromise.
- **Hash Injection/Pass-the-Hash (PtH) Attack:** A hash injection/PtH attack allows an attacker to inject a compromised hash into a local session and use the hash to validate network resources.

- **Heap Spraying:** Heap spraying attack involves flooding the free space of a target process's memory heap by writing multiple copies of malicious code into specific memory locations by exploiting existing vulnerabilities such as buffer overflows.
- **Host Integrity Monitoring:** Host integrity monitoring involves taking a snapshot of the system state using the same tools before and after analysis, to detect changes made to the entities residing on the system.
- **Hardware Protocol Analyzer:** A hardware protocol analyzer is a piece of equipment that captures signals without altering the traffic in a cable segment.
- **Honey Trap:** The honey trap is a technique where an attacker targets a person online by pretending to be an attractive person and then begins a fake online relationship to obtain confidential information about the target company.
- **Hoax Letters:** Emails that issue warnings to the user about new viruses, Trojans, or worms that may harm the user's system.
- **HTTP GET/POST Attack:** In an HTTP GET attack, attackers use a time-delayed HTTP header to maintain HTTP connections and exhaust web server resources.
- **HTTP Strict Transport Security (HSTS):** HTTP Strict Transport Security (HSTS) is a web security policy that protects HTTPS websites against MITM attacks.
- **Honeypot:** A honeypot is an information system resource that is expressly set up to attract and trap people who attempt to penetrate an organization's network.
- **High-Interaction Honeypots:** Unlike their low- and medium-interaction counterparts, high-interaction honeypots do not emulate anything, they run actual vulnerable services or software on production systems with real OS and applications.
- **Honeynets:** Honeynets are networks of honeypots. They are very effective in determining the entire capabilities of the adversaries.
- **HTTP Response-Splitting Attack:** An HTTP response-splitting attack is a web-based attack in which the attacker tricks the server by injecting new lines into response headers, along with arbitrary code.
- **HTML Smuggling:** HTML smuggling is a type of web attack in which an attacker injects malicious code into a HTML script to compromise a web page.
- **HTTP/2 Continuation Flood Attack:** The HTTP/2 continuation flood attack involves exploiting the handling mechanism of HTTP/2 CONTINUATION frames to exhaust the target Apache server.
- **Hotfixes:** Hotfixes are an update to fix a specific customer issue and not always distributed outside the customer organization.
- **HTML Encoding:** An HTML encoding scheme is used to represent unusual characters so that they can be safely combined within an HTML document.
- **Hex Encoding:** The HTML encoding scheme uses the hex value of every character to represent a collection of characters for transmitting binary data.
- **Hotspot:** Hotspots refer to areas with Wi-Fi availability, where users can enable Wi-Fi on their devices and connect to the Internet.
- **Hybrid Cloud:** It is a cloud environment comprised of two or more clouds (private, public, or community) that remain unique entities but are bound together to offer the benefits of multiple deployment models.
- **HMAC:** HMAC is a type of message authentication code (MAC) that combines a cryptographic key with a cryptographic hash function.
- **Homomorphic Encryption:** Homomorphic encryption allows users to secure and leave their data in an encrypted format even while it is being processed or manipulated.

- **Hardware-Based Encryption:** Hardware-based encryption uses computer hardware for assisting or replacing the software when the data encryption process is underway.
- **HSM:** Hardware security module (HSM) is an additional external security device that is used in a system for crypto-processing and can be used for managing, generating, and securely storing cryptographic keys.
- **Hard Drive Encryption:** Hard drive encryption is a technology where the data stored in the hardware can be encrypted using a wide range of encryption options.
- **Hash Collision Attack:** A hash collision attack is performed by finding two different input messages that result in the same hash output.

I

- **Integrity:** The trustworthiness of data or resources in terms of preventing improper or unauthorized changes.
- **Information Warfare:** The term information warfare or InfoWar refers to the use of information and communication technologies (ICT) to gain competitive advantages over an opponent.
- **Indicators of Compromise (IoCs):** Indicators of Compromise (IoCs) are the clues, artifacts, and pieces of forensic data found on the network or operating system of an organization that indicate a potential intrusion or malicious activity in the organization's infrastructure.
- **Industrial Spies:** Individuals who perform corporate espionage by illegally spying on competitor organizations and focus on stealing information such as blueprints and formulas.
- **Information Assurance (IA):** IA refers to the assurance that the integrity, availability, confidentiality, and authenticity of information and information systems is protected during the usage, processing, storage, and transmission of information.
- **Incident Management:** Incident management is a set of defined processes to identify, analyze, prioritize, and resolve security incidents to restore normal service operations as quickly as possible and prevent future recurrence of the incident.
- **Incident Handling and Response:** Incident handling and response (IH&R) is the process of taking organized and careful steps when reacting to a security incident or cyberattack.
- **ISO/IEC 27701:2022:** Specifies the requirements and framework for establishing, implementing, maintaining, and continually improving an ISMS to ensure confidentiality, integrity, and availability of information.
- **ISO/IEC 27701:2019:** ISO/IEC 27701:2019 extends the ISO/IEC 27001 framework to include privacy management, specifically focusing on protecting personally identifiable information (PII).
- **ISO/IEC 27002:2022:** ISO/IEC 27002:2022 outlines the best practices and control objectives for critical cybersecurity areas such as access control, cryptography, and security personnel.
- **ISO/IEC 27005:2022:** ISO/IEC 27005:2022 provides comprehensive guidelines for information security risk management and supports the ISMS requirements specified in ISO/IEC 27001.
- **ISO/IEC 27032:2023:** ISO/IEC 27032:2023 explains the relationship among the Internet, Web, network security, and cybersecurity, providing a comprehensive overview of Internet security and identifying key stakeholders and their roles.
- **ISO/IEC 27040:2024:** ISO/IEC 27040:2024 provides the detailed technical requirements and guidance for achieving data storage security through careful planning, design, documentation, and implementation.
- **Impersonation:** Pretending to be a legitimate or authorized person and using the phone or other communication medium to mislead targets and trick them into revealing information.

- **ICMP ECHO Ping Scan:** ICMP ECHO ping scans involve sending ICMP ECHO requests to a host. If the host is live, it will return an ICMP ECHO reply.
- **ICMP ECHO Ping Sweep:** Ping sweep is used to determine the live hosts from a range of IP addresses by sending ICMP ECHO requests to multiple hosts. If a host is alive, it will return an ICMP ECHO reply.
- **ICMP Address Mask Ping Scan:** ICMP address mask ping is another alternative to the traditional ICMP ECHO ping, where the attackers send an ICMP address mask query to the target host to acquire information related to the subnet mask.
- **Inverse TCP Flag Scan:** Attackers send TCP probe packets with a TCP flag (FIN, URG, PSH) set or with no flags, where no response implies that the port is open, whereas an RST response means that the port is closed.
- **IDLE/IPID Header Scan:** The IDLE/IPID header scan is a TCP port scan method that can be used to send a spoofed source address to a computer to determine what services are available.
- **IP Address Decoy:** IP address decoy technique refers to generating or manually specifying the IP addresses of decoys in order to evade an IDS or firewall.
- **IP Address Spoofing:** IP spoofing refers to changing the source IP addresses so that the attack appears to be coming from someone else.
- **Internal Vulnerability Scanning:** Internal scanning involves scrutinizing the internal network to find exploits and vulnerabilities.
- **Integer Overflow:** An integer overflow occurs when an arithmetic function generates and attempts to store an integer value larger than the maximum value that the allocated memory space can store.
- **Image Steganography:** In image steganography, the information is hidden in image files of different formats such as .PNG, .JPG, and .BMP.
- **Injector:** A program that injects its code into other vulnerable running processes and changes how they execute to hide or prevent its removal.
- **IRDP Spoofing:** The attacker sends a spoofed IRDP router advertisement message to the host on the subnet, causing it to change its default router to whatever the attacker chooses.
- **Insider Attack:** An insider attack involves using privileged access to intentionally violate rules or cause threat to the organization's information or information systems in any form.
- **Identity Theft:** Identity theft is a crime in which an imposter steals your personally identifiable information such as name, credit card number, social security or driver's license numbers, etc. to commit fraud or other crimes.
- **ICMP Flood Attack:** ICMP flood attacks are a type of attack in which attackers send large volumes of ICMP echo request packets to a victim system directly or through reflection networks.
- **Ingress Filtering:** Ingress filtering prevents the source address spoofing of Internet traffic.
- **IPSec:** IPSec is a protocol suite developed by the IETF for securing IP communications by authenticating and encrypting each IP packet of a communication session.
- **IoT Device Vulnerability Scanning:** IoT device vulnerability scanning provides insights into weaknesses across IoT devices and systems that are exposed to or connected to the Internet.
- **Intrusion Detection System (IDS):** An intrusion detection system (IDS) is a software system or hardware device that inspects all inbound and outbound network traffic for suspicious patterns that may indicate a network or system security breach.
- **Intrusion Prevention System (IPS):** IPS are continuous monitoring systems that often sit behind firewalls as an additional layer of protection.

- **Insertion Attack:** Insertion is the process by which the attacker confuses the IDS by forcing it to read invalid packets.
- **Injection Flaws:** Injection flaws are web application vulnerabilities that allow untrusted data to be interpreted and executed as part of a command or query.
- **In-band SQL Injection:** An attacker uses the same communication channel to perform the attack and retrieve the results.
- **Input Validation:** Input validation helps developers to prevent user-supplied data influencing the logic of the code.
- **Industrial, Scientific, and Medical (ISM) Band:** This band is a set of frequencies used by the international industrial, scientific, and medical communities.
- **Inter-Chip Privilege Escalation Attack:** The inter-chip privilege escalation attack exploits the underlying vulnerabilities in wireless chips that handle wireless communications such as Bluetooth and Wi-Fi.
- **iOS Trustjacking:** iOS Trustjacking is a vulnerability that can be exploited by an attacker to read messages and emails and capture sensitive information from a remote location without the victim's knowledge.
- **iOS Method Swizzling:** Method swizzling, also known as monkey patching, is a technique that involves modifying the existing methods or adding new functionality at runtime.
- **IoT:** Internet of Things (IoT), also known as Internet of Everything (IoE), refers to the network of devices having IP addresses and the capability to sense, collect, and send data using embedded sensors, communication hardware and processors.
- **IoT Device Management:** IoT device management helps in supporting IoT solutions by using any software tools and processes and helps in onboarding latest devices securely and promptly.
- **Industrial Network:** A network of automated control systems is known as an industrial network.
- **Industrial Protocols:** Protocols used for serial communication and communication over standard Ethernet. Ex: S7, CDA, CIP, Modbus, etc.
- **IT/OT Convergence (IIOT):** IT/OT convergence is the integration of IT computing systems and OT operation monitoring systems to bridge the gap between IT/OT technologies for improving overall security, efficiency, and productivity.
- **ICS:** ICS is often referred to as a collection of different types of control systems and their associated equipment such as systems, devices, networks, and controls used to operate and automate several industrial processes.
- **Infrastructure-as-a-Service (IaaS):** This service provides virtual machines and other abstracted hardware and operating systems (OSs), which may be controlled through a service application programming interface (API).
- **Identity-as-a-Service (IDaaS):** This cloud computing service offers authentication services to the subscribed enterprises and is managed by a third-party vendor to provide identity and access management services.

J

- **JIT Spraying:** Attackers use just-in-time (JIT) spraying techniques to execute arbitrary code on a victim's system by exploiting vulnerabilities in the JIT compilation feature in many modern web browsers.
- **Jailbreaking:** Jailbreaking is defined as the process of installing a modified set of kernel patches that allows users to run third-party applications not signed by the OS vendor.
- **Jamming Attack:** Jamming is a type of attack in which the communications between wireless IoT devices are jammed so that they can be compromised.

K

- **Kerberos:** Kerberos is a network authentication protocol that provides strong authentication for client/server applications through secret-key cryptography.
- **Kerberoasting (Cracking TGS):** Kerberoasting is an attack technique that targets the Kerberos authentication protocol to obtain and crack the password hashes of service accounts in an Active Directory environment.
- **Kernel Exploits:** Kernel exploits refer to programs that can exploit vulnerabilities present in the kernel to execute arbitrary commands or code with higher privileges.
- **Keylogger:** Keystroke loggers are programs or hardware devices that monitor each keystroke as the user types on a keyboard, logs onto a file, or transmits them to a remote location.
- **Kubernetes:** Kubernetes, also known as K8s, is an open-source, portable, extensible, orchestration platform developed by Google for managing containerized applications and microservices.
- **Key Stretching:** Key stretching refers to the process of strengthening a key that might be slightly too weak, usually by making it longer.

L

- **LDAP:** Lightweight directory access protocol (LDAP) is an Internet protocol for accessing distributed directory services.
- **Lawful Interception:** Lawful interception refers to legally intercepting data communication between two end points for surveillance on the traditional telecommunications, Voice over Internet Protocol (VoIP), data, and multiservice networks.
- **Low-interaction Honeypots:** Low-interaction honeypots emulate only a limited number of services and applications of a target system or network.
- **LDAP Injection Attack:** An LDAP injection attack works in the same way as an SQL injection attack, but it exploits user parameters to generate an LDAP query.
- **LPWAN:** Low Power Wide Area Networking (LPWAN) is a wireless telecommunication network, designed to provide long-range communications between two endpoints.
- **LWM2M:** Lightweight Machine-to-Machine (LWM2M) is an application-layer communication protocol used for application-level communication between IoT devices; it is used for IoT device management.
- **Living Off the Cloud Attack (LotC):** Living Off the Cloud (LotC) is a modern evolution of the "living off the land" attack, in which attackers target victim's SaaS and IaaS-based applications to carry out malicious activities such as data exfiltration.

M

- **MITRE ATT&CK Framework:** MITRE ATT&CK is a globally accessible knowledge base of adversary tactics and techniques based on real-world observations.
- **Maintaining Access:** Maintaining access refers to the phase when the attacker tries to retain their ownership of the system.
- **Management Information Base (MIB):** MIB is a virtual database containing a formal description of all the network objects that can be managed using SNMP.
- **Manual Vulnerability Scanning:** Manual vulnerability scanning refers to the process of manually identifying, evaluating, and validating security vulnerabilities in systems, networks, and applications.
- **Mask Attack:** Mask attack is similar to brute-force attacks but recovers passwords from hashes with a more specific set of characters based on information known to the attacker.

- **Mobile Application Scanning:** Mobile application scanning aims at protecting the privacy of data across mobile applications and APIs.
- **Markov-Chain Attack:** Attackers gather a password database and split each password entry into 2- and 3-character long syllables; using these character elements, a new alphabet is developed, which is then matched with the existing password database.
- **Malware:** Malware is malicious software that damages or disables computer systems and gives limited or full control of the systems to the malware creator for the purpose of theft or fraud.
- **Malicious Code:** A command that defines malware's basic functionalities such as stealing data and creating backdoors.
- **Malware Analysis:** Malware analysis is a process of reverse engineering a specific piece of malware to determine the origin, functionality, and potential impact of a given type of malware.
- **MAC Flooding:** MAC flooding involves the flooding of the CAM table with fake MAC address and IP pairs until it is full.
- **MAC Spoofing/Duplicating:** A MAC duplicating attack is launched by sniffing a network for MAC addresses of clients who are actively associated with a switch port and re-using one of those addresses.
- **Malicious Insider:** A disgruntled or terminated employee who steals data or destroys the company's networks intentionally by introducing malware into the corporate network.
- **Multi-Vector Attack:** In multi-vector DDoS attacks, the attackers use combinations of volumetric, protocol, and application-layer attacks to disable the target system or service.
- **Man-in-the-Middle/Manipulator-in-the-Middle Attack:** The man-in-the-middle attack is used to intrude into an existing connection between systems and intercept the messages being exchanged.
- **Man-in-the-Browser/Manipulator-in-the-Browser Attack:** The man-in-the-browser attack uses a Trojan horse to intercept the calls between the browser and its security mechanisms or libraries.
- **Medium-interaction Honeypots:** Medium-interaction honeypots simulate a real OS as well as applications and services of a target network.
- **Malware Honeypots:** Malware honeypots are used to trap malware campaigns or malware attempts over the network infrastructure.
- **MarioNet Attack:** MarioNet is a browser-based attack that runs malicious code inside the browser, and the infection persists even after closing or browsing away from the malicious webpage through which infection has spread.
- **Manual Web Application Security Testing:** It involves testing a web application using manually designed data, customized code, and some browser extension tools to detect vulnerabilities and weaknesses associated with the applications.
- **Mobile Spam:** Mobile phone spam, also known as SMS spam, text spam, or m-spam, refers to unsolicited messages sent in bulk form to known/unknown phone numbers/email IDs to target mobile phones.
- **Mobile Device Management (MDM):** Mobile Device Management (MDM) provides platforms for over-the-air or wired distribution of applications and data and configuration settings for all types of mobile devices, including mobile phones, smartphones, and tablet computers.
- **MQTT:** Message Queuing Telemetry Transport (MQTT) is an ISO standard lightweight protocol used to transmit messages for long-range wireless communication.
- **Multimedia over Coax Alliance (MoCA):** MoCA is a type of network protocol that provides high-definition videos and related content to homes over existing coaxial cables.

- **Mobile Backend-as-a-Service (MBaaS):** This cloud computing service allows app developers to integrate their front-end applications with backend infrastructure through an application programming interface (API) and software development kit (SDK).
- **Multi Cloud:** It is a dynamic heterogeneous environment that combines workloads across multiple cloud vendors that are managed via one proprietary interface to achieve long-term business goals.
- **Microservices:** Monolithic applications are broken down into cloud-hosted sub-applications called microservices that work together, each performing a unique task.
- **Man-in-the-Cloud (MITC) Attack:** MITC attacks are performed by abusing cloud file synchronization services such as Google Drive or Drop Box for Data compromise, command and control (C&C), data exfiltration, and remote access.
- **MD5:** The MD5 algorithm takes a message of arbitrary length as the input and then outputs a 128-bit fingerprint or message digest of the input.
- **MD6:** MD6 uses a Merkle-tree-like structure to allow for large-scale parallel computation of hashes for very long inputs.
- **Multilayer Hashing Calculators:** Multilayer hashing, also known as nested hashing or recursive hashing, is a technique in which a hash function is applied multiple times to the input or output of a previous hash operation.

N

- **Non-Repudiation:** A guarantee that the sender of a message cannot later deny having sent the message and that the recipient cannot deny having received the message.
- **Network Indicators:** Network indicators are useful for command and control, malware delivery, identifying the operating system, and other tasks.
- **Network Scanning:** Network scanning refers to a set of procedures used for identifying hosts, ports, and services in a network.
- **NTP:** Network Time Protocol (NTP) is designed to synchronize the clocks of networked computers.
- **National Vulnerability Database (NVD):** A U.S. government repository of standards-based vulnerability management data represented using the Security Content Automation Protocol (SCAP).
- **Network-based Vulnerability Scanning:** Determines possible network security attacks that may occur on the organization's system.
- **Non-Credentialed/Unauthenticated Vulnerability Scanning:** Non-credentialed vulnerability scanning is a security testing method that assesses systems, networks, and applications without using valid credentials to log into the target system.
- **NTLM Relay Attack:** An NTLM relay attack involves an attacker intercepting and relaying NTLM authentication requests between a client and server to impersonate the client and gain unauthorized access.
- **NTFS Data Stream:** NTFS Alternate Data Stream (ADS) is a Windows hidden stream, which contains metadata for the file, such as attributes, word count, author name and access, and modification time of the files.
- **Natural Language Processing (NLP):** Natural language processing (NLP) is a branch of artificial intelligence that focuses on the interaction between computers and humans through natural language.
- **Network Level Hijacking:** Network level hijacking can be defined as the interception of packets during the transmission between a client and the server in a TCP or UDP session.

- **Network Address Translation (NAT):** Network address translation separates IP addresses into two sets and enables the LAN to use these addresses for internal and external traffic separately.
- **Near-Field Communication (NFC):** NFC is a type of short-range communication that uses magnetic field induction to enable communication between two electronic devices.
- **NAND Glitching:** NAND glitching is the process of gaining privileged root access while booting a device, which can be performed by making a ground connection to the serial I/O pin of a flash memory chip.
- **Next-Generation Secure Web Gateway (NG SWG):** NG SWG is a cloud-based security solution that protects an organization's network from cloud-based threats, malware infections, and data theft activities.

O

- **Organized Hackers:** Miscreants or hardened criminals who use rented devices or botnets to perform various cyber-attacks to pilfer money from victims.
- **OS Discovery/Banner Grabbing:** Banner grabbing or OS fingerprinting is the method used to determine the operating system running on a remote target system.
- **Overpass-the-Hash Attack:** It is a type of credential theft-and-reuse attack using which attackers perform malicious activities on compromised devices or environments.
- **Obfuscator:** A program that conceals its code and intended purpose via various techniques, and thus, makes it hard for security mechanisms to detect or remove it.
- **Obfuscating:** Obfuscating is an IDS evasion technique used by attackers who encode the attack packet payload in such a way that the destination host can decode the packet but not the IDS.
- **OAuth:** OAuth is an authorization protocol that allows a user to grant limited access to their resources on a site to a different site without having to expose their credentials.
- **Output Encoding:** Output encoding is used to encode the input to ensure it is properly sanitized before being passed to the database.
- **Orthogonal Frequency-Division Multiplexing (OFDM):** An OFDM is a method of digital modulation of data in which a signal, at a chosen frequency, is split into multiple carrier frequencies that are orthogonal (occurring at right angles) to each other.
- **Omnidirectional Antenna:** Omnidirectional antennas radiate electromagnetic (EM) energy in all directions.
- **OTP Hijacking:** Attackers hijack OTPs and redirect them to their personal devices using different techniques such as social engineering and SMS jacking.
- **OT:** Operational Technology (OT) is the software and hardware designed to detect or cause changes in industrial operations through direct monitoring and/or controlling of industrial physical devices.

P

- **PoisonGPT:** PoisonGPT is an AI-powered tool that introduces malicious models into otherwise trusted AI systems.
- **PentestGPT:** PentestGPT was designed to assist penetration testers by automating various aspects of the testing process.
- **Passive Attacks:** Passive attacks involve intercepting and monitoring network traffic and data flow on the target network and do not tamper with the data.
- **Procedures:** "Procedures" are organizational approaches that threat actors follow to launch an attack.
- **Payment Card Industry Data Security Standard (PCI DSS):** The Payment Card Industry Data Security Standard (PCI DSS) is a proprietary information security standard for organizations that handle cardholder information for major debit, credit, prepaid, e-purse, ATM, and POS cards.

- **Passive Footprinting:** Passive footprinting involves gathering information about the target without direct interaction.
- **Packet Fragmentation:** Packet fragmentation refers to the splitting of a probe packet into several smaller packets (fragments) while sending it to a network.
- **Proxy Server:** A proxy server is an application that can serve as an intermediary for connecting with other computers.
- **Physical Security Vulnerability Scanning:** Physical security vulnerability scanning involves conducting a comprehensive examination of physical assets to proactively identify various vulnerabilities associated with them.
- **Password Cracking:** Attackers use password cracking techniques to gain unauthorized access to vulnerable systems .
- **Password Guessing:** Password guessing is a password-cracking technique that involves attempting to log on to the target system with different passwords manually.
- **Password Spraying Attack:** Password spraying attack targets multiple user accounts simultaneously using one or a small set of commonly used passwords.
- **Pass-the-Ticket Attack:** Pass the Ticket is a technique used for authenticating a user to a system that is using Kerberos without providing the user's password.
- **PRINCE Attack:** An advanced version of a combinator attack where instead of taking input from two different dictionaries, attackers use a single input dictionary to build chains of combined words.
- **Password Salting:** Password salting is a technique where a random string of characters are added to the password before calculating their hashes.
- **Proof-of-Concept (PoC):** Proof-of-concept (PoC) is the demonstration of the existence and impact of a vulnerability in software or networks.
- **Privilege Escalation:** A privilege escalation attack is the process of gaining more privileges than were initially acquired.
- **Pivoting:** Attackers use the pivoting technique to compromise a system, gain remote shell access on it, and further bypass the firewall to pivot via the compromised system to access other vulnerable systems in the network.
- **Point-of-Sale Trojans:** Point-of-sale (POS) Trojans are a type of financial fraudulent malware that target POS and payment equipment such as credit card/debit card readers.
- **Packer:** A program that allows all files to bundle together into a single executable file via compression to bypass security software detection.
- **Payload:** A piece of software that allows control over a computer system after it has been exploited.
- **Potentially Unwanted Application or Applications (PUAs):** Also known as grayware or junkware, are potentially harmful applications that may pose severe risks to the security and privacy of data stored in the system where they are installed.
- **Portable Executable (PE):** The Portable Executable (PE) format is an executable file format used on Windows OS, which stores the information that a Windows system requires to manage the executable code.
- **Packet Sniffing:** Packet sniffing is the process of monitoring and capturing all data packets passing through a given network using a software application or hardware device.
- **Passive Sniffing:** It involves monitoring packets sent by others without sending any additional data packets in the network traffic.

- **Piggybacking:** Piggybacking usually implies entry into a building or security area with the consent of the authorized person.
- **Pop-Up Windows:** Windows that suddenly pop up while surfing the Internet and ask for user information to login or sign-in.
- **Phishing:** Phishing is the practice of sending an illegitimate email claiming to be from a legitimate site in an attempt to acquire a user's personal or account information.
- **Pharming:** Pharming is a social engineering technique in which the attacker executes malicious programs on a victim's computer or server, and when the victim enters any URL or domain name, it automatically redirects the victim's traffic to an attacker-controlled website.
- **Ping of Death Attack:** In a Ping of Death (PoD) attack, an attacker tries to crash, destabilize, or freeze the targeted system or service by sending malformed or oversized packets using a simple ping command.
- **Pulse Wave DDoS Attack:** In a pulse wave DDoS attack, attackers send a highly repetitive, periodic train of packets as pulses to the target victim every 10 minutes, and each specific attack session can last for a few hours to days.
- **Peer-to-Peer Attack:** A peer-to-peer attack is a form of DDoS attack in which the attacker exploits a number of bugs in peer-to-peer servers to initiate a DDoS attack.
- **Permanent Denial-of-Service Attack:** Permanent DoS, also known as phlashing, refers to attacks that cause irreversible damage to system hardware.
- **Protocol Anomaly Detection:** In this type of detection, models are built to explore anomalies in the way in which vendors deploy the TCP/IP specification.
- **Packet Filtering Firewall:** In a packet filtering firewall, each packet is compared to a set of criteria before it is forwarded.
- **Pure Honeypots:** Pure honeypots emulate the real production network of a target organization.
- **Production Honeypots:** Production honeypots are deployed inside the production network of the organization along with other production servers.
- **Port Scanning:** Port scanning is used to identify open ports and the services running on these ports.
- **Patch:** A patch is a small piece of software designed to fix problems, security vulnerabilities, and bugs and improve the performance of a computer program or its supporting data.
- **Pass-the-Cookie Attack:** The pass-the-cookie attack occurs when attackers obtain a clone of a cookie from the user's browser and uses the cookie to establish a session with the target web server.
- **Parabolic Grid Antenna:** A parabolic grid antenna uses the same principle as a satellite dish, but it does not have a solid dish. It consists of a semi-dish in the form of a grid consisting of aluminum wires.
- **Purdue Model:** The Purdue model is derived from the Purdue Enterprise Reference Architecture (PERA) model, which is a widely used to describe internal connections and dependencies of important components in the ICS networks.
- **Programmable Logic Controller (PLC):** A programmable logic controller (PLC) is a small solid-state control computer where instructions can be customized to perform a specific task.
- **Platform-as-a-Service (PaaS):** This offers development tools, configuration management, and deployment platforms on-demand, which can be used by subscribers to develop custom applications.
- **Public Cloud:** In this model, the provider makes services such as applications, servers, and data storage available to the public over the Internet.
- **Private Cloud:** A private cloud, also known as the internal or corporate cloud, is a cloud infrastructure operated by a single organization and implemented within a corporate firewall.

- **Post-quantum Cryptography:** Post-quantum cryptography is an advanced cryptographic algorithm designed to protect security systems from attacks initiated on both conventional and quantum computers.
- **Public Key Infrastructure (PKI):** PKI is a set of hardware, software, people, policies, and procedures required to create, manage, distribute, use, store, and revoke digital certificates.
- **Pretty Good Privacy (PGP):** It is often used for data compression, digital signing, encryption and decryption of messages, emails, files, and directories, and to enhance the privacy of email communications.
- **Padding Oracle Attack:** In a padding oracle attack (also known as a Vaudenay attack), attackers exploit the padding validation of an encrypted message to decipher the ciphertext.

Q

- **QRLJacking:** QRLJacking is a type of social engineering attack that exploits the QR Code Login method in various web applications to hijack login sessions and gain unauthorized access to victims' accounts.
- **Quantum Cryptography:** This cryptography is processed based on quantum mechanics, such as quantum key distribution (QKD), using photons instead of mathematics as a part of encryption.
- **Quantum Cryptanalysis:** Quantum cryptanalysis is the process of cracking cryptographic algorithms using a quantum computer.

R

- **Red Hat Hackers:** Red hats adopt aggressive tactics, such as black hat hackers, with the intent of neutralizing threats before damaging resources.
- **Reconnaissance**: Reconnaissance refers to the preparatory phase where an attacker seeks to gather information about a target prior to launching an attack.
- **Risk:** Risk refers to the degree of uncertainty or expectation that an adverse event may cause damage to the system.
- **Risk Matrix:** The risk matrix scales the risk occurrence or likelihood probability, along with its consequences or impact.
- **Risk Management**: Risk management is the process of reducing and maintaining risk at an acceptable level by means of a well-defined and actively employed security program.
- **Risk Identification:** Identifies the sources, causes, consequences, and other details of the internal and external risks affecting the security of the organization.
- **Risk Assessment:** Assesses the organization's risk and provides an estimate of the likelihood and impact of the risk.
- **Risk Treatment:** Selects and implements appropriate controls for the identified risks.
- **Risk Tracking:** Ensures appropriate controls are implemented to handle known risks and calculates the chances of a new risk occurring.
- **Return-Oriented Programming (ROP) Attack:** Return-oriented programming is an exploitation technique used by attackers to execute arbitrary malicious code in the presence of security protections such as code signing and executable space protection.
- **RPC**: Remote Procedure Call (RPC) allows clients and servers to communicate in distributed client/server programs.
- **Resource Exhaustion:** A resource exhaustion attack damages the server by sending multiple resource requests from different locations to exploit software bugs or errors, thereby hanging the system and server or causing a system crash.

- **Race Condition:** A race condition is an undesirable incident that occurs when a software or system program depends on the execution of processes in a sequence and on the timing of the programs.
- **Replay Attack:** In a replay attack, packets and authentication tokens are captured using a sniffer. After the relevant information is extracted, the tokens are placed back on the network to gain access.
- **Rainbow Table:** A rainbow table is a precomputed table that contains word lists like dictionary files, brute force lists, and their hash values.
- **Relaying:** Attackers use the relaying technique to access resources present on other systems via the compromised system such a way that the requests to access the resources are coming from the initially compromised system.
- **Rootkits:** Rootkits are programs that hide their presence as well as attacker's malicious activities, granting them full access to the server or host at that time, and in the future.
- **Rich Text Format (RTF) Injection:** RTF injection involves exploiting features of Microsoft Office such as RTF template files that are stored locally or in a remote machine.
- **Ransomware:** Ransomware is a type of malware that restricts access to the computer system's files and folders and demands an online ransom payment to the malware creator(s) to remove the restrictions.
- **Rogue DHCP Server Attack:** The attacker sets up a rogue DHCP server on the network and responds to DHCP requests with bogus IP addresses resulting in compromised network access.
- **Reverse Social Engineering:** The attacker presents him/herself as an authority and the target seeks his or her advice before or after offering the information that the attacker needs.
- **Reverse Tabnabbing:** Reverse tabnabbing involves a seemingly legitimate website that deceives users into opening a malicious link, which then alters the content of the original tab to a phishing site.
- **RST Hijacking:** RST hijacking involves injecting an authentic-looking reset (RST) packet using a spoofed source address and predicting the acknowledgment number.
- **Research Honeypots:** Research honeypots are high-interaction honeypots primarily deployed by research institutes, governments, or military organizations to gain detailed knowledge about the actions of intruders.
- **RASP:** Runtime application self-protection (RASP) provides security to web and non-web application running on a server.
- **Reflector Antennas:** Reflector antennas are used to concentrate EM energy that is radiated or received at a focal point.
- **RFID Cloning Attack:** RFID cloning involves capturing the data from a legitimate RFID tag and then creating its clone using a new chip.
- **Reverse Engineering:** Reverse engineering is the process of analyzing and extracting the source code of a software or application, and if needed, regenerating it with required modifications.
- **RC4:** RC4 is a variable key-size symmetric-key stream cipher with byte-oriented operations, and it is based on the use of a random permutation.
- **RC5:** RC5 is a fast symmetric-key block cipher designed by Ronald Rivest for RSA Data Security (now RSA Security).
- **RC6:** RC6 is a symmetric-key block cipher derived from RC5. It is a parameterized algorithm with a variable block size, key size, and number of rounds.
- **Rivest Shamir Adleman (RSA):** Ron Rivest, Adi Shamir, and Leonard Adleman formulated RSA, a public-key cryptosystem for Internet encryption and authentication.
- **RIPEMD-160:** RACE Integrity Primitives Evaluation Message Digest (RIPEMD) is a 160-bit hash algorithm developed by Hans Dobbertin, Antoon Bosselaers, and Bart Preneel.

- **Rainbow Table Attack:** A rainbow table attack is a type of cryptography attack where an attacker uses a rainbow table to reverse cryptographic hash functions.
- **Related-Key Attack:** An attacker launch a related key attack by exploiting the mathematical relationship between keys in a cipher to gain access over encryption and decryption functions.
- **Race Attack:** A race attack is a double-spending attack that exploits the delay in transaction confirmation in blockchain networks to obtain goods or services without actually paying for them and effectively spends the same coin twice.

S

- **Suicide Hackers:** Suicide hackers are individuals who aim to bring down critical infrastructure for a "cause" and are not worried about facing jail terms or any other kind of punishment.
- **Script Kiddies:** Script kiddies are unskilled hackers who compromise systems by running scripts, tools, and software developed by real hackers.
- **State-Sponsored Hackers:** State-sponsored hackers are individuals employed by the government to penetrate, gain top-secret information from, and damage the information systems of other governments.
- **ShellGPT:** An AI-powered tool that ethical hackers and cybersecurity professionals can use to perform various tasks.
- **Strategic Threat Intelligence:** Strategic threat intelligence provides high-level information regarding cybersecurity posture, threats, details about the financial impact of various cyber activities, attack trends, and the impact of high-level business decisions.
- **Scanning:** Scanning refers to the pre-attack phase when the attacker scans the network for specific information based on information gathered during reconnaissance.
- **Supervised Learning:** Supervised learning uses algorithms that input a set of labeled training data to attempt to learn the differences between the given labels.
- **Sarbanes Oxley Act (SOX):** Enacted in 2002, the Sarbanes-Oxley Act aims to protect the public and investors by increasing the accuracy and reliability of corporate disclosures.
- **Shoulder Surfing:** In the shoulder surfing technique, an attacker stands behind the victim and secretly observes the victim's activities on the computer, such as keystrokes while entering usernames, passwords, and so on.
- **Stealth Scan (Half-open Scan):** Stealth scanning involves abruptly resetting the TCP connection between the client and server before the completion of three-way handshake signals, thus leaving the connection half-open.
- **SCTP INIT Scanning:** Attackers send an INIT chunk to the target host, and an INIT+ACK chunk response implies that the port is open, whereas an ABORT Chunk response means that the port is closed.
- **SCTP COOKIE ECHO Scanning:** Attackers send a COOKIE ECHO chunk to the target host, and no response implies that the port is open, whereas an ABORT Chunk response means that the port is closed.
- **Source Routing:** Source routing refers to sending a packet to the intended destination with a partially or completely specified route (without firewall-/IDS-configured routers) in order to evade an IDS or firewall.
- **Source Port Manipulation:** Source port manipulation refers to manipulating actual port numbers with common port numbers in order to evade an IDS or firewall.
- **SNMP Enumeration:** SNMP enumeration is the process of enumerating user accounts and devices on a target system using SNMP.
- **SSDP:** Simple Service Discovery Protocol (SSDP) is a network protocol that generally communicates with machines when querying them with routable IPv4 or IPv6 multicast addresses.

- **SMB:** Server Message Block (SMB) is a transport protocol that is generally used by Windows systems for providing shared access to files, printers, and serial ports as well as remote access to Windows services.
- **Security Accounts Manager (SAM) Database:** Windows uses the Security Accounts Manager (SAM) database or Active Directory Database to manage user accounts and passwords in hashed format (a one-way hash).
- **Spyware:** Spyware is a stealthy program that records the user's interaction with the computer and the Internet without the user's knowledge and sends the information to the remote attackers.
- **Screen-Capturing Spyware:** Screen-capturing spyware is a program that allows you to monitor computer activities by taking snapshots or screenshots of the computer on which the program is installed.
- **Steganography:** Steganography is a technique of hiding a secret message within an ordinary message and extracting it at the destination to maintain confidentiality of data.
- **Spam/Email Steganography:** Spam/email steganography refers to the technique of sending secret messages by hiding them in spam/email messages.
- **Steganalysis:** Steganalysis is the art of discovering and rendering covert messages using steganography.
- **Skeleton Key Attack:** A skeleton key is a form of malware that attackers use to inject false credentials into domain controllers (DCs) to create a backdoor password.
- **Silver Ticket Attack:** A silver ticket attack is a post-exploitation technique implemented by an attacker to steal legitimate users' credentials and create a fake Kerberos Ticket Granting Service (TGS) ticket.
- **Sheep Dip Computer:** Sheep dipping refers to the analysis of suspect files, incoming messages, etc. for malware.
- **Static Malware Analysis:** It involves going through the executable binary code without executing it to have a better understanding of the malware and its purpose.
- **SNMP:** Simple Network Management Protocol (SNMP) is a TCP/IP-based protocol used for exchanging management information between devices connected on a network.
- **SMTP:** Simple Mail Transfer Protocol (SMTP) is used for transmitting email messages over the Internet.
- **System Baselining:** Baselining refers to the process of capturing the system state (taking a snapshot of the system) when the malware analysis begins, which can be compared with the system's state after executing the malware file.
- **SPAN Port:** A SPAN port is a port that is configured to receive a copy of every packet that passes through a switch.
- **STP Attack:** Attackers connect a rogue switch into the network to change the operations of the STP protocol and sniff all the network traffic.
- **SAD DNS Attack:** SAD DNS is a new variant of DNS cache poisoning, in which an attacker injects harmful DNS records into a DNS cache to divert all traffic toward their own servers.
- **Social Engineering:** Social engineering is the art of convincing people to reveal confidential information.
- **Spam Email:** Irrelevant, unwanted, and unsolicited emails that attempt to collect financial information, social security numbers, and network information.
- **Scareware:** Malware that tricks computer users into visiting malware infested websites, or downloading/buying potentially malicious software.
- **Spear Phishing:** Attackers send spear phishing to send a message with specialized, social engineering content directed at a specific person, or a small group of people.
- **Spimming:** A variant of spam that exploits Instant Messaging platforms to flood spam across the networks.

- **SMiShing:** SMiShing (SMS phishing) is the act of using SMS text messaging system of cellular phones or other mobile devices to lure users into instant action, such as downloading malware, visiting a malicious webpage, or calling a fraudulent phone number.
- **Smurf Attack:** In a Smurf attack, the attacker spoofs the source IP address with the victim's IP address and sends a large number of ICMP ECHO request packets to an IP broadcast network.
- **SYN Flood Attack:** In a SYN attack, the attacker sends a large number of SYN requests to the target server (victim) with fake source IP addresses.
- **Spoofed Session Flood Attack:** Attackers create fake or spoofed TCP sessions by carrying multiple SYN, ACK, and RST or FIN packets.
- **Session Hijacking:** Session hijacking refers to an attack in which an attacker seizes control of a valid TCP communication session between two computers.
- **Signature Recognition:** Signature recognition, also known as misuse detection, tries to identify events that indicate an abuse of a system or network resource.
- **Stateful Multilayer Inspection Firewall:** Stateful multilayer inspection firewalls combine the aspects of the other three types of firewalls (Packet Filtering, Circuit-Level Gateways, and Application-Level Firewalls).
- **Spam Honeypots:** Spam honeypots specifically target spammers who abuse vulnerable resources such as open mail relays and open proxies.
- **Spider Honeypots:** Spider honeypots are also called spider traps. These honeypots are specifically designed to trap web crawlers and spiders.
- **Session Splicing:** Session splicing is a technique used to bypass the IDS where an attacker splits the attack traffic into many packets such that no single packet triggers the IDS.
- **SSH Brute Force Attack:** Attackers use SSH protocols to create an encrypted SSH tunnel between two hosts to transfer unencrypted data over an insecure network.
- **Same-Site Attack:** Same-site attacks, also known as related-domain attacks, occur when an attacker targets a subdomain of a trusted organization and attempts to redirect users to an attacker-controlled web page.
- **Static Application Security Testing (SAST):** It is also referred to as a white-box testing approach, in which the complete system architecture (including its source code) or application/software to be tested is already known to the tester.
- **Source Code Review:** Source code reviews are used to detect bugs and irregularities in the developed web applications.
- **16-bit Unicode Encoding:** It replaces unusual Unicode characters with "%u" followed by the character's Unicode code point expressed in hexadecimal.
- **SQL Injection:** SQL injection is a technique used to take advantage of un-sanitized input vulnerabilities to pass SQL commands through a web application for execution by a backend database.
- **Service Set Identifier (SSID):** An SSID is a 32-alphanumeric-character unique identifier given to a wireless local area network (WLAN) that acts as a wireless identifier of the network.
- **Simjacker:** Simjacker is a vulnerability associated with a SIM card's S@T browser (SIMalliance Toolbox Browser), a pre-installed software incorporated in SIM cards to provide a set of instructions.
- **Sybil Attack:** The attacker uses multiple forged identities to create a strong illusion of traffic congestion, affecting communication between neighboring nodes and networks.
- **Side-Channel Attack:** The attacker extracts information about encryption keys by observing the emission of signals i.e. "side channels" from IoT devices.

- **Supervisory Control and Data Acquisition (SCADA):** SCADA is a centralized supervisory control system that is used for controlling and monitoring industrial facilities and infrastructure.
- **Safety Instrumented Systems (SIS):** An SIS is an automated control system designed to safeguard the manufacturing environment in case of any hazardous incident in the industry.
- **Software-as-a-Service (SaaS):** This cloud computing service offers application software to subscribers on-demand over the Internet.
- **Security-as-a-Service (SECaaS):** It provides services such as penetration testing, authentication, intrusion detection, anti-malware, security incident and event management.
- **Serverless Computing:** Serverless computing also known as serverless architecture or Function-as-a-Service (FaaS), is a cloud-based application architecture where application infrastructure and supporting services are provided by the cloud vendor as they are needed.
- **S3 Buckets:** Simple storage service (S3) is a scalable cloud storage service used by Amazon AWS, where files, folders, and objects are stored via Web APIs.
- **SAML:** Security Assertion Markup Language (SAML) is a popular open-standard protocol used for authentication and authorization between communicating parties.
- **Security Groups:** It is a basic security measure implemented in cloud infrastructure to provide security to virtual instances.
- **Symmetric Encryption:** Symmetric encryption (secret-key, shared-key, and private-key) uses the same key for encryption as it does for decryption.
- **Serpent:** Serpent uses a 128-bit symmetric block cipher with 128-, 192-, or 256-bit key sizes.
- **Secure Hashing Algorithm (SHA):** This algorithm generates a cryptographically secure one-way hash; it was published by the National Institute of Standards and Technology as a US Federal Information Processing Standard.
- **Secure Sockets Layer (SSL):** SSL is an application layer protocol developed by Netscape for managing the security of message transmission on the Internet.

T

- **Tactics, Techniques, and Procedures (TTPs):** The term Tactics, Techniques, and Procedures (TTPs) refers to the patterns of activities and methods associated with specific threat actors or groups of threat actors.
- **Tactics:** "Tactics" are the guidelines that describe the way an attacker performs the attack from beginning to the end.
- **Techniques:** "Techniques" are the technical methods used by an attacker to achieve intermediate results during the attack.
- **Technical Threat Intelligence:** Technical threat intelligence provides information about resources an attacker uses to perform an attack; this includes command and control channels, tools, and other items.
- **Threat Modeling:** Threat modeling is a risk assessment approach for analyzing the security of an application by capturing, organizing, and analyzing all the information that affects the security of an application.
- **The Digital Millennium Copyright Act (DMCA):** It defines the legal prohibitions against the circumvention of technological protection measures employed by copyright owners to protect their works, and against the removal or alteration of copyright management information.
- **Traceroute:** Traceroute programs work on the concept of ICMP protocol and use the TTL field in the header of ICMP packets to discover the routers on the path to a target host.
- **TCP SYN Ping Scan:** TCP SYN ping is a host discovery technique for probing different ports to determine if the port is online and to check if it encounters any firewall rule sets.

- **Toggle-Case Attack**: Attackers try all possible combinations of upper and lower cases of a word present in the input dictionary.
- **Telephone/Cellphone Spyware**: Telephone/cellphone spyware is a software tool that gives you full access to monitor a victim's telephone or cellphone.
- **Trojan:** It is a program in which the malicious or harmful code is contained inside apparently harmless programming or data in such a way that the code can get control and cause damage, such as ruining the file allocation table on your hard disk.
- **Tailgating**: Tailgating implies accessing a building or secured area without the consent of the authorized person.
- **Tabnabbing**: In a tabnabbing attack, a malicious webpage tricks users by changing its content to resemble a familiar site, such as a bank login page, capturing their credentials when they switch back to the tab.
- **Throttling**: Throttling entails the setting up of routers for server access with a logic to throttle incoming traffic levels that are safe for the server.
- **TCP SACK Panic Attack:** TCP SACK panic attack is a remote attack vector in which attackers attempt to crash the target Linux machine by sending SACK packets with malformed MSS.
- **TCP/IP Hijacking:** TCP/IP hijacking involves using spoofed packets to seize control of a connection between a victim and target machine.
- **Two-Factor Authentication:** A two-factor authentication provides an extra layer of protection as it provides another vector of authentication in addition to a user's password.
- **Transit Gateway:** A transit gateway is a network routing solution that establishes and manages communication between an on-premises consumer network and VPCs via a centralized unit.
- **Triple Data Encryption Standard (3DES):** It performs DES three times with three different keys.
- **Twofish**: Twofish uses a block size of 128 bits and key sizes up to 256 bits. It is a Feistel cipher.
- **Threefish:** Threefish is a large tweakable symmetric-key block cipher in which the block and key sizes are equal, i.e., 256, 512, and 1024.
- **TEA:** Tiny Encryption Algorithm (TEA) is a Feistel cipher that uses 64 rounds.
- **TPM:** Trusted platform module (TPM) is a crypto-processor or chip that is present on the motherboard that can securely store the encryption keys, and it can perform many cryptographic operations.
- **Transport Layer Security (TLS):** TLS is a protocol to establish a secure connection between a client and a server and ensure the privacy and integrity of information during transmission.

U

- **Unsupervised Learning:** Unsupervised learning makes use of algorithms that input unlabeled training data to attempt to deduce all the categories without guidance.
- **UDP Ping Scan:** Attackers send UDP packets to target hosts, and a UDP response indicates that the host is active.
- **USB Spyware:** USB spyware is a program designed for spying on a computer, which copies spyware files from a USB device onto the hard disk without any request or notification.
- **UDP Flood Attack:** An attacker sends spoofed UDP packets at a very high packet rate to a remote host on random ports of a target server using a large source IP range.
- **UDP Hijacking:** A network-level session hijacking where the attacker sends forged server reply to a victim's UDP request before the intended server replies to it.

- **URL Encoding**: URL encoding is the process of converting URL into valid ASCII format so that data can be safely transported over HTTP.
- **UTF-8:** It is a variable-length encoding standard that uses each byte expressed in hexadecimal and preceded by the % prefix.
- **Union SQL Injection:** In a UNION SQL injection, an attacker combines a forged query with a query requested by the user using a UNION clause.
- **USB Encryption**: USB encryption is an additional feature for USB storage devices that offers onboard encryption services.

V

- **Vulnerability Research:** Vulnerability research is the process of analyzing protocols, services, and configurations to discover the vulnerabilities and design flaws that will expose an operating system and its applications to exploit, attack, or misuse.
- **Vulnerability Assessment:** Vulnerability assessment is an in-depth examination of the ability of a system or application, including current security procedures and controls, to withstand the exploitation.
- **Vulnerability Exploitation:** Vulnerability exploitation involves the execution of multiple complex, interrelated steps to gain access to a remote system.
- **Video Steganography:** Video steganography refers to hiding secret information in a carrier video file.
- **Virus:** A virus is a self-replicating program that produces its own copy by attaching itself to another program, computer boot sector or document.
- **Vishing:** Vishing (voice or VoIP phishing) is an impersonation technique (electronic fraud) in which the attacker tricks individuals to reveal personal and financial information using voice technology such as the telephone system, VoIP, etc.
- **VPN:** A VPN is a private network constructed using public networks, such as the Internet.
- **Vulnerability Scanning:** Vulnerability scanning involves analyzing protocols, services, and configurations to discover vulnerabilities and design flaws that may expose an operating system and its applications to exploitation, attack, or misuse.
- **Vulnerability Analysis**: Vulnerability analysis is the systematic process of identifying, evaluating, and prioritizing security weaknesses in systems, networks, applications, or protocols.
- **Vulnerability Assessment Reports:** A vulnerability assessment report is a comprehensive document that details the findings of a vulnerability assessment.
- **Virtual Private Cloud (VPC):** VPC is a secure and independent private cloud environment that resides within the public cloud.
- **Vulnerability**: A vulnerability refers to a weakness in the design or implementation of a system that can be exploited to compromise the security of the system.
- **Video Spyware**: Video spyware is software for video surveillance installed on a target computer without the user's knowledge.
- **Very Small Aperture Terminal (VSAT):** VSAT is a communication protocol that is used for data transfer using small dish antennas for both broadband and narrowband data.

W

- **White Hats:** White hats or penetration testers are individuals who use their hacking skills for defensive purposes.
- **WormGPT:** WormGPT is an AI-powered tool that assists cybersecurity professionals in automating the generation of worm-like scripts and payloads.
- **Website Footprinting:** Website footprinting refers to the monitoring and analysis of the target organization's website for information.
- **Whois:** Whois is a query and response protocol used for querying databases that store the registered users or assignees of an Internet resource, such as a domain name, an IP address block, or an autonomous system.
- **Wireless Network Vulnerability Scanning:** Wireless network scanning determines the vulnerabilities in an organization's wireless networks.
- **Wire Sniffing:** Packet sniffing is a form of wire sniffing or wiretapping in which hackers sniff credentials during transit by capturing Internet packets.
- **Windows Management Instrumentation (WMI):** WMI is a feature in Windows administration that provides a platform for accessing Windows system resources locally and remotely.
- **Windows Remote Management (WinRM):** WinRM is a Windows-based protocol designed to allow a user to run an executable file, modify system services, and the registry on a remote system.
- **Whitespace Steganography:** In white space steganography, the user hides the messages in ASCII text by adding white spaces to the ends of the lines.
- **Wiretapping:** Wiretapping is the process of the monitoring of telephone and Internet conversations by a third party.
- **Whaling:** A whaling attack is a type of phishing that targets high profile executives like CEO, CFO, politicians, and celebrities who have complete access to confidential and highly valuable information.
- **Web Server:** A web server is a computer system that stores, processes, and delivers web pages to clients via HTTP.
- **Website Defacement:** Website defacement refers to unauthorized changes made to the content of a single web page or an entire website, resulting in changes to the visual appearance of the web page or website.
- **Web Cache Poisoning Attack:** An attacker forces the web server's cache to flush its actual cache content and sends a specially crafted request to store in the cache.
- **Web Server Misconfiguration:** Server misconfiguration refers to configuration weaknesses in web infrastructure that can be exploited to launch various attacks on web servers such as directory traversal, server intrusion, and data theft.
- **Website Mirroring:** Website mirroring copies an entire website and its content onto a local drive.
- **Web Applications:** Web applications provide an interface between end users and web servers through a set of web pages that are generated at the server end or contain script code to be executed dynamically within the client web browser.
- **Web Service:** A web service is an application or software that is deployed over the Internet and uses standard messaging protocols such as SOAP, UDDI, WSDL, and REST to enable communication between applications developed for different platforms.
- **Web-based Timing Attack:** A web-based timing attack is a type of side-channel attack performed by attackers to retrieve sensitive information such as passwords from web applications by measuring the response time taken by the server.

- **Web Spidering/Crawling:** Web spiders/crawlers automatically discover the hidden content and functionality by parsing HTML forms and client-side JavaScript requests and responses.
- **WS-Address Spoofing:** In a WS-address spoofing attack, an attacker sends a SOAP message containing fake WS-address information to the server. The <ReplyTo> header consists of the address of the endpoint selected by the attacker rather than the address of the web service client.
- **Web API:** Web API is an application programming interface that provides online web services to client-side apps for retrieving and updating data from multiple online sources.
- **Webhooks:** Webhooks are user-defined HTTP callback or push APIs that are raised based on events triggered, such as receiving a comment on a post or pushing code to the registry.
- **Web Shell:** A web shell is a malicious piece of code or script that is developed using server-side languages such as PHP, ASP, PERL, RUBY, and Python and are then installed on a target server.
- **Web Application Fuzz Testing:** Web application fuzz testing (fuzzing) is a black-box testing method. It is a quality checking and assurance technique used to identify coding errors and security loopholes in web applications.
- **Whitelist Validation:** Whitelist validation is an effective technique in which only the list of entities that have been approved for secured access are accepted.
- **Wi-Fi:** Wireless network (Wi-Fi) refers to WLANs based on IEEE 802.11 standard, which allows the device to access the network from anywhere within an AP range.
- **Wired Equivalent Privacy (WEP):** WEP is a security protocol defined by the 802.11b standard; it was designed to provide a wireless LAN with a level of security and privacy comparable to that of a wired LAN.
- **Wi-Fi Protected Access (WPA):** WPA is a security protocol defined by 802.11i standards; it uses a Temporal Key Integrity Protocol (TKIP) that utilizes the RC4 stream cipher encryption with 128-bit keys and 64-bit MIC integrity check to provide stronger encryption and authentication.
- **WPA2:** WPA2 is an upgrade to WPA, and it includes mandatory support for counter mode with cipher block chaining message authentication code protocol (CCMP), an AES-based encryption mode with strong security.
- **WPA3:** WPA3 is an advanced implementation of WPA2 providing trailblazing protocols and uses the AES-GCMP 256 encryption algorithm.
- **Wireless Traffic Analysis:** Wireless traffic analysis enables attackers to identify vulnerabilities and susceptible victims in a target wireless network.
- **Wireless Intrusion Prevention Systems:** Wireless intrusion prevention systems (IPSs) protect networks against wireless threats and enable administrators to detect and prevent various network attacks.
- **Wrapping Attack:** A wrapping attack is performed during the translation of the SOAP message in the TLS layer where attackers duplicate the body of the message and sends it to the server as a legitimate user.
- **Web of Trust (WOT):** Web of trust (WoT) is a trust model of PGP, OpenPGP, and GnuPG systems.

X

- **Xmas Scan:** Xmas scan is a type of inverse TCP scanning technique with the FIN, URG, and PUSH flags set to send a TCP frame to a remote device.
- **XML External Entity Attack:** XML External Entity attack is a server-side request forgery (SSRF) attack that can occur when a misconfigured XML parser allows applications to parse XML input from an unreliable source.

Y

- **Yagi Antenna:** A Yagi antenna, also called Yagi–Uda antenna, is a unidirectional antenna commonly used in communications at a frequency band of 10 MHz to VHF and UHF.
- **YAK:** YAK is a public-key-based Authenticated Key Exchange (AKE) protocol.

Z

- **Zero-trust Principles:** Zero-trust principles constitute a set of standardized user pre-verification procedures that requires all users to be authenticated before providing access to any resource.
- **Zones and Conduits:** A network segregation technique used to isolate the networks and assets to impose and maintain strong access control mechanisms.
- **Zero Trust Network:** The Zero Trust model is a security implementation that assumes that every user trying to access the network is not a trusted entity by default and verifies every incoming connection before allowing access to the network.
- **Zero-Day Vulnerabilities:** Zero-day vulnerabilities are unknown vulnerabilities in software/hardware that are exposed but not yet patched.

Notes:

References

Module 01: Introduction to Ethical Hacking

1. [2006], Ethical Hacking, from http://neworder.box.sk/news/921.
2. [2006], Hacker methodology, from http://www.hackersecuritymeasures.com/.
3. [2006], The Cybercrime Act 2001 Australia, Germany, Singapore Chapter 50A: Computer misuse Act, from http://www.cybercrimelaw.net/laws/countries/australia.html, http://www.cybercrimelaw.net/laws/countries/germany.html, http://www.mosstingrett.no/info/legal.html#29.
4. [2006], Computer Misuse Act 1990 Chapter 18 Unauthorized access to computer material, from http://www.cybercrimelaw.net/laws/countries/uk.html.
5. Police and Justice Act 2006, http://www.opsi.gov.uk/acts/acts2006/ukpga_20060048_en_7#pt5-pb2.
6. Dan Goodin, [2012], Zero-day attacks are meaner, more rampant than we ever thought, from https://arstechnica.com/information-technology/2012/10/zero-day-attacks-are-meaner-and-more-plentiful-than-thought/.
7. Frank T. Bass, (1998), SECURITY POLICY: TARGET, CONTENT, & LINKS, from https://csrc.nist.gov/csrc/media/publications/conference-paper/1998/10/08/proceedings-of-the-21st-nissc-1998/documents/paperg4.pdf.
8. [2009], Anatomy of the Hack - Hands-on Security, from https://www.slideshare.net/NewBU/anatomy-of-the-hack-handson-security-information-assurance-club.
9. Nick Sutton and Jonathan Knight, (2014), Online in 60 Seconds – A Year Later, from https://blog.qmee.com/online-in-60-seconds-infographic-a-year-later/.
10. Vangie Beal, Insider attack, from https://www.webopedia.com/TERM/I/insider_attack.html.
11. Ian Sutherland, Is Ethical Hacking Actually Ethical or even Legal?, from https://ianhsutherland.com/ethical-hacking/.
12. Is ethical hacking legal?, from https://www.answers.com/Q/Is_ethical_hacking_legal?#slide=2.
13. PCI SSC Data Security Standards Overview, from https://www.pcisecuritystandards.org/pci_security/how.
14. [2010], Payment Card Industry (PCI) Data Security Standard, from https://www.pcisecuritystandards.org/document_library?category=pcidss&document=pci_dss.
15. ISO/IEC 27001:2013, from https://www.iso.org/standard/54534.html.
16. Health Information Privacy, from https://www.hhs.gov/hipaa/index.html.
17. [2002], PUBLIC LAW 107–204—JULY 30, from https://www.investor.gov/introduction-investing/investing-basics/role-sec/laws-govern-securities-industry#sox2002.
18. Executive Summary Digital Millennium Copyright Act Section 104 Report, from http://www.copyright.gov/reports/studies/dmca/dmca_executive.html.
19. [1998], The Digital Millennium Copyright Act of 1998 U.S. Copyright Office Summary, from https://www.copyright.gov/legislation/dmca.pdf.
20. [2002], Federal Information Security Management Act (FISMA) Implementation Project, from https://csrc.nist.gov/projects/risk-management.
21. Alan B. Sterneckert, Critical Incident Management, from https://www.taylorfrancis.com/books/9781420000047.
22. Joe Jenkins, [2000], Internet Security and Your Business - Knowing the Risks, from https://community.broadcom.com/home.
23. [2006], Critical Infrastructure Threats and Terrorism, from https://irp.fas.org/threat/terrorism/sup2.pdf.
24. Juan Andrés Guerrero-Saade, Costin Raiu, Kurt Baumgartner, (2017), Kaspersky Security Bulletin: Threat Predictions for 2018, from https://securelist.com/ksb-threat-predictions-for-2018/83169/.
25. Giovanni Vigna, (2017), The 2018 Cyberthreat Landscape—Predictions and Trends, from https://blogs.vmware.com/networkvirtualization/threat-intelligence/.
26. Scott Rosenberg, (2017), Firewalls Don't Stop Hackers. AI Might., from https://www.wired.com/story/firewalls-dont-stop-hackers-ai-might/.
27. Nick Ismail, (2017), The role of AI in cyber security, from http://www.information-age.com/role-ai-cyber-security-123465795/.
28. Vangie Beal, Machine Learning, from https://www.webopedia.com/TERM/M/machine-learning.html.
29. ELIEZER KANAL, (2017), Machine Learning in Cybersecurity, from https://insights.sei.cmu.edu/sei_blog/2017/06/machine-learning-in-cybersecurity.html.

30. Machine Learning in Cyber Security Domain – 1: Fundamentals, from https://www.nomshield.com/machine-learning-in-cyber-security-domain-1-fundamentals/.
31. Christy Pettey, (2016), Security Embraces Advanced Analytics and Machine Learning, from https://www.gartner.com/smarterwithgartner/security-embraces-advanced-analytics-and-machine-learning/.
32. Kasey Panetta, (2017), Gartner 7 Top Security Predictions for 2017, from https://www.gartner.com/smarterwithgartner/7-top-security-predictions-for-2017/.
33. Stamford, Conn, (2017), Gartner Says Four Vectors Are Transforming the Security Software Market, from https://www.gartner.com/newsroom/id/3731817.
34. (2017), Cybersecurity's Next Frontier: 80+ Companies Using Artificial Intelligence to Secure The Future In One Infographic, from https://www.cbinsights.com/research/cybersecurity-artificial-intelligence-startups-market-map/.
35. Martin Anderson, (2019), The State of AI in Cyber Security in 2019, From https://www.flexion.com/blog/ai-in-cyber-security.
36. (2019), Artificial Intelligence in Cybersecurity Market, from https://www.marketsandmarkets.com/Market-Reports/artificial-intelligence-security-market-220634996.html.
37. Remesh Ramachandran, (2019), How Artificial Intelligence Is Changing Cyber Security Landscape and Preventing Cyber Attacks, from https://www.entrepreneur.com/en-in/technology/how-artificial-intelligence-is-changing-cyber-security/339509.
38. TJ Horan, (2018), 5 Keys to Using AI and Machine Learning in Fraud Detection, from https://www.fico.com/blogs/5-keys-using-ai-and-machine-learning-fraud-detection.
39. Chigozie-Okwum C. C. and Ajah Ifeyinwa Angela, (2019), Botnet Identification Using Machine Learning Techniques: A Survey, from https://www.researchgate.net/publication/334284867_Botnet_Identification_Using_Machine_Learning_Techniques_A_Survey.
40. Raghav Bharadwaj, (2019), Artificial Intelligence in Cybersecurity – Current Use Cases and Capabilities, from https://emerj.com/ai-sector-overviews/artificial-intelligence-cybersecurity/.
41. Sergio Caltagirone, Andrew Pendergast, and Christopher Betz, (2013), Diamond Model of Intrusion Analysis, from https://www.threatintel.academy/diamond/.
42. Sergio Caltagirone, The Diamond Model of Intrusion Analysis, from https://www.threatintel.academy/wp-content/uploads/2020/07/diamond_summary.pdf.
43. Howard Poston, (2020), How to use the MITRE ATT&CK® framework and diamond model of intrusion analysis together, from https://www.infosecinstitute.com/resources/mitre-attck/how-to-use-the-mitre-attck-framework-and-diamond-model-of-intrusion-analysis-together/.
44. Alissa Knight, (2016), Digital Forensics According to the FORZA Model and Diamond Model for Intrusion Analysis, from https://cybersecurity.att.com/blogs/security-essentials/digital-forensics-according-to-the-forza-model-and-diamond-model-for-intrusion-analysis.
45. Andy Pendergast, Diamond Model of Intrusion Analysis, from https://threatconnect.com/blog/tag/diamond-model-of-intrusion-analysis/.
46. What is a Red Hat Hacker?, From https://www.techslang.com/definition/what-is-a-red-hat-hacker/.

Module 02: Footprinting and Reconnaissance

47. Search Operators, from https://www.googleguide.com/advanced_operators_reference.html.
48. Extract Website Information from https://archive.org.
49. Simson Garfinkel and David Cox, (2009), Finding and Archiving the Internet Footprint, from https://simson.net/clips/academic/2009.BL_InternetFootprint.pdf.
50. CHAPTER 2 [Footprinting], from https://e-cq.net/resources/wp/footprinting-encored.pdf.
51. Manic Velocity, Footprinting And The Basics Of Hacking, from http://web.textfiles.com/hacking/footprinting.txt.
52. Regional Internet Registry, From https://en.wikipedia.org/wiki/Regional_Internet_registry.
53. Arik R. Johnson, What is competitive intelligence?, from http://www.aurorawdc.com/whatisci.htm.
54. Manic Velocity, (2002), Footprinting: The Basics of Hacking: Hack in The Box, from https://news.hitb.org/node/5359.
55. P. Mockapetris, (1987), RFC 1034 [Domain Names - Concepts and Facilities], from https://datatracker.ietf.org/doc/html/rfc1034.
56. Reporting network abuse: Spamming and hacking, from https://help.apnic.net/s/article/About-network-abuse.
57. Kevin Beaver, How to Use Footprinting to Plan an Ethical Hack, from https://www.dummies.com/article/technology/cybersecurity/how-to-use-footprinting-to-plan-an-ethical-hack-168175/.
58. Ali Jahangiri, Google Hacking, from http://www.alijahangiri.org/publication/Google-Hacking-by-Ali-Jahangiri.pdf.
59. Richard A. Goodman, Competitive Intelligence, from https://www.anderson.ucla.edu/rosenfeld-library/learn/competitive-intelligence#A-1.
60. Sam Bowne, Chapter 1: Footprinting, from http://samsclass.info/124/ppt/ch01.ppt.

61. (1998). Chapter 6: Competitive Intelligence, from http://gaius.cbpp.uaa.alaska.edu/afef/IMIchapter-6.htm.
62. Eddie Sutton, Footprinting: What is it and How Do You Erase Them, from http://www.infosecwriters.com/text_resources/pdf/Footprinting.pdf.
63. Locating Exploits and Finding Targets, from http://media.techtarget.com/searchSoftwareQuality/downloads/Google_Hacking_Penetration_Testers.pdf.
64. Johnny Long, (2008), Google Hacking: Ten security searches that work, from https://www.techtarget.com/searchitchannel/tips.
65. Google hacking, from https://en.wikipedia.org/wiki/Google_hacking.
66. Google Tips, from http://www.pafumi.net/Google_Tips.html.
67. (2018), Google Hacking Database (GHDB), from https://www.exploit-db.com/google-hacking-database.
68. Nihad A. Hassan, Rami Hijazi, (2018), Open Source Intelligence Methods and Tools_ A Practical Guide to Online Intelligence, Newyork, USA: Apress.
69. (2020), Video search engine, from https://en.wikipedia.org/wiki/Video_search_engine.
70. (2019), Metasearch engine, from https://en.wikipedia.org/wiki/Metasearch_engine#Advantages.
71. Subarna Kumar Das, (2006), Role of Meta Search Engines in Web-Based Information System: Fundamentals and Challenges, from http://ir.inflibnet.ac.in:8080/ir/bitstream/1944/1327/1/445-454.pdf.
72. Rafay Baloch, (2015), Ethical Hacking and Penetration Testing Guide, London, Newyork: CRC Press.
73. Timothy Shim, (2020), How to Access the Dark Web: Browsing Dark Web, TOR Browser, and Onion Websites, from https://www.webhostingsecretrevealed.net/blog/web-tools/tourist-guide-to-dark-web-accessing-the-dark-web-tor-browser-and-onion-websites/.
74. Paul Bischoff, (2018), Step by step guide to safely accessing the dark net and deep web, from https://www.comparitech.com/blog/vpn-privacy/how-to-access-the-deep-web-and-darknet/.
75. (2019), Usenet newsgroup, from https://en.wikipedia.org/wiki/Usenet_newsgroup.
76. (2019), Sherlock – A Tool to Find Usernames Across Social Networks, from https://latesthackingnews.com/2019/01/30/sherlock-a-tool-to-find-usernames-across-social-networks/.
77. Raj Chandel, (2017), Beginner Guide to Footprinting, from https://www.hackingarticles.in/beginner-guide-footprinting/.
78. Raj Chandel, (2017), POST CATEGORY: Footprinting, from https://www.hackingarticles.in/category/footprinting/.
79. Patrick Engebretson, and David Kennedy, (2013), The Basics of Hacking and Penetration Testing, Second Edition, USA: Elsevier.
80. Dafydd Stuttard, and Marcus Pinto, (2011), The Web Application Hacker's Handbook, Second Edition, Indianapolis, Indiana: John Wiley & Sons, Inc.
81. Kadam Parikh, (2017), Email Footprinting- Trace and Email and Collect Information from it..!, from https://libraryofhacks.blogspot.com/2017/06/email-footprinting-trace-email-and.html.
82. (2018), What You Need to Know About Code Repository Threats, from https://cyberint.com/blog/threat-intelligence/what-you-need-to-know-about-code-repository-threats/.
83. Itamar Mizrahi, Josh Liburdi, and Toby Kohlenberg, (2021), Data from Information Repositories: Code Repositories, from https://attack.mitre.org/techniques/T1213/003/.
84. (2019), Securing Your Public Source Code Repositories, from https://danielmiessler.com/blog/securing-public-source-code-repositories/.
85. (2021), Recon-ng information gathering tool in Kali Linux, from https://www.geeksforgeeks.org/recon-ng-installation-on-kali-linux/#:~:text=Recon%2Dng%20is%20a%20Web,we%20can%20gather%20all%20information.
86. (2013), Social Network Analysis, from https://www.sciencedirect.com/topics/social-sciences/social-network-analysis#:~:text=The%20resulting%20graph%20can%20reveal,role%20in%20connecting%20groups%20together.
87. Kirichenko Lyudmyla, Radivilova Tamara, and Carlsson Anders, Detecting cyber threats through social network analysis: short survey, from https://arxiv.org/ftp/arxiv/papers/1805/1805.06680.pdf.
88. Chris Sheedy, (2019), Social network analysis: what it is and why it matters, from https://intheblack.cpaaustralia.com.au/careers-and-workplace/what-is-social-network-analysis.
89. Premsankar Chakkingal, (2013), Introduction to Social Network Analysis, from https://www.slideshare.net/premsankarchakkingal/introduction-to-social-network-analysis.

Module 03: Scanning Networks

90. Fyodor, (1998), Remote OS detection via TCP/IP Stack FingerPrinting, from https://nmap.org/nmap-fingerprinting-article.txt.
91. Remote OS Detection, from https://nmap.org/book/osdetect.html.
92. Katherine Knickerbocker, Spoofing, from http://all.net/CID/Attack/papers/Spoofing.html.

93. Explanation of the Three-Way Handshake via TCP/IP, from https://learn.microsoft.com/en-us/troubleshoot/windows-server/networking/three-way-handshake-via-tcpip.
94. The Art of Port Scanning - by Fyodor, from https://nmap.org/nmap_doc.html.
95. Hacking Exposed, from https://www.scribd.com/document/62708034/Hacking-Exposed-Book.
96. Chris McNab, Network Security Assessment, from https://www.trustmatta.com/downloads/pdf/Matta_IP_Network_Scanning.pdf.
97. Steven J. Templeton, and Karl E. Levitt, Detecting Spoofed Packets, from http://seclab.cs.ucdavis.edu/papers/DetectingSpoofed-DISCEX.pdf.
98. Thierry Lagarde, (2014), AutoScan Network, from https://autoscan-network.en.lo4d.com/windows.
99. Avi Kak, (2019), Port Scanning, Vulnerability Scanning, Packet Sniffing, and Intrusion Detection, and Penetration Testing, from https://engineering.purdue.edu/kak/compsec/NewLectures/Lecture23.pdf.
100. Renaud Deraison, Ron Gula, and Todd Hayton, (2009), Passive Vulnerability Scanning Introduction, from https://docs.huihoo.com/nessus/passive_scanning_tenable.pdf.
101. Lance Cottrell, Anonymizer Limitations: Logs, from https://www.livinginternet.com/i/is_anon.htm.
102. D. Dieterle, (2013), SSDP Scanning for UPnP Vulnerabilities, from https://cyberarms.wordpress.com/2013/08/23/ssdp-scanning-for-upnp-vulnerabilities/.
103. Sharan R, Hacking Techniques - Scanning Networks and Countermeasures, from http://hack-o-crack.blogspot.com/2010/12/hacking-techniques-scanning-networks.html.
104. Chris McNab, Network Security Assessment, from https://www.trustmatta.com/downloads/pdf/Matta_IP_Network_Scanning.pdf.
105. Wade Stich, Banner Grabbing with Telnet, from https://siekta.wordpress.com/wp-content/uploads/2010/10/bannergrabbing.pdf.
106. Network scanning, from https://www.techtarget.com/searchnetworking/definition/network-scanning.
107. Ryan Spangler, (2003), Analysis of Remote Active Operating System Fingerprinting Tools, from http://www.packetwatch.net/documents/papers/osdetection.pdf.
108. Ryan Spangler, (2003), Analysis of Remote Active Operating System Fingerprinting Tools, from https://www.beyondsecurity.com/.
109. (2000), How Anonymizers Work, from https://www.livinginternet.com/i/is_anon_work.htm.
110. (2008), Scanning - HSC Guides - Ethical Hacker, from http://sqlinjections.blogspot.com/2009/04/scanning-hsc-guides-ethical-hacker.html.
111. Lance Spitzner, (2000), Passive Fingerprinting, from https://community.broadcom.com/home.
112. Prabhaker Mateti, (2001), Port Scanning, from https://web1.cs.wright.edu/~pmateti/Courses/499/Probing/index.html.
113. Wenliang (Kevin) Du, TCP Protocols, from http://www.cis.syr.edu/~wedu/Teaching/cis758/LectureNotes/TCP.doc.
114. (2008), Proxy Servers and Anonymizers, from https://www.gohacking.com/what-is-proxy-server-and-how-it-works/.
115. Port Scanning without the SYN flag, from http://bsd.openmet.ru/base/soc/p49-15.txt.html.
116. Christian Starkjohann, What is sslproxy?, from https://kuix.de/sslproxy/README.txt.
117. Bálint KOZMAN, (2002), Anonymizers, from http://www.fm.uni-linz.ac.at/staff/sonntag/LTAEC_Budapest/Anonymizers/index.html.
118. Firewall/IDS Evasion and Spoofing, from https://nmap.org/book/man-bypass-firewalls-ids.html.
119. (2016), How can the Nmap tool be used to evade a firewall/IDS?, from https://security.stackexchange.com/questions/121900/how-can-the-nmap-tool-be-used-to-evade-a-firewall-ids.
120. (2012), Nmap – Techniques for Avoiding Firewalls, from https://pentestlab.blog/2012/04/02/nmap-techniques-for-avoiding-firewalls/.
121. Rajesh Deodhar, (2011), Advanced Nmap: Scanning Firewalls, from http://opensourceforu.com/2011/02/advanced-nmap-scanning-firewalls/.
122. Taylor Gibb, (2012), Hacker Geek: OS Fingerprinting With TTL and TCP Window Sizes, from https://www.howtogeek.com/104337/hacker-geek-os-fingerprinting-with-ttl-and-tcp-window-sizes/.
123. Evrik Hjelmvik, (2011), Passive OS Fingerprinting, from http://www.netresec.com/?page=Blog&month=2011-11&post=Passive-OS-Fingerprinting.
124. Chris Sanders, (2011), Operating System Fingerprinting with Packets [Part 1], from http://techgenix.com/operating-system-fingerprinting-packets-part1/.
125. (2012), Penetration Testing Lab, from https://pentestlab.blog/2012/08/17/nmap-cheat-sheet/.
126. Host Discovery Techniques, from https://nmap.org/book/host-discovery-techniques.html.
127. Gordon "Fyodor" Lyon, (2008), Nmap_network_scanning, United States: Insecure.Com LLC.

128. Paulino Calderon, (2012), Discovering hosts with UDP ping scans, from https://subscription.packtpub.com/book/networking_and_servers/9781849517485/2/ch02lvl1sec24/discovering-hosts-with-udp-ping-scans.
129. Sairam Jetty, (2018), Network Scanning Cookbook Practical network security using Nmap and Nessus 7, Birmingham, UK: Packt Publishing.
130. (2014), Ping Sweeps and Port Scans, from https://computersecurity123.wordpress.com/2014/07/16/ping-sweeps-and-port-scans/.
131. Kevin Beaver, Prevent Network Hacking with Port Scanners, from https://www.dummies.com/programming/networking/prevent-network-hacking-with-port-scanners/.
132. (2018), When is 'Timestamp' and 'Timestamp Reply' are used in ICMP protocol?, from https://networkengineering.stackexchange.com/questions/50611/when-is-timestamp-and-timestamp-reply-are-used-in-icmp-protocol.
133. (2020), Internet Control Message Protocol, from https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol.
134. Savia Lobo, (2018), Discovering network hosts with 'TCP SYN' and 'TCP ACK' ping scans in Nmap, from https://hub.packtpub.com/discovering-network-hosts-with-tcp-syn-and-tcp-ack-ping-scans-in-nmaptutorial/.
135. Paulino Calderon, (2012), Discovering hosts with UDP ping scans, from https://subscription.packtpub.com/book/networking_and_servers/9781849517485/2/ch02lvl1sec24/discovering-hosts-with-udp-ping-scans.
136. Port Scanning Techniques, from https://nmap.org/book/man-port-scanning-techniques.html.
137. (2014), Nmap Cheat Sheet, from https://highon.coffee/blog/nmap-cheat-sheet/.
138. TCP FIN, NULL, and Xmas Scans (-sF, -sN, -sX), from https://nmap.org/book/scan-methods-null-fin-xmas-scan.html.
139. TCP Maimon Scan (-sM), from https://nmap.org/book/scan-methods-maimon-scan.html.
140. Philippe Langlois, (2013), SCTPscan Finding entry points to S57 Networks & Telecommunication Backbones, from https://www.slideshare.net/slideshow/black-hat-europe-philippe-langlois-sctpscan/18617727.
141. Nmap Network Scanning IPv6 Scanning (-6), from https://nmap.org/book/port-scanning-ipv6.html.
142. Jason Wood, (2016), Nmap Scanning IPv6 Addresses, from https://www.youtube.com/watch?v=WWHyg4Qjz6c.
143. Nmap Network Scanning Service and Version Detection, from https://nmap.org/book/man-version-detection.html.
144. Nmap Network Scanning Scan Time Reduction Techniques, from https://nmap.org/book/reduce-scantime.html.
145. Taylor Gibb, (2012), Hacker Geek: OS Fingerprinting with TTL and TCP Window Sizes, from https://www.howtogeek.com/104337/hacker-geek-os-fingerprinting-with-ttl-and-tcp-window-sizes/.
146. Erik Hjelmvik, (2011), Passive OS Fingerprinting, from https://www.netresec.com/?page=Blog&month=2011-11&post=Passive-OS-Fingerprinting.
147. Chris Sanders, (2011), Operating System Fingerprinting with Packets (Part 1), from http://techgenix.com/operating-system-fingerprinting-packets-part1/.
148. TCP window size – Wireshark and Windows, from https://microsoftserver2012.wordpress.com/2016/10/06/tcp-window-size-wireshark-and-windows/.
149. Subin Siby, (2019), Default TTL (Time To Live) Values of Different OS, from https://subinsb.com/default-device-ttl-values/.
150. Chris Greer, (2018), How TCP Works - Window Scaling and Calculated Window Size, from https://www.youtube.com/watch?v=2PiVHsthrNU.
151. OS Detection, from https://nmap.org/book/man-os-detection.html.
152. Service and Version Detection, from https://nmap.org/book/man-version-detection.html.
153. Ron Bowes, (2008), My Scripting Experience with Nmap, from https://www.skullsecurity.org/2008/what-time-is-it.
154. Ron Bowes, File smb-os-discovery, from https://nmap.org/nsedoc/scripts/smb-os-discovery.html.
155. IPv6 fingerprinting, from https://nmap.org/book/osdetect-ipv6-methods.html.
156. Jonathan Hassell, (2003), What is IP spoofing? And 5 ways to prevent it, from https://www.csoonline.com/article/2115848/data-protection-ip-spoofing.html.
157. Prateek Parashar, (2021), Evading Firewall/IDS during network reconnaissance using nmap, from https://infosecwriteups.com/evading-firewall-ids-during-network-reconnaissance-using-nmap-7dc393138178.
158. (2012), Nmap – Techniques for Avoiding Firewalls, from https://pentestlab.blog/2012/04/02/nmap-techniques-for-avoiding-firewalls/.
159. Irfan Shakeel, (2019), Nmap evade firewall and scripting, from https://www.infosecinstitute.com/resources/hacking/nmap-evade-firewall-scripting/.
160. Paulino Calderon, MAC address spoofing, from https://www.oreilly.com/library/view/nmap-network-exploration/9781786467454/bf4ba4ba-23ae-4267-ad5c-8a5bbfb2374c.xhtml.

161. Network Scanning Countermeasures, from http://etutorials.org/Networking/network+security+assessment/Chapter+4.+IP+Network+Scanning/4.7+Network+Scanning+Countermeasures/.
162. Craig Bodrick, (2019), Defending Against Port Scan Attacks, from https://www.turn-keytechnologies.com/blog/article/defending-against-port-scan-attacks/.
163. Port Scanning and Service Discovery, from https://flylib.com/books/en/2.356.3.17/1/.
164. Chandrakant Patil, (2020), How To Defend Against Port Scan Attacks, from https://hackersonlineclub.com/how-to-defend-against-port-scan-attacks/.
165. SIMON BATT, (2021), What Is Port Scanning and How Does It Work?, from https://www.makeuseof.com/what-is-port-scanning/.
166. Aniket Pandey, (2021), Port Scanning based Attacks, from https://bhaifi.com/blog/what-is-port-scanning-and-how-to-use-it-to-attack-2021/.
167. (2017), IIS Server Hardening – Banner Grabbing Prevention Techniques, from https://www.yeahhub.com/iis-server-hardening-banner-grabbing-prevention-techniques.
168. IP Spoofing: How it Works and How to Prevent It, from https://www.kaspersky.com/resource-center/threats/ip-spoofing.
169. Pannashree Saha, (2021), How to protect against IP spoofing?, from https://www.encryptionconsulting.com/how-to-protect-against-ip-spoofing/.
170. Anastasios Arampatzis, (2020), What Is IP Spoofing and How to Prevent It?, from https://www.venafi.com/blog/what-ip-spoofing-and-how-prevent-it.
171. David Balaban, (2020), 11 Types of Spoofing Attacks, from https://www.securitymagazine.com/articles/91980-types-of-spoofing-attacks-every-security-professional-should-know-about.
172. (2021), What it is IP Spoofing, How to Protect Against It, from https://www.keyfactor.com/blog/what-it-is-ip-spoofing-how-to-protect-against-it/.
173. Oswaldo Fonseca, Italo Cunha, Elverton Fazzion, Wagner Meira, Brivaldo Alves da Silva, Ronaldo A. Ferreira, and Ethan Katz-Bassett, (2021), Identifying Networks Vulnerable to IP Spoofing, from https://ieeexplore.ieee.org/abstract/document/9360876.
174. Prakash Veeraraghavan, (2020), NAT++: An Efficient Micro-NAT Architecture for Solving IP-Spoofing Attacks in a Corporate Network, from https://www.mdpi.com/2079-9292/9/9/1510.

Module 04: Enumeration

175. The Dark Side of NTFS (Microsoft's Scarlet Letter), from http://www.infosecwriters.com/texts.php?op=display&id=53.
176. RPCCLIENT, from https://seratahosting.com/manpages/rpcclient.html.
177. smtp-user-enum User Documentation, from http://pentestmonkey.net/tools/user-enumeration/smtp-user-enum.
178. What is SNMP?, from http://www.wtcs.org/snmp4tpc/snmp.htm.
179. Jan van Oorschot, Jeroen Wortelboer and Dirk Wisse, (2001), SNMP - The Mission Statement, from https://community.broadcom.com/home.
180. Kris Evers, (2006), AT&T hack exposes 19,000 identities, from https://www.cnet.com/news/at-t-hack-exposes-19000-identities/.
181. (2019), Linux finger command, from https://www.computerhope.com/unix/ufinger.htm.
182. (2008), Enumeration, from http://www.vulnerabilityassessment.co.uk/enum.htm.
183. (2009), Net view, from https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-xp/bb490719(v=technet.10)?redirectedfrom=MSDN.
184. Djrand, (2003), Low-Level Enumeration With TCP/IP, from https://packetstormsecurity.com/papers/bypass/Low-Level_Enumeration_With_TCP.txt.
185. LDAP Enumeration, from http://www.vulnerabilityassessment.co.uk/enum_ldap.htm.
186. SMTP Enumeration and more, from http://www.vulnerabilityassessment.co.uk/enum_smtp.htm.
187. (2019), Linux rpcinfo command, from https://www.computerhope.com/unix/urpcinfo.htm.
188. Arun Thomas, (2012), IPsec VPN Penetration Testing with BackTrack Tools, from http://opensourceforu.com/2012/01/ipsec-vpn-penetration-testing-backtrack-tools/.
189. Nadeem Unuth, (2019), What is SIP and How does it work?, from https://www.lifewire.com/what-is-sip-3426669.
190. Irfan Shakeel, (2016), VoIP Network Recon: Footprinting, Scanning, and Enumeration, from https://www.infosecinstitute.com/resources/penetration-testing/voip-network-recon-footprinting-scanning-and-enumeration/#gref.
191. Enumerating and Breaking VoIP, from http://garage4hackers.com/attachment.php?attachmentid=120&d=1321865311.
192. Endler & Collier, (2006), Enumerating a VoIP Network, from http://www.hackingvoip.com/presentations/sample_chapter3_hacking_voip.pdf.

193. (2003). What is RPC?, from https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc787851(v=ws.10)?redirectedfrom=MSDN.
194. Enumerating Unix RPC Services, from http://etutorials.org/Networking/network+security+assessment/Chapter+12.+Assessing+Unix+RPC+Services/12.1+Enumerating+Unix+RPC+Services/.
195. Adam Bertram, Using Windows 10 Administrative Shares, from https://www.businessnewsdaily.com/11017-windows-10-administrative-shares.html.
196. Sean Wilkins, (2012). TCP/IP Ports and Protocols, from http://www.pearsonitcertification.com/articles/article.aspx?p=1868080.
197. Shahmeer Amir, (2017), Penetration Testing of an FTP Server, from https://shahmeeramir.com/penetration-testing-of-an-ftp-server-19afe538be4b.
198. Raj Chandel, (2017), Penetration Testing on Telnet (Port 23), from https://www.hackingarticles.in/penetration-testing-telnet-port-23/.
199. Network File System (NFS), from https://web.mit.edu/rhel-doc/5/RHEL-5-manual/Deployment_Guide-en-US/ch-nfs.html.
200. (2019). NetBIOS Suffix Definitions, from https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-brws/0c773bdd-78e2-4d8b-8b3d-b7506849847b.
201. (2019). Enumerate NetBIOS Shares with NBTScan & Nmap Scripting Engine, from https://null-byte.wonderhowto.com/how-to/enumerate-netbios-shares-with-nbtscan-nmap-scripting-engine-0193957/.
202. NET.exe VIEW, from https://ss64.com/nt/net-view.html.
203. (2018). The SNMP Management Information Base (MIB), from https://learn.microsoft.com/en-us/windows/win32/snmp/the-snmp-management-information-base-mib-.
204. ntpdate(8) - Linux man page, from https://linux.die.net/man/8/ntpdate.
205. Satyam Singh, (2018), Exploiting NFS Share, from https://resources.infosecinstitute.com/exploiting-nfs-share/#gref.
206. Joe Norton, (2017), Exploiting Metasploitable Without Metasploit — NFS Enumeration and Exploiting Misconfiguration, from https://medium.com/@joe_norton/exploiting-metasploitable-without-metasploit-nfs-enumeration-and-exploiting-misconfiguration-86504ccd15b9.
207. (2017). NFS, from https://pentestacademy.wordpress.com/2017/09/20/nfs/.
208. Margaret Rouse, (2019), Network File System (NFS), from https://www.techtarget.com/searchenterpriselesdesktop/definition/Network-File-System.
209. (2019). Network File System, from https://en.wikipedia.org/wiki/Network_File_System.
210. (2019). Common SMTP port numbers, from https://docs.mailpoet.com/article/59-default-ports-numbers-smtp-pop-imap.
211. (2012). SMTP User Enumeration, from https://pentestlab.blog/2012/11/20/smtp-user-enumeration/.
212. Duarte Silva, File smtp-enum-users, from https://nmap.org/nsedoc/scripts/smtp-enum-users.html.
213. Heyder Andrade and Nebulus, (2018). SMTP User Enumeration Utility, from https://www.rapid7.com/db/modules/auxiliary/scanner/smtp/smtp_enum.
214. Suzanne Goldlust, (2018), What is DNS Cache snooping?, from https://kb.isc.org/docs/aa-00509.
215. Dejan Lukan, (2012), DNS Cache Snooping, from https://www.infosecinstitute.com/resources/hacking/dns-cache-snooping/#gref.
216. Johannes Weber, (2016), How to walk DNSSEC Zones: dnsrecon, from https://weberblog.net/how-to-walk-dnssec-zones-dnsrecon/.
217. What Are Domain Name System Security Extensions?, from https://whatismyipaddress.com/dnssec.
218. (2014). DNSSEC:NSEC vs. NSEC3, from https://www.internetsociety.org/resources/deploy360/2014/dnssecnsec-vs-nsec3/.
219. DNSSEC – What is It and Why is It Important?, from https://www.icann.org/resources/pages/dnssec-what-is-it-why-important-2019-03-05-en.
220. How DNSSEC Works, from https://www.cloudflare.com/dns/dnssec/how-dnssec-works/.
221. NSEC3-Records (Next Secure v. 3), from https://simpledns.com/help/nsec3-records.
222. (2020). Telnet, from https://en.wikipedia.org/wiki/Telnet.
223. Dinesh Thakur, Telnet - What is Telnet?, from http://ecomputernotes.com/computernetworkingnotes/services-and-applications/what-is-telnet.
224. Kevin Beaver, (2014), How to detect and defend against a TCP port 445 exploit and attacks, from https://www.techtarget.com/searchsecurity/answer/Detecting-and-defending-against-TCP-port-445-attacks.
225. Raj Chandel, (2019), SMB Penetration Testing (Port 445), from https://www.hackingarticles.in/smb-penetration-testing-port-445/.
226. Shahmeer Amir, (2017), Penetration Testing of an FTP Server, from https://shahmeeramir.com/penetration-testing-of-an-ftp-server-19afe538be4b.

227. Raj Chandel, (2017), FTP Penetration Testing on Wiedows (Port 21), from https://www.hackingarticles.in/ftp-penetration-testing-windows/.
228. Margaret Rouse, (2019), IPv6 (Internet Protocol Version 6), from https://www.techtarget.com/searchnetworking/definition/IPv6-Internet-Protocol-Version-6.
229. (2020), IPv6, from https://en.wikipedia.org/wiki/IPv6.
230. Srinivas, (2016), SNMP Pentesting, from https://www.infosecinstitute.com/resources/penetration-testing/snmp-pentesting/#gref.
231. (2011), Enumeration Countermeasures, from http://luizfirmino.blogspot.com/2011/09/enumeration-countermeasures.html.
232. Kevin Beaver, (2014), How to detect and defend against a TCP port 445 exploit and attacks, from https://www.techtarget.com/searchsecurity/answer/Detecting-and-defending-against-TCP-port-445-attacks.
233. Kevin Beaver, Countermeasures against NFC attacks, from https://books.google.co.in/books?id=xuF3Rr2uD3YC&pg=PR13&lpg=PR13&dq=NFS+enumeration+countermeasures&source=bl&ots=58RSPmXEf7&sig=ACfU3U2rFLwulSj9ztVenCLItdQMaCLd7A&hl=en&sa=X&ved=2ahUKEwiwsctW9ILjAhUQ8HMBHcAUAvYQ6AEwBHoECAkQAQ#v=onepage&q=NFS%20attacks&f=false.
234. Stuart McClure, Joel Scambray, and George Kurtz, (2009), Hacking Exposed 6, Tenth Edition, USA: McGraw-Hill Companies, Inc.
235. Tim Keary, (2022), snmpwalk Examples & Commands for Windows and Linux, from https://www.comparitech.com/net-admin/snmpwalk-examples-windows-linux.
236. Harley, (2021), Enumerating SNMP for Pentesting (UDP Ports 161, 162), From https://infinitelogins.com/2021/02/21/enumerating-snmp-for-pentesting-udp-ports-161-162.
237. Daniel Miller, Nmap snmp-info NSE Script, from https://www.infosecmatter.com/nmap-nse-library/?nse=snmp-info.
238. (2019), Enumerating SNMP Servers with NMAP, from https://medium.com/@minimalist.ascent/enumerating-snmp-servers-with-nmap-89aaf33bcc28.
239. (2019), SNMP Enumeration Basics - Mischief HTB PenTest/Hacking Basics for UDP 161 (SNMPWALK), from https://www.youtube.com/watch?v=qt7XlO2Ly54.
240. (2021), SMTP (Simple Mail Transfer Protocol), from https://0xffsec.com/handbook/services/smtp/.
241. Kanishka, (2017), SMTP enumeration with Kali Linux, from https://www.hackercoolmagazine.com/smtp-enumeration-with-kali-linux-nmap-and-smtp-user-enum/.
242. Enumerating users in an SMTP server, from https://www.oreilly.com/library/view/nmap-6-network/9781849517485/ch06s05.html.
243. Scanner SMTP Auxiliary Modules, from https://www.offsec.com/metasploit-unleashed/scanner-smtp-auxiliary-modules/.
244. SMTP enumeration, from https://subscription.packtpub.com/book/networking-and-servers/9781788623179/2/ch02lvl1sec34/smtp-enumeration.
245. (2012), SMTP User Enumeration, from https://pentestlab.blog/tag/smtp-user-enum/.
246. Heyder Andrade, (2019), SMTP User Enumeration Utility - Metasploit, from https://www.infosecmatter.com/metasploit-module-library/?mm=auxiliary/scanner/smtp/smtp_enum.
247. (2018), SMTP User Enumeration Utility, from https://www.rapid7.com/db/modules/auxiliary/scanner/smtp/smtp_enum/.
248. Raj Chandel, (2017), 4 Ways to DNS Enumeration, from https://www.hackingarticles.in/4-ways-dns-enumeration/.
249. (2022), Pentesting DNS, from https://book.hacktricks.xyz/pentesting/pentesting-dns.
250. John R. Bond, Nmap dns-nsec-enum NSE Script, from https://www.infosecmatter.com/nmap-nse-library/?nse=dns-nsec-enum.
251. Esteban Borges, (2021), DNS Enumeration: Top DNS Recon Tools and Techniques, from https://www.recordedfuture.com/threat-intelligence-101/tools-and-techniques/dns-enumeration.
252. (2020), DNS enumeration - Nmap, from https://www.youtube.com/watch?v=vktnESGB0Cs.
253. 389, 636, 3268, 3269 - Pentesting LDAP, from https://book.hacktricks.xyz/pentesting/pentesting-ldap.
254. (2020), Exploiting LDAP Server NULL Bind, from https://laptrinhx.com/exploiting-ldap-server-null-bind-2433586944/.
255. (2021), Quick Tutorial : SNMP Enumeration, from https://allabouttesting.org/quick-tutorial-snmp-enumeration.
256. Raghu Chakravarthula, (2021), What is enumeration?, from https://www.infosecinstituto.com/resources/penetration-testing/what-is-enumeration/.
257. Indar Schusterbauer, (2020), What is SNMP enumeration?, from https://findanyanswer.com/what-is-snmp-enumeration.
258. Adnan, SNMP Enumeration Countermeasures, from https://www.hackguide4u.com/2010/02/snmp-enumeration-countermeasures.html.
259. Tim Keary, (2022), What are some Common SNMP vulnerabilities and how do you protect your network?, from https://www.comparitech.com/net-admin/common-snmp-vulnerabilities.

260. Ioson Zhou. (2010). How do you prevent ldap enumeration?, from https://learn.microsoft.com/en-us/archive/msdn-technet-forums/cf79c2f2-4467-45cb-8041-abc6abb56b33.

261. Adeem Marwani, (2021), Detecting LDAP Reconnaissance, from https://blog.blacklanternsecurity.com/p/detecting-ldap-reconnaissance?s=r.

262. Madhukar Raina, (2021), Detecting LDAP enumeration and Bloodhound's Sharphound collector using AD Decoys, from https://medium.com/securonix-tech-blog/detecting-ldap-enumeration-and-bloodhound-s-sharphound-collector-using-active-directory-decoys-dfcB40f2f644.

263. (2022). General guidelines for securing Network File System, from https://www.ibm.com/docs/en/aix/7.2?topic=security-general-guidelines-securing-network-file-system.

264. Maciej Zalwert, (2021), Ethical Hacking: Enumeration techniques with examples and tools, from https://maciejzalwert.medium.com/ethical-hacking-part-3-0-20-enumeration-techniques-with-examples-and-tools-6c274408754.

265. The Complete Guide to Cybersecurity for Small to Medium Sized Businesses, from https://powerconsulting.com/smb-cybersecurity/.

266. Paul Rubens, (2017), How to Prevent DNS Attacks, from https://www.esecurityplanet.com/networks/how-to-prevent-dns-attacks/.

267. Esteban Borges, (2018). 8 tips to prevent DNS attacks, from https://securitytrails.com/blog/8-tips-to-prevent-dns-attacks.

268. What is DNS Hijacking? Basic Methods Of Protection, from https://www.wallarm.com/what/what-is-dns-hijacking-basic-methods-of-protection.

269. Josh Lake, (2020), What is a DNS Attack? Types of DNS Attacks and How to Prevent Them, from https://www.comparitech.com/blog/information-security/what-is-dns-attack/.

270. Rishabh Sharma, (2020), Active Subdomain Enumeration, from https://notworkintelligence.ai/active-subdomain-enumeration-part-2/.

271. Enumeration Definition Scanning Identifies Live Hosts and Running, from https://slidetodoc.com/enumeration-definition-scanning-identifies-live-hosts-and-running/.

272. Van Glass, (2024), What port does SFTP use?, from https://www.jscape.com/blog/what-port-does-sftp-use.

273. Moulik, (2023), Information Gathering AMASS, from https://techyrick.com/amass-full-tutorial/.

274. Nick Gkogkos, How to Use OWASP Amass: An Extensive Tutorial, from https://www.dionach.com/how-to-use-owasp-amass-an-extensive-tutorial/.

275. (2023). Reconnaissance 101: Subdomain Enumeration, from https://blog.projectdiscovery.io/recon-series-2/.

Module 05: Vulnerability Analysis

276. Microsoft Vulnerability Research (MSVR), from https://www.microsoft.com/en-us/msrc/msvr.

277. Glossary of Vulnerability Testing Terminology, from https://www.ee.oulu.fi/research/ouspg/Glossary/.

278. Thomas R. Peltier, Justin Peltier, and John A. Blackley, (2017), Technical (Bottom-Up) Methodology, from https://www.taylorfrancis.com/books/9780429210747.

279. Renaud Deraison and Ron Gula, (2009), Blended Security Assessments, from https://www.tenable.com/sites/drupal.dmz.tenablesecurity.com/files/uploads/documents/whitepapers/Blended%20Security%20Assessments.pdf.

280. Steven Weil, (2003), How to obtain a high-quality vulnerability assessment, from https://searchsecurity.techtarget.com/tip/How-to-obtain-a-high-quality-vulnerability-assessment.

281. Dr. D. Polemi, G. Valvis, (2005), Vulnerability Assessment Report Format Data Model, from http://www.kti.ae.poznan.pl/conferences/i3e/papers/george_valvis.pdf.

282. (2011). What is a vulnerability assessment?, from http://resecure.me/pdf/17542.pdf.

283. Marcelo Silva, (2012), Vulnerability Assessment, from https://www.slideshare.net/slideshow/info-security-vulnerability-assessment/14461823.

284. Common Vulnerability Scoring System Calculator, from https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator.

285. (2019). Common Weakness Enumeration, from https://en.wikipedia.org/wiki/Common_Weakness_Enumeration.

286. (2015). Testing Scan Credentials for More Accurate Vulnerability Assessment, From https://www.tripwire.com/state-of-security/vulnerability-management/testing-scan-credentials-for-more-accurate-vulnerability-assessment/.

287. (2011). Credentialed vs Non-Credentialed scans, from https://success.qualys.com/discussions/s/question/0D52L00004TntpBSAR/credentialed-vs-noncredentialed-scans.

288. Syamini Sreedharan, What is Vulnerability Assessment? Testing Process, VAPT Scan Tool, from https://www.guru99.com/vulnerability-assessment-testing-analysis.html.

289. Mayra Cortes, (2020), Vulnerability Scanning: Pros, Cons and Best Practices, from https://cynexlink.com/latest-articles/pros-and-cons-of-vulnerability-scanning/.

290. Marco Van Den Oever, (2019). What are the pros and cons of vulnerability scanning tools?, from https://www.computest.nl/en/knowledge-platform/blog/what-are-pros-and-cons-vulnerability-scanning-tools/.

291. Secure Configuration Assessment, from https://www.vistainfosec.com/service/secure-configuration-assessment-service.

292. (2018). Mobile Application Security Assessments: The Best Practices to Launch and Maintain a Secure App, from https://www.secureworks.com/blog/mobile-application-security-assessments.

293. (2019). Tips For Creating a Strong Vulnerability Assessment Report, from https://blog.rsisecurity.com/tips-for-creating-a-strong-vulnerability-assessment-report/.

294. Sample Network Vulnerability Assessment Report, from https://purplesec.us/wp-content/uploads/2019/03/Sample-Network-Security-Vulnerability-Assessment-Report-Purplesec.pdf.

295. Dashboard Details, from https://docs.tenable.com/nessus/Content/Dashboard.htm.

296. (2019). How To: Run Your First Vulnerability Scan with Nessus, from https://www.tenable.com/blog/how-to-run-your-first-vulnerability-scan-with-nessus.

Module 06: System Hacking

297. Gregory Kipper, (2003). Steganography in Depth, from https://www.taylorfrancis.com/books/9780429211065.

298. S.J. Templeton, K.E. Levitt, (2003). Detecting spoofed packets, from https://ieeexplore.ieee.org/document/1194882.

299. NTLM Authentication in Java, from http://www.luigidragone.com/software/ntlm-authentication-in-java/.

300. Samir K Bandyopadhyay, Debnath Bhattacharyya, Debashis Ganguly, Swarnendu Mukherjee, and Poulami Das, A Tutorial Review on Steganography, from http://www.jiit.ac.in/jiit/ic3/IC3_2008/IC3-2008/APP2_21.pdf.

301. Ricky Magalhaes, (2003). Using passwords as a defense mechanism to improve Windows security, from http://techgenix.com/passwords_improve_windows_security_part2/.

302. Andreas Westfeld and Andreas Pfitzmann, Attacks on Steganographic Systems, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.04.5075&rep=rep1&type=pdf.

303. DeiiSanai and HidenobuSeki, (2004). Optimized Attack for NTLM2 Session Response, from https://www.blackhat.com/presentations/bh-asia-04/bh-jp-04-pdfs/bh-jp-04-seki.pdf.

304. Zhi Wang, Xuxian Jiang, Weidong Cui, and Xinyuan Wang, (2008). Countering Persistent Kernel Rootkits Through Systematic Hook Discovery, from https://www.microsoft.com/en-us/research/publication/countering-persistent-kernel-rootkits-through-systematic-hook-discovery/.

305. Brute force attack - Wikipedia, the free encyclopedia, from https://en.wikipedia.org/wiki/Brute-force_attack.

306. Passwords, from http://media.techtarget.com/searchSecurity/downloads/HackingforDummiesCh07.pdf.

307. The Hack FAQ: Password Basics, from https://www.nmrc.org/pub/faq/hackfaq/hackfaq-04.html.

308. Microsoft Technical Security Notifications, from https://www.microsoft.com/en-us/msrc/technical-security-notifications?rtc=1.

309. Bojan Smojver, (2002). Linux Today - ZDNet Australia: Threats Move Beyond Linux to Windows, From https://www.linuxtoday.com/security/2002121100426scsvnt.

310. Russell Kay, (2006). Sidebar: A Simple Rootkit Example, from https://www.computerworld.com/article/2560499/cybercrime-hacking/sidebar--a-simple-rootkit-example.html.

311. Russell Kay, (2006). Rootkits offer the lure of total control, from https://www.computerworld.com/uk/.

312. Steganography (a secretly hidden coding that dates back to ancient ...), from https://wordinfo.info/unit/3403?letter=S&spage=9.

313. Fred B. Schneider, Authentication, from http://www.cs.cornell.edu/Courses/cs513/2000sp/NL10.html.

314. CS513: System Security - Topic Outline, from http://www.cs.cornell.edu/courses/cs513/2005fa/02.outline.html.

315. Mary McMahon, (2014). What Is a Privilege Escalation?, from https://www.easytechjunkie.com/what-is-a-privilege-escalation.htm.

316. Foon, (2002). Exploiting design flaws in the Win32 API for privilege escalation - Shatter Attacks - How to break Windows, from https://www.helpnetsecurity.com/2002/08/08/exploiting-design-flaws-in-the-win32-api-for-privilege-escalation---shatter-attacks---how-to-break-windows/.

317. Scott Sutherland, (2009). Windows Privilege Escalation Part 1: Local Administrator Privileges, from https://www.netspi.com/blog/technical-blog/network-penetration-testing/windows-privilege-escalation-part-1-local-administrator-privileges/.

318. Windows Privilege Escalation Fundamentals, from http://www.fuzzysecurity.com/tutorials/16.html.

319. Siva Ram, (2010). DLL Hijacking Attacks, from https://www.maravis.com/dll-hijacking-attacks/.

320. (2018). Dynamic-Link Library Security, from https://learn.microsoft.com/en-us/windows/win32/dlls/dynamic-link-library-security?redirectedfrom=MSDN.

321. Srikanth Ramesh, How to Hack Windows Administrator Password, from https://www.gohacking.com/hack-windows-administrator-password/.

322. Aleksandr Matrosov, (2013). Mysterious Avatar rootkit with API, SDK, and Yahoo Groups for C&C communication, from https://www.welivesecurity.com/2013/05/01/mysterious-avatar-rootkit-with-api-sdk-and-yahoo-groups-for-cc-communication/.

323. Win32/Rootkit.Avatar, from https://www.welivesecurity.com/en/about-eset-research/.

324. James Wyke, The ZeroAccess rootkit, from https://news.sophos.com/en-us/category/serious-security/#Introduction

325. (2013). Hack Like a Pro: How to Cover Your Tracks & Leave No Trace Behind on the Target System, from https://null-byte.wonderhowto.com/how-to/hack-like-pro-cover-your-tracks-leave-no-trace-behind-target-system-0148123/.

326. Sarah Granger, (2002). The Simplest Security: A Guide To Better Password Practices, from https://community.broadcom.com/home.

327. Gaining Access Using Application and Operating System Attacks, from https://www.techtarget.com/searchsecurity/definition/cyber-attack.

328. Jesper M. Johenson, Windows Passwords: Everything You Need To Know, from http://download.microsoft.com/download/a/d/0/ad0f04a3-21b2-4d79-9049-f5fadb632ace/SEC401-JesperJohansson.pdf.

329. Dr-Hack, (2009). Hash injection Attacks in a Windows Network, from https://blog.dr-hack.net/hash-injection-attacks-in-a-windows-network/.

330. How to prevent Windows from storing a LAN manager hash of your password in Active Directory and local SAM databases, from https://learn.microsoft.com/en-us/troubleshoot/windows-server/windows-security/prevent-windows-store-lm-hash-password.

331. System Hacking: Part II, from http://archive.visualstudiomagazine.com/books/chapters/0072260815.pdf.

332. Fred B. Schneider, Authentication, from http://www.cs.cornell.edu/courses/cs513/2000sp/NL10.html.

333. Gary C. Kessler, (2001). Steganography: Hiding Data Within Data, https://www.garykessler.net/library/steganography.html.

334. Gary C. Kessler, (2015). An Overview of Steganography for the Computer Forensics Examiner, from, https://www.garykessler.net/library/fsc_stego.html.

335. Soumyendu Das, Steganography and Steganalysis: Different Approaches, from https://arxiv.org/ftp/arxiv/papers/1111/1111.3758.pdf.

336. Link Local Multicast Name Resolution (LLMNR) and NetBIOS Name Service (NBT-NS), from https://www.ccsecuritycenter.org/remediation/llmnr-nbt-ns.

337. Jon Sternstein, Local Network Attacks: LLMNR and NBT-NS Poisoning, from https://www.sternsecurity.com/blog/local-network-attacks-llmnr-and-nbt-ns-poisoning.

338. LLMNR / NBT-NS Spoofing Attack Network Penetration Testing, from https://www.aptive.co.uk/blog/llmnr-nbt-ns-spoofing/.

339. HollyGraceful, (2015). Stealing Accounts: LLMNR and NBT-NS Spoofing, from https://www.gracefulsecurity.com/stealing-accounts-llmnr-and-nbt-ns-poisoning/

340. Mucahit Karadag, (2016). What is LLMNR & WPAD and How to Abuse Them During Pentest?, from https://pentest.blog/what-is-llmnr-wpad-and-how-to-abuse-them-during-pentest/.

341. William Hurer-Mackay, (2016). LLMNR and NBT-NS Poisoning Using Responder, from https://www.4armed.com/blog/llmnr-nbtns-poisoning-using-responder/.

342. Link Local Multicast Name Resolution (LLMNR) and NetBIOS Name Service (NBT-NS), from https://www.ccsecuritycenter.org/remediation/llmnr-nbt-ns.

343. (2019). Exploitation for Privilege Escalation, from https://attack.mitre.org/wiki/Technique/T1068.

344. (2018). Dylib Hijacking, from https://attack.mitre.org/wiki/Technique/T1157.

345. Patrick Wardle, (2015). Dylib hijacking on OS X, from https://www.virusbulletin.com/virusbulletin/2015/03/dylib-hijacking-os-x/.

346. (2015). Dylib hijacking on OS X, from https://news.ycombinator.com/item?id=9239811.

347. (2018). Meltdown and Spectre: Critical processor vulnerabilities, from https://www.enisa.europa.eu/publications/info-notes/meltdown-and-spectre-critical-processor-vulnerabilities.

348. Justin Ellingwood, (2018). How To Protect Your Server Against the Meltdown and Spectre Vulnerabilities, from https://www.digitalocean.com/community/tutorials/how-to-protect-your-server-against-the-meltdown-and-spectre-vulnerabilities.

349. Josh Fruhlinger, (2018). Spectre and Meltdown explained: What they are, how they work, what's at risk, from https://www.csoonline.com/article/3247868/vulnerabilities/spectre-and-meltdown-explained-what-they-are-how-they-work-whats-at-risk.html.

350. Tom Ueltschi, Travis Smith, Jared Atkinson, Robby Winchester, (2017). Access Token Manipulation, from https://attack.mitre.org/wiki/Technique/T1134.

351. Access Tokens, from https://learn.microsoft.com/en-us/windows/win32/secauthz/access-tokens?redirectedfrom=MSDN.

352. (2017). Application Shimming, from https://attack.mitre.org/wiki/Technique/T1138.

353. Stefan Kanthak and Travis Smith, (2017). File System Permissions Weakness, from https://attack.mitre.org/wiki/Technique/T1044.

354. Travis Smith and Leo Loobeek, (2017), Scheduled Task, from https://attack.mitre.org/wiki/Technique/T1053.
355. (2017), Launch Daemon, from https://attack.mitre.org/wiki/Technique/T1160.
356. (2017), Plist Modification, from https://attack.mitre.org/wiki/Technique/T1150.
357. (2017), Setuid and Setgid, from https://attack.mitre.org/wiki/Technique/T1166.
358. (2017), Web Shell, from https://attack.mitre.org/wiki/Technique/T1100.
359. Agathoklis Prodromou, (2016), An introduction to web Shells-Part-1, from https://www.acunetix.com/blog/articles/introduction-web-shells-part-1/.
360. (2015), Covering Tracks of Attacks, from https://www.infosecinstitute.com/resources/hacking/covering-tracks-of-attacks/#gref.
361. (2018), Microsoft NTLM, from https://learn.microsoft.com/en-us/windows/win32/secauthn/microsoft-ntlm.
362. Amrita Mitra, (2017), What is Pass The Hash Attack?, from https://www.thesecuritybuddy.com/vulnerabilities/what-is-a-pass-the-hash-attack/.
363. (2019), Pass the hash, from https://en.wikipedia.org/wiki/Pass_the_hash.
364. (2007), Pass-The-Hash Toolkit, from https://www.coresecurity.com/corelabs-research-special/open-source-tools/pass-hash-toolkit
365. Elad Shamir, (2019), Internal Monologue Attack - Retrieving NTLM Hashes without Touching LSASS (Repost), from https://shenaniganslabs.io/2019/01/14/Internal-Monologue.html.
366. (2018), Retrieving NTLM Hashes without touching LSASS: the "Internal Monologue" Attack, from https://www.andreafortuna.org/2018/03/26/retrieving-ntlm-hashes-without-touching-lsass-the-internal-monologue-attack/.
367. Yaron Ziner, (2017), Advanced Techniques Attackers Use to Crack Passwords, from https://www.infosecinstitute.com/resources/penetration-testing/advanced-techniques-attackers-use-crack-passwords/#gref.
368. Adam Chester, (2017), Kerberos AD Attacks - More Roasting with AS-REP, from https://blog.xpnsec.com/kerberos-attacks-part-2/.
369. Jeff Petters, (2018), Kerberos Authentication Explained, from https://www.varonis.com/blog/kerberos-authentication-explained/.
370. Aki Jos, (2018), AS-REP Roasting – Cracking User Account Password, from https://akijosberryblog.wordpress.com/2018/01/17/as-rep-roasting-cracking-user-account-password/
371. Sean Metcalf, (2015), Cracking Kerberos TGS Tickets Using Kerberoast – Exploiting Kerberos to Compromise the Active Directory Domain, from https://adsecurity.org/?p=2293.
372. Ryan Becwar, and Vincent Le Toux, (2019), Pass the Ticket, from https://attack.mitre.org/techniques/T1097/.
373. Chris Stoneff, (2018), Defending Against Pass-the-Ticket Attacks, from https://www.beyondtrust.com/blog/entry/defending-against-pass-the-ticket-attacks.
374. (2017), Cracking Passwords: 11 Password Attack Methods (And How They Work), from https://datarecovery.com/rd/cracking-passwords-11-password-attack-methods-work/.
375. Jens Steube, (2013), Advanced password guessing, from https://hashcat.net/events/p13/js-ags-htf13.pdf.
376. William Hurer-Mackay, (2016), How to Perform a Combinator Attack Using Hashcat, from https://www.4armed.com/blog/hashcat-combinator-attack/.
377. Atom, (2010), Automated Password Cracking: Use oclHashcat To Launch A Fingerprint Attack, from https://www.question-defense.com/2010/08/15/automated-password-cracking-use-oclhashcat-to-launch-a-fingerprint-attack.
378. The Different Types of Password Cracking Techniques, from https://password-managers.bestreviews.net/the-different-types-of-password-cracking-techniques/.
379. Daniel Turner, (2012), Hashcat Per Position Markov Chains, from https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/hashcat-per-position-markov-chains/.
380. Lisa Bock, Defend against password attacks, from https://www.linkedin.com/learning/ethical-hacking-system-hacking/defend-against-password-attacks.
381. Daniel Doc Sewell, Offline Password Cracking: The Attack and the Best Defense, from https://www.alpinesecurity.com/blog/offline-password-cracking-the-attack-and-the-best-defense-against-it.
382. Samantha Rorke, (2017), Protecting your Network against Brute Force Password attacks, from https://www.zerofox.com/looking-glass-cyber/.
383. Adam Gordon, (2017), How to prevent password attacks and other exploits, from https://www.techtarget.com/searchsecurity/tip/How-to-prevent-password-attacks-and-other-exploits.
384. What is a buffer overflow? Learn about buffer overrun vulnerabilities, exploits & attacks, from https://www.veracode.com/security/buffer-overflow.
385. Buffer Overflow Attack with Example, from https://www.geeksforgeeks.org/buffer-overflow-attack-with-example/
386. (2020), Buffer overflow, from https://en.wikipedia.org/wiki/Buffer_overflow.
387. Differences between Stack and Heap, from http://net-informations.com/faq/net/stack-heap.htm.

388. Stack vs Heap Memory Allocation, from https://www.geeksforgeeks.org/stack-vs-heap-memory-allocation/.
389. Subhashini Rai, (2019), What is Stack Based Buffer Overflow?, from https://hackersonlineclub.com/stack-based-buffer-overflow/.
390. Heap overflow and Stack overflow, from https://www.geeksforgeeks.org/heap-overflow-stack-overflow/.
391. (2018), 32-Bit Windows Buffer Overflows Made Easy, from https://vetsec.org/.
392. Tomasz Andrzej Nidecki, (2019), What is a Buffer Overflow, from https://www.acunetix.com/blog/web-security-zone/what-is-buffer-overflow/.
393. Buffer Overflow Attack, from https://www.imperva.com/learn/application-security/buffer-overflow/.
394. Avoiding Buffer Overflows and Underflows, from https://developer.apple.com/library/archive/documentation/Security/Conceptual/SecureCodingGuide/Articles/BufferOverflows.html.
395. Buffer Overflow Defenses, from http://www.users.di.uniroma1.it/~parisi/Risorse/BO-defense.pdf.
396. Bradley Green, Exploitation Development and Implementation, from https://www.iup.edu/WorkArea/DownloadAsset.aspx?id=170683.
397. Mutaz Alsalal, (2018), Identifying Named Pipe Impersonation and Other Malicious Privilege Escalation Techniques, from https://securityintelligence.com/identifying-named-pipe-impersonation-and-other-malicious-privilege-escalation-techniques/.
398. Azeria, Privilege Escalation, from https://azeria-labs.com/privilege-escalation/.
399. (2017), Unquoted Service Path, from https://pentestlab.blog/2017/03/09/unquoted-service-path/.
400. Jonathan, (2015), Common Windows Privilege Escalation Vectors, from https://www.toshellandback.com/2015/11/24/ms-priv-esc/.
401. Abed, (2017), Pivots and Relays for Extreme Post-Exploitation Control, from https://www.semurity.com/pivots-and-relays-for-extreme-post-exploitation-control/.
402. (2012), Post Exploitation – Port Forwarding, from https://pentestlab.blog/2012/04/22/post-exploitation-port-forwarding/.
403. (2018), Access Tokens, from https://learn.microsoft.com/en-us/windows/win32/secauthz/access-tokens?redirectedfrom=MSDN.
404. Windows Shim Database (SDB) Parser (shims), from https://tzworks.net/prototype_page.php?proto_id=33.
405. (2020), sudo, from https://en.wikipedia.org/wiki/Sudo.
406. Margaret Rouse, (2005), sudo (superuser do), from https://www.techtarget.com/searchsecurity/definition/sudo-superuser-do.
407. Privilege Escalation, from https://chryzsh.gitbooks.io/pentestbook/privilege_escalation_-_linux.html.
408. Raj Chandel, (2018), Linux Privilege Escalation using SUID Binaries, from https://www.hackingarticles.in/linux-privilege-escalation-using-suid-binaries/.
409. Barrow, (2016), Use a Misconfigured SUID Bit to Escalate Privileges & Get Root, from https://null-byte.wonderhowto.com/how-to/use-misconfigured-suid-bit-escalate-privileges-get-root-0173929/.
410. David Lodge, (2015), Exploiting SUID Executables, from https://www.pentestpartners.com/security-blog/exploiting-suid-executables/.
411. Justin Ellingwood, (2018), How To Protect Your Server Against the Meltdown and Spectre Vulnerabilities, from https://www.digitalocean.com/community/tutorials/how-to-protect-your-server-against-the-meltdown-and-spectre-vulnerabilities.
412. (2018), Meltdown and Spectre Side-Channel Vulnerability Guidance, from https://www.cisa.gov/news-events/alerts/2018/01/04/meltdown-and-spectre-side-channel-vulnerability-guidance.
413. Mike Burns, (2018), Running the Intel Meltdown Detection Tool, from https://medium.com/think-stack/running-the-intel-meltdown-detection-tool-d4d92735e605.
414. James Sanders, (2019), Spectre and Meltdown explained: A comprehensive guide for professionals, from https://www.techrepublic.com/article/spectre-and-meltdown-explained-a-comprehensive-guide-for-professionals/.
415. (2019), Execution, from https://attack.mitre.org/tactics/TA0002/.
416. (2020), Keystroke logging, from https://en.wikipedia.org/wiki/Keystroke_logging.
417. (2020), Defending Against Keyloggers, from https://support.logmeininc.com/central/help/defending-against-keyloggers.
418. Linda McGlasson, (2010), How to Beat Keyloggers, from https://www.bankinfosecurity.com/how-to-beat-keyloggers-a-2999.
419. (2018), How to Protect Yourself From Keyloggers, from https://www.expressvpn.com/blog/what-is-malware/.
420. (2013), Defending against Keyloggers, from https://homegurutraining.wordpress.com/2013/08/23/defendingagainstkeylogger/.
421. Ben Alonzo, (2017), Simple tips for defending against keylogger attacks, from https://ultratechlife.com/tech/simple-tips-for-defending-against-keylogger-attack/.
422. Five ways to keep keyloggers away from your data, from https://www.northbridgeinsurance.ca/blog/five-ways-to-keep-keyloggers-away/.

423. (2020). Rootkit, from https://en.wikipedia.org/wiki/Rootkit.
424. Rootkit, from https://www.imperva.com/learn/application-security/rootkit/.
425. Andrei Raul Ardelean, Claudiu Stefan Coblis, Cristofor Ochinca, and Cristian Alexandru Istrate, Inside Scranos – A Cross Platform, Rootkit-Enabled Spyware Operation, from https://www.bitdefender.com/files/News/CaseStudies/study/253/Bitdefender-Whitepaper-RootKit-CREAT3432-en-EN.pdf.
426. Catalin Cimpanu, (2019), Scranos rootkit expands operations from China to the rest of the world, from https://www.zdnet.com/article/scranos-rootkit-expands-operations-from-china-to-the-rest-of-the-world/.
427. Garrett Gross, (2016), Rootkit Detection: Techniques and Best Practices, from https://cybersecurity.att.com/blogs/security-essentials/rootkit-detection-techniques-and-best-practices.
428. (2020), Rootkit Detection, from https://en.wikipedia.org/wiki/Rootkit#Detection.
429. Randy Franklin Smith, (2006), Defending Against Rootkits, From https://www.itprotoday.com/security/defending-against-rootkits.
430. Mike Chapple, (2015), Security Matters: Rootkit Attacks and How to Prevent Them, from http://www.gocertify.com/articles/security-matter-rootkit-attacks-and-how-to-prevent-them.html.
431. Jesus Olguin, (2016), Steganalysis, the Counterpart of Steganography, from https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/steganalysis-the-counterpart-of-steganography/.
432. Clear-EventLog, from https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.management/clear-eventlog?view=powershell-5.1.
433. Ivan Jeric, (2018), How to clear the Event Log in Windows 10, 7, from https://windowsreport.com/clear-event-log-windows-8/.
434. (2020), Atomic Test #1 - Clear Logs, from https://github.com/redcanaryco/atomic-red-team/blob/master/atomics/T1070/T1070.md#atomic-test-1---clear-logs.
435. Anand Khanse, (2018), Cipher command line tool in Windows 10, from https://www.thewindowsclub.com/cipher-command-line-tool-windows.
436. (2018), MS-DOS and Windows command line cipher command, from https://www.computerhope.com/cipher.htm.
437. Nihad Ahmad Hassan, Rami Hijazi, and Heivi Salminen, (2017), Data Hiding Techniques in Windows OS: A Practical Approach to Investigation and Defense, from https://books.google.co.in/books?id=sy2ICgAAQBAJ&pg=PA267&lpg=PA267&dq=techniques+for+wiping+attack+traces&source=bl&ots=__ahJJl6i6&sig=ACfU3U2xKfyKvtI68E4MB53SLgroMBx1w&hl=en&sa=X&ved=2ahUKEwic2ISI9pzjAhVTdCsKHvEEAaA4ChDoATANegQICRAB#v=onepage&q=cipher.exe&f=false.
438. Trisha, (2014), How to Disable Virtual Memory in Windows 10, from https://www.trishtech.com/2014/11/disable-virtual-memory-in-windows-10/.
439. (2017), Enable Last Access timestamp for files and folder in Windows, from https://www.opentechguides.com/how-to/article/windows-10/129/enable-last-access-time.html.
440. Richard Falk, (2016), How to disable Sleep Mode or Hibernation, from https://www.pugetsystems.com/support/guides/how-to-disable-sleep-mode-or-hibernation-793/.
441. Clear Thumbnails Cache – Guide for Windows XP, Vista, 7, 8, 8.1, 10, from https://ecosmart.net/wiki/clear-thumbnails-cache/.
442. (2021), Detecting Password Spraying Attacks: Threat Research Release May 2021, from https://www.splunk.com/en_us/blog/security/detecting-password-spraying-attacks-threat-research-release-may-2021.html.
443. Ryan Brooks, (2022), What Is Password Spraying, and How Can You Spot and Block Attacks?, from https://blog.netwrix.com/2020/10/28/password-spraying/.
444. Password Spraying, from https://book.hacktricks.xyz/windows/active-directory-methodology/password-spraying.
445. Password Spraying, from https://doubleoctopus.com/security-wiki/threats-and-tools/password-spraying/#.
446. (2020), How to: Use Mask Attack in hashcat, from https://www.youtube.com/watch?v=fkX0zfWH59k.
447. Mask Attack, from https://hashcat.net/wiki/doku.php?id=mask_attack.
448. Hashcat manual: How to Use the Program for Cracking Passwords, from https://miloserdov.org/?p=953.
449. Vijay Kumar, (2021), Hashcat Tutorial on Brute force and Mask Attack Step by Step Guide, from https://www.cyberpratibha.com/hashcat-tutorial-for-password-cracking/.
450. Pedro Tavares, (2020), Hashcat Tutorial for Beginners, from https://www.infosecinstitute.com/resources/hacking/hashcat-tutorial-beginners/.
451. Hoda Naghibijouybari, Ajaya Neupane, Zhiyun Qian, and Nael Abu-Ghazaleh, Rendered Insecure: GPU Side Channel Attacks are Practical, from https://www.cs.ucr.edu/~nael/pubs/ccs18.pdf.
452. (2022), Graphics Processing Unit, from https://en.wikipedia.org/wiki/Graphics_processing_unit.
453. Jolene Dobbin, (2019), GPU vs CPU: What Matters Most for PC Gaming?, from https://www.hp.com/us-en/shop/tech-takes/gpu-vs-cpu-for-pc-gaming.

454. (2018). GPU Side-Channel Attacks can Enable Spying on Web Activity Password Stealing, from https://www.helpnetsecurity.com/2018/11/06/gpu-side-channel-attacks/.

455. (2017). WEBCAST: Demo of Domain Password Audit Tool, from https://www.blackhillsinfosec.com/webcast-demo-domain-password-audit-tool/.

456. Do Son, (2018). DPAT: Domain Password Audit Tool for Pentesters, from https://securityonline.info/dpat-domain-password-audit-tool-pentesters/.

457. Tricia Howard, (2020). LLMNR & NBT-NS Poisoning and Credential Access using Responder, from https://www.cynet.com/attack-techniques-hands-on/llmnr-nbt-ns-poisoning-and-credential-access-using-responder.

458. Jon Sternstein, (2013). Local Network Attacks: LLMNR and NBT-NS Poisoning Background, from https://www.sternsecurity.com/blog/local-network-attacks-llmnr-and-nbt-ns-poisoning.

459. Eric Kuehn, Matthew Demaske, and AdoptForward, (2021). Adversary-in-the-Middle: LLMNR/NBT-NS Poisoning and SMB Relay, from https://attack.mitre.org/techniques/T1557/001.

460. (2022). Return-oriented programming, from https://en.wikipedia.org/wiki/Return-oriented_programming#:~:text=Return%2Doriented%20programming%20is%20an,program%2C%20often%20a%20buffer%20overrun.

461. Return-oriented Programming, from https://developer.arm.com/documentation/102433/0100/Return-oriented-programming.

462. (2019). Return Oriented Programming (ROP) Attacks, from https://www.infosecinstitute.com/resources/hacking/return-oriented-programming-rop-attacks/.

463. Michael Hill, (2022). Exploit Chains Explained: How and Why Attackers Target Multiple Vulnerabilities, from https://www.csoonline.com/article/3645449/exploit-chains-explained-how-and-why-attackers-target-multiple-vulnerabilities.html?upd=1642424410843.

464. Emerging Cyber Threat Vulnerability Chaining, from https://libertyadvisorgroup.com/insight/cyber-update-vulnerability-chaining/.

465. Karim Habeeb, (2021). Active Directory Domain Enumeration Part-1 With Powerview, from https://nored0x.github.io/red-teaming/active-directory-domain-enumeration-part-1/.

466. Hebbob, Active Directory Enumeration with PowerShell, from https://www.exploit-db.com/docs/english/46990-active-directory-enumeration-with-powershell.pdf.

467. Raj Chandel, (2021). Active Directory Enumeration: PowerView, from https://www.hackingarticles.in/active-directory-enumeration-powerview/.

468. AD Enumeration Toolkit, from https://academy.hackthebox.com/course/preview/active-directory-powerview/ad-enumeration-toolkit.

469. (2022). Trust Relationships Between Domains on Windows, from https://www.ibm.com/docs/en/db2/11.5?topic=windows-trust-relationships-between-domains.

470. Docusnap X - User Manual, from https://www.docusnap.com/help/docusnap-x/user/docusnap-documentation-map-files-active-directory.html.

471. Jeff Melnick, (2017). What Is an AD Domain?, from https://blog.netwrix.com/2017/01/31/active-directory-domain/.

472. Edges, from https://bloodhound.readthedocs.io/en/latest/data-analysis/edges.html.

473. (2022). BloodHound, from https://attack.mitre.org/software/S0521/.

474. Introduction to GhostPack, from https://specterops.gitbook.io/ghostpack/.

475. Defining Buffer Overflow Attacks and How to Defend Against Them, from https://www.okta.com/identity-101/buffer-overflow-attacks/.

476. Megan Kaczanowski, (2021). What is a Buffer Overflow Attack – and How to Stop It, from https://www.freecodecamp.org/news/buffer-overflow-attacks/.

477. Malav Vyas, (2020). How to Protect Against Buffer Overflow Attack, from https://www.securecoding.com/blog/how-to-protect-against-buffer-overflow-attack/.

478. (2019). Bypass UAC and Escalate Privileges on Windows Using Metasploit, from https://null-byte.wonderhowto.com/how-to/bypass-uac-escalate-privileges-windows-using-metasploit-0196076.

479. Raj Chandel, (2018). Multiple Ways to Bypass UAC using Metasploit, from https://www.hackingarticles.in/multiple-ways-to-bypass-uac-using-metasploit.

480. Casey Smith, Stefan Kanthak, (2022). Abuse Elevation Control Mechanism: Bypass User Account Control, from https://attack.mitre.org/techniques/T1548/002.

481. (2022). Boot or Logon Initialization Scripts, from https://attack.mitre.org/techniques/T1037/.

482. Mohans, (2020). Escalate Domain Privileges, from https://medium.com/redteam-blueteam-series/escalate-domain-privileges-e53ae4027856.

483. Sean Metcalf, (2016). Sneaky Active Directory Persistence #17: Group Policy, from https://adsecurity.org/?p=2716.

484. Kevin Joyce, (2021), What is DCSync Attack?, from https://blog.netwrix.com/2021/11/30/what-is-dcsync-an-introduction/.
485. Josh Van Cott, (2020), What are DCSync and DCShadow Active Directory attacks?, from https://www.lepide.com/blog/what-are-dcsync-and-dcshadow-active-directory-attacks/.
486. (2021), Domain Controller, from https://en.wikipedia.org/wiki/Domain_controller.
487. Kirsten Gantenbein, (2021), What is DCSync and How to Protect Against It, from https://www.extrahop.com/company/blog/2021/dcsync-definition-and-protection/.
488. QOMPLX Knowledge: DCSync Attacks Explained, from https://www.qomplx.com/kerberos_dcsync_attacks_explained/.
489. DCSync Attack: Definition, Examples, and Prevention, from https://www.extrahop.com/resources/attacks/dcsync/.
490. Jeff Petters, (2020), What is Mimikatz: The Beginner's Guide, from https://www.varonis.com/blog/what-is-mimikatz.
491. Sean Metcalf, (2015), Mimikatz DCSync Usage, Exploitation, and Detection, from https://adsecurity.org/?p=1729.
492. Wayne Silva, (2021), Access Token Manipulation: Parent PID Spoofing, from https://attack.mitre.org/techniques/T1134/004/.
493. (2020), Parent PID Spoofing, from https://pentestlab.blog/2020/02/24/parent-pid-spoofing/.
494. Saurav Lacoul, (2020), Parent Process ID (PPID) Spoofing, from https://cybdefnp.wordpress.com/2020/07/05/parent-process-id-ppid-spoofing/.
495. Parent PID Spoofing, from https://dmcxblue.gitbook.io/red-team-notes-2-0/red-team-techniques/defense-evasion/untitled-1/parent-pid-spoofing.
496. (2021), Parent PID Spoofing – Defense Evasion, Privilege Escalation, from https://zerodollarsoc.com/attack-defence-catalogue/techniques/t1134-004-parent-pid-spoofing-defense-evasion-privilege-escalation/.
497. Paul Speulstra, (2020), Event Triggered Execution: Accessibility Features, from https://attack.mitre.org/techniques/T1546/008/.
498. Alain Homewood and Vincent Le Toux, (2021), Access Token Manipulation: SID-History Injection, from https://attack.mitre.org/techniques/T1134/005/.
499. Dcurwin, Shsagir, and Mjmbaldwin, (2022), Security assessment: Unsecure SID History Attributes, from https://learn.microsoft.com/en-us/defender-for-identity/security-assessment-unsecure-sid-history-attribute.
500. Vikram Navali, (2021), Windows Security Identifier (SID) History Injection Exposure, from https://www.sentinelone.com/blog/windows-sid-history-injection-exposure-blog/.
501. (2020), Event Triggered Execution: Component Object Model Hijacking, from https://attack.mitre.org/techniques/T1546/015/.
502. Giovanni López, (2020), Component Object Model Hijacking, from https://attackiq.com/2020/09/26/component-object-model-hijacking/.
503. COM Hijacking for Persistence, from https://cyberstruggle.org/com-hijacking-for-persistence/.
504. Yaniv Assor, (2018), COM Hijacking – Windows Overlooked Security Vulnerability, from https://www.cyberbit.com/blog/endpoint-security/com-hijacking-windows-overlooked-security-vulnerability/.
505. Persistence: Component Object Model (COM) Hijacking, from https://stmxcsr.com/persistence/com-hijacking.html.
506. (2017), Use Keylogger in Metasploit Framework, from https://www.yeahhub.com/use-keylogger-in-metasploit-framework/.
507. Using a Keylogger with Metasploit, from https://www.offensive-security.com/metasploit-unleashed/keylogging/.
508. (2020), Remote Keylogger Attack using Metasploit | Penetration Testing | J Techcode, from https://techcode.blogspot.com/2020/01/remote-keylogger-attack-using.html.
509. 5 Step Using Metasploit Meterpreter Keylogger (Keylogging), from https://www.hacking-tutorial.com/hacking-tutorial/5-step-using-metasploit-meterpreter-keylogger-keylogging/#sthash.vjh5QN3q.dpbs.
510. Linux Post-exploitation, from https://pentestwiki.org/post-exploitation/.
511. Finding Writable Files, from https://www.oreilly.com/library/view/linux-security-cookbook/0596003919/ch09s11.html#:~:text=The%20chmod%20command%20can%20disable,other%20bits%20for%20further%20restrictions.
512. (2021), Using OpenSSL s_client Commands to Test SSL Connectivity, from https://docs.pingidentity.com/bundle/solution-guides/page/lqs1569423823079.html.
513. (2018), Authenticated WMI Exec via Powershell, from https://www.rapid7.com/db/modules/exploit/windows/local/ps_wmi_exec/.
514. (2020), Powershell: Calculate File Hash - MD5, SHA256, SHA1, from https://www.top1p.ca/2020/10/powershell-calculate-file-hash-md5.html.
515. Mark Russinovich, (2021), PsExec, from https://learn.microsoft.com/en-us/sysinternals/downloads/psexec.
516. Christian, (2020), How to query a list of installed programs in Windows via Windows Settings, Control Panel, WMIC, PowerShell and Windows Registry, from https://techdirectarchive.com/2020/08/17/how-to-query-a-list-of-installed-programs-in-windows-via-windows-settings-control-panel-wmic-powershell-and-windows-registry/.

517. Ross Doughty, (2012), How to: Rebooting From CMD using WMIC, from https://community.spiceworks.com/how_to/6205-rebooting-from-cmd-using-wmic.

518. (2019), Windows: List Services – CMD and PowerShell, from https://www.shellhacks.com/windows-list-services-cmd-powershell/.

519. Srini, Enable/Disable Firewall From Command Line, from https://www.windows-commandline.com/enable-disable-firewall-command-line/.

520. Mitch Bartlett, Command to Add or Remove Computer from Domain, from https://www.technipages.com/command-to-add-or-remove-computer-from-domain.

521. Raj Chandel, (2020), Lateral Movement: Over Pass the Hash, from https://www.hackingarticles.in/lateral-movement-over-pass-the-hash/.

522. QOMPLX Knowledge: OverPass The Hash Attacks, from https://www.qomplx.com/qomplx-knowledge-overpass-the-hash-attacks/.

523. Over Pass the Hash/Pass the Key, From https://book.hacktricks.xyz/windows/active-directory-methodology/over-pass-the-hash-pass-the-key

524. Jeff Warren, (2019), How to Detect Overpass-the-Hash Attacks, from https://blog.netwrix.com/2022/10/04/overpass-the-hash-attacks/.

525. Blake Strom and Travis Smith, (2021), Use Alternate Authentication Material: Pass the Hash, from https://attack.mitre.org/techniques/T1550/002/.

526. (2020), Boot or Logon Initialization Scripts: Logon Script (Windows), from https://attack.mitre.org/techniques/T1037/001/.

527. (2021), Boot or Logon Autostart Execution – Persistence, Privilege Escalation, from https://aerodollarsec.com/attack-defence-catalogue/techniques/t1547-boot-or-logon-autostart-execution-persistence-privilege-escalation/.

528. Raj Chandel, (2021), Windows Privilege Escalation: Logon Autostart Execution (Registry Run Keys), from https://www.hackingarticles.in/windows-privilege-escalation-logon-autostart-execution-registry-run-keys/.

529. Raj Chandel, (2021) , Windows Privilege Escalation: Boot Logon Autostart Execution (Startup Folder), from https://www.hackingarticles.in/windows-privilege-escalation-boot-logon-autostart-execution-startup-folder/#:~:text=Raj%20Chandel%27s%20Blog-,Windows%20Privilege%20Escalation%3A%20Boot%20Logon%20Autostart%20execution%20(Startup%20Folder),escalate%20privileges%20or%20persistence%20attacks.&text=When%20a%20user%20signs%20in,the%20Registry%20or%20startup%20folder.

530. Dcurwin, Shsagir, and Mimbaldwin, (2021), Tutorial: Domain Dominance Playbook, from https://learn.microsoft.com/en-us/defender-for-identity/manage-security-alerts.

531. Ed Williams and Edward Millington, (2021), OS Credential Dumping: LSASS Memory, from https://attack.mitre.org/techniques/T1003/001/.

532. (2020), Dridex – From Word to Domain Dominance, from https://thedfirreport.com/2020/08/03/dridex-from-word-to-domain-dominance/.

533. (2021), System Services: Service Execution, from https://attack.mitre.org/techniques/T1569/002/.

534. (2022), DPAPI - Extracting Passwords, from https://book.hacktricks.xyz/windows/windows-local-privilege-escalation/dpapi-extracting-passwords.

535. Jeff Warren, (2017), Extracting User Password Data With Mimikatz DCSync, from https://blog.netwrix.com/2022/09/30/extracting-user-password-data-with-mimikatz-dcsync/.

536. QOMPLX Knowledge: Skeleton Key Attack Detection, from https://www.qomplx.com/qomplx-knowledge-skeleton-key-attack-detection

537. (2018), Skeleton Key, from https://pentestlab.blog/2018/04/10/skeleton-key

538. (2022), Skeleton Key, from https://book.hacktricks.xyz/windows/active-directory-methodology/skeleton-key.

539. Microsoft Active Directory Golden Ticket Attacks Explained: QOMPLX Knowledge, from https://www.qomplx.com/qomplx-knowledge-golden-ticket-attacks-explained/.

540. Bryan Patton, (2021), Golden ticket attacks: How they work — and how to defend against them, from https://blog.quest.com/golden-ticket-attacks-how-they-work-and-how-to-defend-against-them/.

541. (2019), Golden Ticket, from https://www.hypr.com/golden-ticket/.

542. Kirsten Gantenbein, (2021), What are Kerberos Golden Ticket Attacks and How to Detect Them, from https://www.extrahop.com/company/blog/2021/detect-kerberos-golden-ticket-attacks/.

543. Jeff Petters, (2020), Kerberos Attack: How to Stop Golden Tickets?, from https://www.varonis.com/blog/kerberos-how-to-stop-golden-tickets.

544. (2022), Domain Persistence – AdminSDHolder, from https://pentestlab.blog/2022/01/04/domain-persistence-adminsdholder/.

545. Raj Chandel, (2020), Domain Persistence AdminSDHolder, from https://www.hackingarticles.in/domain-persistence-adminsdholder/.

546. AdminSDHolder Attack, from https://www.netwrix.com/adminsdholder_modification_ad_persistence.html.

547. (2021). Backdooring AdminSDHolder for Persistence, from https://www.ired.team/offensive-security-experiments/active-directory-kerberos-abuse/how-to-abuse-and-backdoor-adminsdholder-to-obtain-domain-admin-persistence.
548. (2020). Persistence – WMI Event Subscription, from https://pentestlab.blog/2020/01/21/persistence-wmi-event-subscription/.
549. Brent Murphy and David French. (2022). Event Triggered Execution: Windows Management Instrumentation Event Subscription, from https://attack.mitre.org/techniques/T1546/003/.
550. Raj Chandel, (2020), Defense Evasion: Hide Artifacts, from https://www.hackingarticles.in/defense-evasion-hide-artifacts/.
551. Lori Kaufman, (2018), How to Hide Files and Folders From Prying Eyes on Linux, from https://www.makeuseof.com/tag/hide-files-folders-linux/.
552. Sandy Writtenhouse. (2022). How to See Hidden Files on Your Mac, from https://www.makeuseof.com/tag/show-hidden-files-mac/.
553. (2022). Hide Artifacts, from https://attack.mitre.org/techniques/T1564/
554. (2020). What is Rootkit and How to Prevent yourself from such Malware, from https://www.adwebtech.com/a-detail-guide-on-rootkit/
555. What are Rootkits and How to Prevent Them?, from https://enterprise.xcitium.com/what-are-rootkits/.
556. (2022). Everything You Need to Know About Rootkits and How to Protect Yourself, from https://www.avg.com/en/signal/what-is-rootkit.
557. (2020). What You Need to Know About Rootkits, from https://www.ionos.com/digitalguide/server/security/what-is-a-rootkit/.
558. Andre Leibovici, (2018), How to protect yourself against keyloggers, from https://www.citrix.com/blogs/2022/01/18/protect-against-keyloggers/.
559. Tushar Panhalkar, Defend Against Key loggers, from https://info-savvy.com/defend-against-key-loggers/.
560. Five Ways to Keep Keyloggers Away from Your Data, from https://www.northbridgeinsurance.ca/blog/five-ways-to-keep-keyloggers-away/.
561. Egle Juodyte, (2021). What is a keylogger?, from https://nordvpn.com/blog/keylogger-protection/.
562. Daniel Redfern, (2021), Using mimikatz not samDump2 for windows 10 password extraction., from https://technicalconfessions.com/blogs/2021/using-samdump-for-windows-password-extraction/.
563. (2024). ASREPRoast, from https://book.hacktricks.xyz/windows-hardening/active-directory-methodology/asreproast.
564. Joe Dibley, (2023), Cracking Active Directory Passwords with AS-REP Roasting, from https://blog.netwrix.com/2022/11/03/cracking_ad_password_with_as_rep_roasting/.
565. Karan Patel, (2023). AS-REP Roasting, from https://redfoxsec.com/blog/as-rep-roasting/.
566. (2023), Remote NTLM Relay Attack | Relay through a Proxy, from https://systemweakness.com/remote-ntlm-relay-attack-relay-through-a-proxy-7f155dc478b2.
567. Heath Adams, (2023), SMB Relay Attacks and How to Prevent Them, from https://tcm-sec.com/smb-relay-attacks-and-how-to-prevent-them/.
568. (2021), Windows Security Updates for Hackers, from https://blog.bitsadmin.com/windows-security-updates-for-hackers.
569. (2024). Metasploit Architecture, from https://www.offsec.com/metasploit-unleashed/metasploit-architecture/.
570. P. Raquel B. (2023). How to develop exploits, from https://www.linkedin.com/pulse/how-develop-exploits-p-raquel-bise--hy1ue/.
571. Luis Soares, (2023), Understanding Heap Spraying Attacks, from https://www.linkedin.com/pulse/understanding-heap-sprayingattacks-luis-soares-m-sc-/.
572. Examples of vulnerability exploit techniques, from https://documents.managedprotection.pandasecurity.com/PAD360/Help/v77000/Customers/Default/de-de/213.htm
573. Alyssa Snow, (2023), Abusing Active Directory Certificate Services – Part One, from https://www.blackhillsinfosec.com/abusing-active-directory-certificate-services-part-one/.
574. Rashid Feroze, (2023), Linux Privilege Escalation Guide, from https://payatu.com/blog/a-guide-to-linux-privilege-escalation/.
575. Andrew Oliveau, (2023), Escalating Privileges via Third-Party Windows Installers, from https://cloud.google.com/blog/topics/threat-intelligence/privileges-third-party-windows-installers/.
576. Erika Noerenberg and Jimmy Astle, (2022), Abuse Elevation Control Mechanism: Elevated Execution with Prompt, from https://attack.mitre.org/techniques/T1548/004/.
577. Ron Ben Yizhak, (2023), #NoFilter -Abusing Windows Filtering Platform for Privilege Escalation, from https://www.deepinstinct.com/blog/nofilter-abusing-windows-filtering-platform-for-privilege-escalation.
578. (2023), NoFilter Attack: Sneaky Privilege Escalation Method Bypasses Windows Security, from https://thehackernews.com/2023/08/nofilter-attack-sneaky-privilege.html.
579. Jan Vojtěšek, (2024), Lazarus and the FudModule Rootkit: Beyond BYOVD with an Admin-to-Kernel Zero-Day, from https://decoded.avast.io/janvojtesek/lazarus-and-the-fudmodule-rootkit-beyond-byovd-with-an-admin-to-kernel-zero-day/.

580. Rotem Sde-Or and Eliran Voronovitch, (2022). New Milestones for Deep Panda: Log4Shell and Digitally Signed Fire Chili Rootkits, from https://www.fortinet.com/blog/threat-research/deep-panda-log4shell-fire-chili-rootkits.

581. Brink, (2022). Clear Activity History in Windows 11, from https://www.elevenforum.com/t/clear-activity-history-in-windows-11.7814/.

582. Laura Klusaite, (2022). How to see and delete your incognito history, from https://nordvpn.com/blog/incognito-history/.

583. Pranav Bhardwaj, (2023). How to Delete Private Browsing History and Protect Your Privacy, from https://www.makeuseof.com/delete-incognito-history/.

Module 07: Malware Threats

584. The corporate threat posed by email Trojans, from https://www.gfi.com/de.

585. Fausi Qattan & Fredrik Thernelius, (2004). Master's Thesis, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.112.1869&rep=rep1&type=pdf.

586. Commodon Communications - Threats to your Security on the Internet, from http://www.commodon.com/threat/index.htm.

587. van Hauser/THC. Placing Backdoors Through Firewalls, from https://www.cgisecurity.com/lib/placing_backdoors_through_firewalls.txt.

588. David Wells, (1996). Wrappers, from http://www.objs.com/survey/wrap.htm.

589. Trojans FAQ, from http://techgenix.com/trojans-faq/.

590. How to block ICMP tunneling?, from https://listserv.icsalabs.com/pipermail/firewall-wizards/1999-July/006060.html.

591. Candid Wueest, (2015). Financial Trojans in 2014: Takedowns contributed to 53 percent drop in infections, but threat is still prevalent, from https://community.broadcom.com/home.

592. Candid Wueest, (2015). The state of financial Trojans 2014, from https://www.symantec.com/content/dam/symantec/docs/white-papers/state-of-financial-trojans-2014-en.pdf.

593. (2013). Battling with Cyber Warriors- Exploit Kits, from https://www.infosecinstitute.com/resources/hacking/battling-cyber-warriors-exploit-kits/.

594. Joshua Cannell, (2013). Tools of the Trade: Exploit Kits, from https://www.malwarebytes.com/blog/news/2013/02/tools-of-the-trade-exploit-kits.

595. Pierluigi Paganini, (2014). New private Exploit-Kit "Infinity" on the underground, from https://securityaffairs.com/25013/cyber-crime/new-private-exploit-kit-infinity-available-underground.html.

596. Marco Preuss, (2013). ChewBacca - a New Episode of Tor-based Malware, from https://securelist.com/chewbacca-a-new-episode-of-tor-based-malware/58192/.

597. Yotam Gottesman, (2014). RSA Uncovers New POS Malware Operation Stealing Payment Card & Personal Information, from https://community.rsa.com/community/products/netwitness/blog/2014/01/30/rsa-uncovers-new-pos-malware-operation-stealing-payment-card-personal-information.

598. Pierluigi Paganini, (2012). Skynet, the potential use of Tor as a bulletproof botnet, from https://securityaffairs.com/10960/cyber-crime/skynet-the-potential-use-of-tor-as-a-bulletproof-botnet.html.

599. (2013). SpyGate RAT, from http://spygate-rat.blogspot.in/.

600. Marshall Brain, How Computer Viruses Work, from http://www.mindpride.net/root/Extras/how-stuff-works/how_computer_viruses_work.htm.

601. Virus Protection, from http://www.mindpride.net/root/services/virus_alert_map_advisory.htm.

602. Paul Boutin, (2003). An inside view of the worm that crashed the Internet in 15 minutes, from https://www.wired.com/2003/07/slammer/.

603. Mark Russinovich, (2019). Autoruns for Windows v13.96, from https://learn.microsoft.com/en-us/sysinternals/downloads/autoruns.

604. Mark Russinovich, (2011). TCPView v3.05, from https://learn.microsoft.com/en-us/sysinternals/downloads/tcpview.

605. Norman Book on Computer Viruses, from http://download.norman.no/manuals/eng/BOOKON.PDF.

606. Carey Nachenberg, Understanding and Managing Polymorphic Viruses, from https://www.symantec.com/avcenter/reference/striker.pdf.

607. Dr. Alan Solomon and Robert M. Slade, 1990 - VX BBS & Little Black Book [AT&T Attack], 1991 - Tequila, 2001 - Gnuman, Winux Windows/Linux Virus, 2004 - Trojan.Xombe, Randex, Bizex, Witty, from https://www.cknow.com/cms/vtutor/virus-history-summary.html.

608. Ransomware, from https://en.wikipedia.org/wiki/Ransomware.

609. ANGLER EK AND ANOTHER CRYPTOWALL SAMPLE, from http://malware-traffic-analysis.net/2014/05/28/index.html.

610. GENERAL REMOVAL INSTRUCTIONS, from https://www.f-secure.com/v-descs/guides/general-removal-instructions.shtml.

611. Praneeth, (2012), DeadLine's Virus Maker 1.8.5, from http://internetfalcon.blogspot.in/2012/11/deadlines-virus-maker-185.html.
612. Computer Worms, from https://userpages.umbc.edu/~dgorin1/432/worms.htm
613. (2014), Exploit Kits, from https://www.trustwave.com/Resources/SpiderLabs-Blog/Terror-Exploit-Kit--More-like-Error-Exploit-Kit/?page=1&year=0&month=0&tag=Exploit+Kits&LangType=1033.
614. Worm, from https://www.trendmicro.com/vinfo/us/threat-encyclopedia/malware/worm.
615. R. A. Hettinga, (2003), Random Scanning Worms and Sapphire/Slammer's PRNG..., from https://www.mail-archive.com/cryptography@wasabisystems.com/msg03503.html.
616. King Adnan Anjum, (2010), Reverse WWW Shell - Covert Channels Using HTTP, from https://hackguideiu.blogspot.com/2010/03/reverse-www-shell-covert-channels-using.html.
617. DaBoss, (2013), File Extensions, from https://www.cknow.com/cms/vtutor/file-extensions.html.
618. DaBoss, (2013), Companion Files, from https://www.cknow.com/cms/vtutor/companion-files.html.
619. Ed Skoudis, (2003), Trojan horses, from http://www.informit.com/articles/article.aspx?p=102181&seqNum=2.
620. Shahram Monshi Pouri, Nikunj Modi, Trojans and Backdoors, from https://www2.it.uu.se/itwiki.php?page=edu/course/homepage/sakdat/ht06/assignments/pm/programme/modi-monshipouri.pdf&action=browse.
621. (2016), What Is A Banking Trojan And How Does It Work, from https://thecentexguy.com/what-is-a-banking-trojan-and-how-does-it-work/.
622. Swati Khandelwal, (2017), New Windows Trojan Spreads MIRAI Malware To Hack More IoT Devices, from https://thehackernews.com/2017/02/mirai-iot-botnet-windows.html#author-info.
623. Catalin Cimpanu, (2017), New Malware Intentionally Bricks IoT Devices, from https://www.bleepingcomputer.com/news/security/new-malware-intentionally-bricks-iot-devices/.
624. Farzad, (2010), Introduction to Trojans and Backdoors, from https://www.symantec.com/connect/articles/introduction-trojans-and-backdoors.
625. (2017), Virus hoax, from https://www.sophos.com/en-us/threat-center/threat-analyses/hoaxes/virus-hoax.aspx.
626. What is the FAT Virus?, from https://www.easytechjunkie.com/what-is-the-fat-virus.htm#comments.
627. Rohit Dubey, Security and Malware - Web Scripting Viruses, from http://touque.ca/EC/Students/DubeyR/ics_minor_project.html.
628. Margaret Rouse, (2007), E-Mail Virus, from https://searchmidmarketsecurity.techtarget.com/definition/email-virus.
629. E-Mail Virus, from https://www.techopedia.com/definition/15802/email-virus.
630. Common Trojan Ports, from https://www.pcsecurityworld.com/75/common-trojan-ports.html.
631. (2019), Commonly Used Port, from https://attack.mitre.org/techniques/T1043/.
632. (2019), Point-of-sale malware, from https://en.wikipedia.org/wiki/Point-of-sale_malware.
633. (2013), Point-of-Sale Malware Threats, from https://www.secureworks.com/research/point-of-sale-malware-threats.
634. Sergey Yunakovsky, (2017), Neutrino modification for POS-terminals, from https://securelist.com/neutrino-modification-for-pos-terminals/78839/.
635. (2016), Point of Sale (POS), from https://www.malwarebytes.com/blog/threats/point-of-sale-pos.
636. (2017), New Trojan Attacks Point-Of-Sale Systems Seeking Card Info, from https://www.cyberiant.com/2017/07/26/new-trojan-attacks-point-of-sale-systems-seeking-card-info/.
637. Yiyi Miao, (2017), Most Destructive Malware of All Time, from https://www.opswat.com/blog/most-destructive-malware-all-time.
638. (2020), Trojan Downloader, from https://www.malwarebytes.com/blog/detections/trojan-downloader.
639. Norah Armstrong, Inside-Out Attacks. Covert Channel Attacks Inside-out Attacks, from https://docplayer.net/864961-Inside-out-attacks-ivan-buetler-csnc-ch-covert-channel-attacks-inside-out-attacks-seite-1-glarnischstrasse-7-postfach-1671-ch-8640-rapperswil.html.
640. (2015), RIG Exploit Kit Source Code Leak - The End or Just the Beginning of RIG?, from https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/rig-exploit-kit-source-code-leak-the-end-or-just-the-beginning-of-rig/.
641. Ransomware, from https://www.trendmicro.com/vinfo/us/security/news/ransomware/page/1.
642. Penny Hoelscher, (2019), What are virus hoaxes, from https://www.comparitech.com/antivirus/virus-hoax/.
643. Ventsislav Krastev, (2018), Google Critical Security Alert Virus Scam (Gmail) – How to Remove (2019), from https://sensorstechforum.com/google-critical-security-alert-virus-scam-gmail-remove/.
644. Moseley, (2019), Two – thirds of all antivirus applications in Android are Fraud, from https://cybersguards.com/two-thirds-of-all-antivirus-applications-in-android-are-fraud/.
645. Catalin Cimpanu, (2019), Two-thirds of all Android antivirus apps are frauds, from https://www.zdnet.com/article/two-thirds-of-all-android-antivirus-apps-are-frauds/.

646. (2019). Fileless threats, from https://learn.microsoft.com/en-us/defender-endpoint/malware/fileless-threats.

647. Mary Branscombe, (2019). What is fileless malware and how do you protect against it?, from https://www.techrepublic.com/article/what-is-fileless-malware-and-how-do-you-protect-against-it/.

648. Kate Brew, (2019). Fileless Malware Detection: A Crash Course, from https://cybersecurity.att.com/blogs/security-essentials/fileless-malware-detection.

649. Simon Florcza, Donald Castillo, and Mark Manahan, (2018). Security 101: Defending Against Fileless Malware, from https://www.trendmicro.com/vinfo/in/security/news/security-technology/security-101-defending-against-fileless-malware#documentexploits.

650. Lenny Zeltser, (2018). How Fileless Malware Infections Start, from https://www.rapid7.com/solutions/unified-mdr-xdr-vm/.

651. Jareth, (2017). Fileless malware: Invisible threat or scaremongering hype?, from https://www.emsisoft.com/en/blog/29070/fileless-malware-attacks/.

652. What is Fileless Malware?, from https://www.trellix.com/en-in/security-awareness/ransomware/what-is-fileless-malware/.

653. Fileless Malware Attacks, from https://d3pakblog.wordpress.com/2018/05/05/d34n6_fileless-malware-attacks-intro/.

654. Pedro Tavares, (2018). The Art of Fileless Malware, from https://www.infosecinstitute.com/resources/threat-intelligence/art-fileless-malware/#gref.

655. Edmund Brumaghin, (2019). Divergent: "Fileless" NodeJS Malware Burrows Deep Within the Host, from https://blog.talosintelligence.com/2019/09/divergent-analysis.html.

656. Manohar Ghule and Mohd Sadique, (2019). Fileless malware campaign roundup, from https://www.zscaler.com/blogs/research/fileless-malware-campaign-roundup.

657. (2019). Vaporworm Threat: What You Need to Know About the Next Evolution of Malware, from https://www.avertium.com/vaporworm-threat-what-you-need-to-know-about-the-next-evolution-of-malware/.

658. Sudais Asif, (2019). Thousands of Windows PCs infected by Nodersok/Divergent fileless malware, from https://www.hackread.com/windows-pcs-infected-nodersok-divergent-fileless-malware/.

659. Dor Zvi, (2019). Obfuscated Fileless Malware in Cyberattackers' Toolkits: A Closer Look, from https://www.mimecast.com/blog/2019/06/obfuscated-fileless-malware-in-cyberattackers-toolkits-a-closer-look/.

660. Adam Mansour, (2017). How to Block Fileless Malware, from https://www.intelligentnetworks.com/blog/fileless-malware.

661. David Strom, (2019). How to Defend Your Organization Against Fileless Malware Attacks, from https://securityintelligence.com/how-to-defend-your-organization-against-fileless-malware-attacks/.

662. (2018). Fileless Malware: What It Is and How to Stop It, from https://www.tripwire.com/state-of-security/security-awareness/fileless-malware-stop/.

663. Sharron Malaver, (2018). How to Protect Against Fileless Malware Attacks, from https://blog.minerva-labs.com/how-to-protect-against-fileless-malware-attacks.

664. Margaret Rouse, (2019). Fileless malware attack, from https://www.techtarget.com/whatis/definitions/f.

665. Stephen Cooper, (2018). Fileless malware attacks explained, from https://www.comparitech.com/blog/information-security/fileless-malware-attacks/.

666. (2018). Fileless Malware the Stealth Attacker, from https://www.allot.com/resources/TB_FILELESS_MALWARE_THREAT_BULLETIN.pd_.pdf.

667. Trevogh Stankard, (2021). New Ransomware Technique: RTF Template Injection, from https://www.trianhq.com/blog/ransomware-technique-rtf-template-injection/.

668. Jeff Burt, (2021). Nation-State Attackers Use RTF Injection to Easily Spread Malware, from https://www.esecurityplanet.com/threats/nation-state-attackers-rtf-injection-malware/.

669. Ofer Caspi, (2021). AT&T Alien Labs finds new Golang malware (BotenaGo) targeting millions of routers and IoT devices with more than 30 exploits, from https://cybersecurity.att.com/blogs/labs-research/att-alien-labs-finds-new-golang-malware-botenago-targeting-millions-of-routers-and-iot-devices-with-more-than-30-exploits.

670. Lisa Vaas, (2022). BotenaGo Botnet Code Leaked to GitHub, Impacting Millions of Devices, from https://threatpost.com/botenago-botnet-code-leaked-to-github/178059/.

671. John Hammond, (2022). Long Live Log4Shell': CVE-2021-44228 Not Dead Yet, From https://threatpost.com/log4shell-cve-2021-44228/178225/.

672. Threat Landscape Dashboard Exploit Kits, from https://www.mcafee.com/enterprise/fr-ca/threat-center/threat-landscape-dashboard_mcm_moved/exploit-kits0.html.

673. Lawrence Abrams, (2021). ALPHV BlackCat - This Year's Most Sophisticated Ransomware, from https://www.bleepingcomputer.com/news/security/alphv-blackcat-this-years-most-sophisticated-ransomware/.

674. Avigayil Mechtinger, (2020). ELF Malware Analysis 101: Linux Threats No Longer an Afterthought, from https://www.intezer.com/blog/malware-analysis/elf-malware-analysis-101-linux-threats-no-longer-an-afterthought/.

675. Dan Goodin, (2022), Booby-trapped Sites Delivered Potent New Backdoor Trojan to macOS Users, from https://arstechnica.com/information-technology/2022/01/booby-trapped-sites-delivered-potent-new-backdoor-trojan-to-macos-users/.

676. Overview of the Mach-O Executable Format, from https://developer.apple.com/library/archive/documentation/Performance/Conceptual/CodeFootprint/Articles/MachOOverview.html.

677. Phil Stokes, (2019), How to Reverse Malware on macOS Without Getting Infected | Part 2, from https://www.sentinelone.com/blog/how-to-reverse-macos-malware-part-two/.

678. Romain Thomas, (2017), LIEF - Library to Instrument Executable Formats, from https://blog.quarkslab.com/lief-library-to-instrument-executable-formats.html.

679. Avigayil Mechtinger, (2021), ELF Malware Analysis 101: Part 3 - Advanced Analysis, from https://www.intezer.com/blog/malware-analysis/elf-malware-analysis-101-part-3-advanced-analysis/.

680. Ben Martens, What Is a Backdoor & How to Prevent Backdoor Attacks (2022), from https://www.safetydetectives.com/blog/what-is-a-backdoor-and-how-to-protect-against-it/.

681. Backdoor Attack, from https://www.imperva.com/learn/application-security/backdoor-shell-attack/.

682. Hardware Trojan Attacks and Countermeasures, from https://www.techdesignforums.com/practice/guides/hardware-trojan-security-countermeasures/.

683. Combating Fileless Attacks, from https://www.blackberry.com/us/en/solutions/fileless-attacks.

684. Fileless Malware Attacks, from https://d3pakblog.wordpress.com/2018/05/05/d34n6_fileless-malware-attacks-intro/

685. Tomas Meskauskas, (2024), How to eliminate the Mallox ransomware from a computer?, from https://www.pcrisk.com/removal-guides/22190-mallox-ransomware.

686. Lior Rochberger and Shimi Cohen, (2023), Threat Group Assessment: Mallox Ransomware, from https://unit42.paloaltonetworks.com/mallox-ransomware/.

687. (2023), IoT devices and Linux-based systems targeted by OpenSSH trojan campaign, from https://www.microsoft.com/en-us/security/blog/2023/06/22/iot-devices-and-linux-based-systems-targeted-by-openssh-trojan-campaign/.

688. Asheer Malhotra, Holger Unterbrink, Vitor Ventura, and Arnaud Zobec, (2024), TinyTurla Next Generation - Turla APT spies on Polish NGOs, from https://blog.talosintelligence.com/tinyturla-next-generation/.

689. Dennis Fisher, (2024), Tinyturla-Ng Backdoor Has Big Capabilities, from https://duo.com/decipher/tinyturla-ng-backdoor-has-big-capabilities.

690. (2024), NSFOCUS Reveals New Botnet Family RDDoS, from https://nsfocusglobal.com/nsfocus-reveals-new-botnet-family-rddos/.

691. (2023), Reptile Malware Targeting Linux Systems, from https://asec.ahnlab.com/en/55785/#:~:text=Reptile%20is%20a%20Linux%20kernel,numerous%20attack%20cases%20being%20discovered.

692. (2022), Prilex: the pricey pricide credit card complex, from https://securelist.com/prilex-atm-pos-malware-evolution/107551/.

693. (2022), Prilex Point-of-Sale malware back with new capabilities, from https://www.broadcom.com/support/security-center/protection-bulletin/prilex-point-of-sale-malware-back-with-new-capabilities.

694. (2023), Android Banking Trojan Chameleon can now bypass any Biometric Authentication, from https://www.threatfabric.com/blogs/android-banking-trojan-chameleon-is-back-in-action.

695. Fernando Merces, Augusto Remillano II, and Jeminah Molina, (2020), Mirai Botnet Attack IoT Devices via CVE-2020-5902, from https://www.trendmicro.com/en_in/research/20/g/mirai-botnet-attack-iot-devices-via-cve-2020-5902.html.

696. (2023), Trojan:Win32/ZeroBot!MTB, from https://www.microsoft.com/en-us/wdsi/threats/malware-encyclopedia-description?Name=Trojan:Win32/ZeroBot!MTB&threatId=-2147130043.

697. (2022), Behavior:Linux/Xorddos.A, from https://www.microsoft.com/en-us/wdsi/threats/malware-encyclopedia-description?Name=Behavior:Linux/Xorddos.A&threatId=0.

698. (2024), ESET takes part in global operation to disrupt the Grandoreiro banking trojan, from https://www.welivesecurity.com/en/eset-research/eset-takes-part-global-operation-disrupt-grandoreiro-banking-trojan/.

699. Dhivya, (2024), Hackers Using Weaponized Virtual Hard Disk Files to Deliver Remcos RAT, from https://cybersecuritynews.com/deliver-remcos-rat/.

700. Sead Fadilpašić, (2024), This sneaky Android malware has an all-new way to avoid being detected, from https://www.techradar.com/pro/security/this-sneaky-android-malware-has-an-all-new-way-to-avoid-being-detected.

701. (2022), Downloaders Currently the Most Prevalent Android Malware, from https://www.darkreading.com/threat-intelligence/downloaders-currently-the-most-prevalent-android-malware.

702. (2023), SecuriDropper: New Android Dropper-as-a-Service Bypasses Google's Defenses, from https://thehackernews.com/2023/11/securidropper-new-android-dropper-as.html.

703. (2020). GuLoader: A Popular New VB6 Downloader that Abuses Cloud Services, from https://www.proofpoint.com/us/threat-insight/post/guloader-popular-new-vb6-downloader-abuses-cloud-services.

704. (2020). Trickbot disrupted, from https://www.microsoft.com/en-us/security/blog/2020/10/12/trickbot-disrupted/.

705. Charlotte Hammond and Ole Villadsen. (2023). The Trickbot/Conti crypters: Where are they now?, from https://securityintelligence.com/x-force/trickbot-conti-crypters-where-are-they-now/.

706. Bill Toulas. (2024). Hackers abuse QEMU to covertly tunnel network traffic in cyberattacks, from https://www.bleepingcomputer.com/news/security/hackers-abuse-qemu-to-covertly-tunnel-network-traffic-in-cyberattacks/.

707. Tushar Subhra Dutta. (2023). Hackers Use New Set of Hacking Tools to Attack Organizations in U.S. from https://cybersecuritynews.com/hacking-tools-organizations/.

708. Brian Krebs. (2023). Who and What is Behind the Malware Proxy Service SocksEscort?, from https://krebsonsecurity.com/2023/07/who-and-what-is-behind-the-malware-proxy-service-socksescort/.

709. (2023). Hoax Alert: 'Seismic Waves Card' Viral Message On WhatsApp Cannot Hack Your Phones, from https://srilanka.factcrescendo.com/english/seismic-waves-card-viral-message-on-whatsapp-cannot-hack-your-phones/.

710. (2024). Don't Ignore Google's Critical Security Alerts: How to Protect Your Online Identity, from https://veepn.com/blog/google-critical-security-alert/.

711. (2023). From Registry With Love: Malware Registry Abuses, from https://www.splunk.com/en_us/blog/security/from-registry-with-love-malware-registry-abuses.html#:~:text=Fileless%20Execution%20Through%20Registry,fileless%20malware%20on%20victim%20systems.

712. (2023). AI-Generated Malware and How It's Changing Cybersecurity, from https://www.impactmybiz.com/blog/how-ai-generated-malware-is-changing-cybersecurity/.

713. How can emulation help you analyze malware?, from https://www.linkedin.com/advice/0/how-can-emulation-help-you-analyze-malware-pgqfe.

714. (2024). 'Coyote' ugly: Kaspersky unveils banking trojan targeting over 60 institutions, from https://www.kaspersky.com/about/press-releases/2024_coyote-ugly-kaspersky-unveils-banking-trojan-targeting-over-60-institutions.

715. Vlad Constantinescu. (2023). New Fileless Linux Malware PyLoose Targets Cloud Workloads for Cryptomining, from https://www.bitdefender.com/blog/hotforsecurity/new-fileless-linux-malware-pyloose-targets-cloud-workloads-for-cryptomining/.

Module 08: Sniffing

716. Kyle Lai. (2002). Change MAC Address on Win2K & XP, from https://seclists.org/pen-test/2002/Nov/25.

717. Christopher R. Russel. (2001). Penetration Testing with dsniff, from http://www.ouah.org/dsniffintr.htm

718. Telephone tapping or wiretapping, from https://en.wikipedia.org/wiki/Telephone_tapping.

719. Dashdorj Yamkhin, and Youjip Won. Modeling and Analysis of Wireless LAN Traffic, from http://www.dmclab.hanyang.ac.kr/files/publication/journals/international/200911_08.pdf.

720. Sakun. (2011). Overview of Layer 2 Switched Networks and Communication, from http://www.sakunsharma.in/2011/07/overview-layer-2-switched-networks-communication/.

721. R. Droms. (1997). Dynamic Host Configuration Protocol, from https://www.ietf.org/rfc/rfc2131.txt.

722. Yusuf Bhaiji. Understanding, Preventing, Defending Against Layer 2 Attacks, from https://www.sanog.org/resources/sanog15/sanog15-yusuf-l2-security.pdf.

723. Satya P Kumar Somayajula, Yella. Mahendra Reddy, and Hemanth Kuppili. (2011). A New Scheme to Check ARP Spoofing: Prevention of MAN-IN-THE-MIDDLE Attack, from http://www.ijcsit.com/docs/Volume%202/vol2issue4/ijcsit2011020420.pdf.

724. Yusuf Bhaiji. LAYER 2 ATTACKS & MITIGATION TECHNIQUES, from https://www.sanog.org/resources/sanog7/yusuf-L2-attack-mitigation.pdf.

725. Adam Barth, Juan Caballero, and Dawn Song. Secure content sniffing for Web browsers or How to stop papers from reviewing themselves, from https://www.adambarth.com/papers/2009/barth-caballero-song.pdf.

726. Undetectable sniffing on Ethernet, from https://www.askapache.com/hacking/sniffing-ethernet-undetected/.

727. Kirk Hausman, Diane Barrett, and Martin Weiss. (2003). Identifying Nonessential Services and Attacks > Attacks, from http://www.informit.com/articles/article.asp?p=98121&seqNum=2.

728. ARP cache poisoning /ARP spoofing, from https://su2.info/doc/arpspoof.php.

729. Address Resolution Protocol (ARP), from www.erg.abdn.ac.uk/users/gorry/course/inet-pages/arp.html.

730. Packages, from https://www.debian.org/distrib/packages.

731. The Hacker's Ethic, from http://web.textfiles.com/ezines/HWA/hwa-hn34.txt.

732. Alberto Ornaghi and Marco Valleri, Man in the middle attacks, from https://www.blackhat.com/presentations/bh-europe-03/bh-europe-03-valleri.pdf.
733. Tom Olzak, (2006), DNS Cache Poisoning: Definition and Prevention, from https://adventuresinsecurity.com/Papers/DNS_Cache_Poisoning.pdf.
734. Daiji Sanai, (2001), Detection of Promiscuous Nodes using ARP packets, from http://www.securityfriday.com/promiscuous_detection_01.pdf.
735. (2018), Catalyst 4500 Series Switch Cisco IOS Software Configuration Guide, 12.2(25)EW, from https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst4500/12-2/25ew/configuration/guide/conf/port_sec.html.
736. Raj Chandel, (2017), DHCP Penetration Testing, from https://www.hackingarticles.in/dhcp-penetration-testing/.
737. (2016), 7 Popular Layer 2 Attacks, from http://www.pearsonitcertification.com/articles/article.aspx?p=2491767.
738. (2019), VLAN hopping, from https://en.wikipedia.org/wiki/VLAN_hopping#Switch_spoofing.
739. VLAN Hopping, from https://networklessons.com/cisco/ccnp-switch/vlan-hopping.
740. Pam, (2018), VLAN Hopping: How to Prevent an Attack, from https://cybersecurity.att.com/blogs/security-essentials/vlan-hopping-and-mitigation.
741. (2018), Common Attack Types on Switches, from https://digitalfortressllc.wordpress.com/2018/03/22/common-attack-types-on-switches/.
742. Valter Popeskic, STP Layer 2 attack – Manipulating Spanning Tree Protocol settings, from https://howdoesinternetwork.com/2012/stp-attack.
743. (2019), Protecting Against an STP Attack, from https://www.ccexpert.us/configuration-mode/protecting-against-an-stp-attack.html.
744. (2011), BPDU Guard, BPDU Filter, Root Guard, Loop Guard & UDLD, from http://ericleehy.com/index.php/bpdu-guard-bpdu-filter-root-guard-loop-guard-udld/.
745. What is BPDU Guard and how to configure BPDU Guard in Cisco Switches, from http://www.omnisecu.com/cona-security/what-is-bpdu-guard-and-how-to-configure-bpdu-guard-in-cisco-switches.php.
746. (2007), Configuring Spanning Tree PortFast, BPDU Guard, BPDU Filter, UplinkFast, BackboneFast, and Loop Guard, from https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst4000/8-2glx/configuration/guide/stp_enha.html.
747. Jay Miah, (2017), Loop Guard Concept and Implementation, from http://www.jay-miah.co.uk/loop-guard-concept-and-implementation/.
748. (2018), Catalyst 6500 Release 12.2SY Software Configuration Guide, from https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst6500/ios/12-2SY/configuration/guide/sy_swcg/spanning_tree_features.html#27629.
749. (2019), Cisco Nexus 5000 Series NX-OS Software Configuration Guide, from https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus5000/sw/configuration/guide/cli/CLIConfigurationGuide/BasicEthernet.html.
750. (2021), Example: Configuring MAC Limiting, from https://www.juniper.net/documentation/us/en/software/junos/security-services/topics/topic-map/example-configuring-port-limiting.html.
751. Configuring DHCP Filtering, from https://docs.oracle.com/cd/E19859-01/820-3252-11/FP44ucgDHCPFiltering.html#50397227_76096.
752. DHCP Filter, from https://www.tp-link.com/us/configuration-guides/configuring_dhcp_filter/?configurationId=18223#_idTextAnchor009.
753. Simone Cataria, (2021), SAD DNS: A Revival of the DNS Cache Poisoning Attack, from https://www.internetx.com/en/news-detailview/sad-dns-a-revival-of-the-dns-cache-poisoning-attack.
754. Nick Sullivan and Marek Vavruša, (2020), SAD DNS Explained, from https://blog.cloudflare.com/sad-dns-explained.
755. Introduction : SAD DNS, from https://www.saddns.net.

Module 09: Social Engineering

756. Margaret Rouse, (2011), Spear Phishing, from https://www.techtarget.com/searchsecurity/definition/spear-phishing.
757. Terry Turner, Social Engineering – Can Organizations Win the Battle?, from http://www.infosecwriters.com/text_resources/pdf/Social_Engineering_Can_Organizations_Win.pdf.
758. Sharon Gaudin, Social Engineering: The Human Side Of Hacking, from http://www.crimo-research.org/library/Sharon2.htm.
759. Anti-Phishing Resources, from https://www.antiphishing.org/resources/.
760. (2007), Phishing and bogus emails: HM Revenue and Customs examples, from https://www.gov.uk/government/publications/phishing-and-bogus-emails-hm-revenue-and-customs-examples.

761. (2014). How to Protect Insiders from Social Engineering Threats, from https://learn.microsoft.com/en-us/previous-versions/tn-archive/cc875841(v=technet.10)?redirectedfrom=MSDN.
762. (2014). Security Threats, from https://learn.microsoft.com/en-us/previous-versions/tn-archive/cc723507(v=technet.10)?redirectedfrom=MSDN.
763. Melissa Guenther, (2001), Social Engineering, from http://www.iwar.org.uk/comsec/resources/sa-tools/Social-Engineering.pdf.
764. Bodok Frak, (2008), Gone Phishing, from http://web4.uwindsor.ca/units/its/insight/insight.nsf/babe0ebac149fc7b852567d700715d2a/d1b60efa750dfe21852573d7004cbbd8!OpenDocument.
765. Phishing: Examples & its prevention methods, from http://chowkamleeng.blogspot.com/2008/06/phishing-examples-its-prevention.html.
766. Gunter Ollmann, The Phishing Guide (Part 1), from http://www.technicalinfo.net/papers/Phishing.html.
767. Rachna Dhamija, (2006), Why phishing works, from https://dl.acm.org/doi/10.1145/1124772.1124861.
768. (2009), Social engineering, from https://www.techtarget.com/searchsecurity/definition/social-engineering.
769. The Social Engineering Framework, from https://www.social-engineer.org/framework/attack-vectors/impersonation/.
770. Smishing, vishing, and phishing... oh my!, from https://www.forensicaccountingservices.com/fraudvault/smishing-vishing-and-phishing/.
771. Jake Stroup, (2017), The Many Types of Identity Theft, from https://www.thebalance.com/the-8-types-of-identity-theft-1947176.
772. Clari Melo, (2014), Get to Know These Common Types of ID Theft, from https://www.igrad.com/articles/8-types-of-identity-theft.
773. (2015), The 10 Major Types of Identity Theft, from https://www.idtheftauthority.com/types-of-identity-theft/.
774. (2011), The 6 Types of Identity Theft, from https://www.mcafee.com/blogs/.
775. (2015), Identity Theft, from https://completeid.com/types-of-identity-theft/.
776. (2015), Social engineering techniques: Pretexting, diversion theft, phishing, from https://sgros-students.blogspot.com/2015/11/pretexting-diversion-theft-phishing.html.
777. (2020), Social engineering (security), from https://en.wikipedia.org/wiki/Social_engineering_(security)#Other_types.
778. Kevin Mitnick, What is social engineering?, from https://www.knowbe4.com/what-is-social-engineering/#1.
779. Chris Brunau, (2019), 5 Types of Social Engineering Attacks, from https://www.datto.com/blog/5-types-of-social-engineering-attacks.
780. Pierluigi Paganini, (2019), The Most Common Social Engineering Attacks, from https://www.infosecinstitute.com/resources/security-awareness/common-social-engineering-attacks/#gref.
781. The Social Engineering Framework, from https://www.social-engineer.org/framework/influencing-others/pretexting/successful-pretexting/.
782. Erdal Ozkaya and Yuri Diogenes, Cybersecurity - Attack and Defense Strategies, from https://learning.oreilly.com/library/view/cybersecurity-attack/9781788475297/5a6d16cf-6400-411e-bba2-ecbd10ad2d88.xhtml.
783. 2020 Cost of Insider Threats Global Report, from https://www.proofpoint.com/us/resources/threat-reports/cost-of-insider-threats.
784. George Mometos, (2017), The CISO's Guide to Managing Insider Threats, from https://securityintelligence.com/the-cisos-guide-to-managing-insider-threats/.
785. (2016), Managing Insider Threat, from https://www.ey.com/Publication/vwLUAssets/EY-managing-inside-threat/$FILE/EY-managing-inside-threat.pdf.
786. Facts + Statistics: Identity theft and cybercrime, from https://www.iii.org/fact-statistic/facts-statistics-identity-theft-and-cybercrime.
787. Linda Musthaler, (2008), 13 best practices for preventing and detecting insider threats, from https://www.networkworld.com/article/2280365/13-best-practices-for-preventing-and-detecting-insider-threats.html.
788. Insider Threat Prevention Best Practices, from https://www.netwrix.com/Insider_Threat_Prevention_Best_Practices.html.
789. Amanda Hicks, Angler Phishing: What is it?, from https://www.clearviewfcu.org/Learn/about-financial-wellness/Blog/Angler-Phishing-What-is-it#:~:text=Angler%20phishing%20is%20a%20new,personal%20information%20or%20account%20credentials.
790. Eva Velasquez, (2018), What Is Angler Phishing and How Can You Avoid It?, from https://www.experian.com/blogs/ask-experian/what-is-angler-phishing-and-how-can-you-avoid-it/.
791. Jomile Nakutavičiūtė, (2021), 12 Types of Social Engineering Attacks, from https://nordvpn.com/blog/social-engineering/.
792. Sherri Gordon, (2020), Catfishing and How It Relates to Cyberbullying, from https://www.verywellfamily.com/what-is-catfishing-460588.
793. What is Catfishing Online: Signs & How to Tell, from https://www.fortinet.com/resources/cyberglossary/catfishing.
794. (2021), What are Deepfakes? Are They a Security Threat?, from https://www.tessian.com/blog/what-are-deepfakes/.

795. Deepfake, from https://en.wikipedia.org/wiki/Deepfake.
796. (2021). Deepfake Fraud: Security Threats Behind Artificial Faces, from https://www.pandasecurity.com/en/mediacenter/technology/deepfake-fraud/.
797. Defining Insider Threats, from https://www.cisa.gov/topics/physical-security/insider-threat-mitigation/defining-insider-threats.
798. What Is an Insider Threat?, from https://www.fortinet.com/resources/cyberglossary/insider-threats.
799. What is Social Engineering?, From https://www.webroot.com/in/en/resources/tips-articles/what-is-social-engineering.
800. (2021). Social Engineering Countermeasures, from https://www.rangeforce.com/blog/social-engineering-countermeasures.
801. (2021). 10 Types of Social Engineering Attacks, from https://www.crowdstrike.com/cybersecurity-101/types-of-social-engineering-attacks/.
802. Abi Tyas Tunggal, (2022). What is Social Engineering? Definition and Protection Tips for 2022, from https://www.upguard.com/blog/social-engineering.
803. Vinugayathri Chinnasamy, (2020). 10 Ways Businesses Can Prevent Social Engineering Attacks, from https://www.indusface.com/blog/10-ways-businesses-can-prevent-social-engineering-attacks/.
804. Ways to Avoid Social Engineering Attacks, from https://www.kaspersky.co.in/resource-center/threats/how-to-avoid-social-engineering-attacks.
805. Social Engineering Countermeasures, from https://bitdefender.com/social-engineering-countermeasures/.
806. Protect Yourself Against Phishing Scams and Identity Theft, from https://umass.service-now.com/sp?id=kb_article_view&sysparm_article=KB0011051.
807. 5 Best Defenses Against Phishing Attacks, from https://www.egress.com/resources/cybersecurity-information/phishing/5-best-defenses-against-phishing-attacks.
808. David Bisson, (2021). 6 Common Phishing Attacks and How to Protect Against Them, from https://www.tripwire.com/state-of-security/security-awareness/6-common-phishing-attacks-and-how-to-protect-against-them/.
809. Juliana De Groot, (2022). Phishing Attack Prevention: How to Identify & Avoid Phishing Scams in 2022, from https://digitalguardian.com/blog/phishing-attack-prevention-how-identify-avoid-phishing-scams.
810. Paulius Ilevičius, (2020). Online bait and switch scams explained, from https://nordvpn.com/blog/what-is-bait-and-switch/.
811. Tomas Meskauskas, (2023). What kind of scam is "Microsoft Support Alert"?, from https://www.pcrisk.com/removal-guides/13878-microsoft-support-pop-up-scam.
812. Nathaniel Raymond, (2023). Resurgence of Linked In Smart Links Identified in Sizable Credential Phishing Campaign, from https://cofense.com/blog/linkedin-smart-links-credential-phishing-campaign/.
813. What is Clone Phishing?, from https://www.checkpoint.com/cyber-hub/threat-prevention/what-is-phishing/what-is-clone-phishing/#:~:text=A%20clone%20phishing%20attack%20is,using%20a%20fake%20tracking%20email.
814. Deriq Bernard, (2024). E-wallet phishing is latest social engineering method, from https://malaya.com.ph/news_special_feature/e-wallet-phishing-is-latest-social-engineering-method/.
815. Deepa Nagalingam and Megha Sasidhar, (2023). What is reverse tabnabbing and what can you do to stop it?, from https://securityintelligence.com/posts/what-is-reverse-tabnabbing-and-what-can-you-do-to-stop-it/.
816. Aranza Trevino, (2023). What is Search Engine Phishing?, from https://www.keepersecurity.com/blog/2023/04/12/what-is-search-engine-phishing/.
817. What is Consent Phishing? Identifying Third Party App Permission Attacks, from https://abnormalsecurity.com/glossary/consent-phishing.
818. (2022). Introduction of QR code attacks and countermeasures, from https://www.hkcert.org/blog/introduction-of-qr-code-attacks-and-countermeasures.
819. Qrljacking, from https://owasp.org/www-community/attacks/Qrljacking.
820. Hearty and Ricson E, (2024). QR TIGER Product Update: Clone QR Code Feature, from https://www.qrcode-tiger.com/clone-qr-code#QR_TIGERs_clone_QR_code_feature_How_does_it_work.

Module 10: Denial-of-Service

821. (2006). Denial of Service Attacks: Teardrop and Landÿÿ, from http://users.tkk.fi/~lhuovine/study/hacker98/dos.html.
822. (2006). CERT warns of networked denial of service attacks – Computerworld, from http://www.computerworld.com/action/pages.do?command=viewPage&pagePath=/404.
823. Stephen M. Specht and Ruby B. Lee, (2004). Distributed Denial of Service: Taxonomies of Attacks, Tools and Countermeasures, from http://palms.ee.princeton.edu/PALMSopen/DDoS%20Final%20PDCS%20Paper.pdf.
824. Craig A. Huegen, (2005). Denial of Service Attacks: "Smurfing", from http://www.pentics.net/denial-of-service/white-papers/smurf.cgi.

825. Remotely Triggered Black Hole Filtering in IP Version 6 for Cisco IOS, Cisco IOS XE, and Cisco IOS XR Software, from https://sec.cloudapps.cisco.com/security/center/resources/ipv6_remotely_triggered_black_hole.
826. Frank Kargl, Jörn Maier, Stefan Schlott, and Michael Weber, Protecting Web Servers from Distributed Denial of Service Attacks, from http://www10.org/cdrom/papers/409/.
827. (1997). Denial of Service Attacks, from https://insights.sei.cmu.edu/library/1997-tech-tip-denial-of-service-attacks/.
828. Denial of service, from https://searchsecurity.techtarget.com/definition/denial-of-service.
829. Vladimir Golubev, (2005). DoS attacks: crime without penalty, From http://www.crime-research.org/articles/1049/.
830. Gunter Ollmann, (2009). Botnet Communication Topologies, from http://www.technicalinfo.net/papers/PDF/WP_Botnet_Communications_Primer_(2009-06-04).pdf.
831. Renaud BIDOU, Fighting the Botnet Ecosystem, from http://www.iv2-technologies.com/FightingBotnetEcosystem.pdf.
832. Ping of death, from https://searchsecurity.techtarget.com/definition/ping-of-death.
833. Jason Anderson, (2001). An Analysis of Fragmentation Attacks, from http://www.ouah.org/fragma.html.
834. Mariusz Burdach, (2003). Hardening the TCP/IP stack to SYN attacks, from https://www.symantec.com/connect/articles/hardening-tcpip-stack-syn-attacks.
835. Deepak Singh Rana, Naveen Garg, and Sushil Kumar Chamoli, (2012). Citations: TCP SYN Flooding and IP Spoofing Attacks (Researchindex), from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.434.8352&rep=rep1&type=pdf.
836. Stephen Specht and Ruby Lee, (2003). Taxonomies of Distributed Denial of Service Networks, Attacks Tools, and Countermeasures, from https://www.princeton.edu/~rblee/DDoS%20Survey%20Paper_v7final.doc.
837. Gary C. Kessler, (2000). Defenses against Distributed Denial-Of-Service, from https://www.garykessler.net/library/ddos.html.
838. The Distributed Reflection DoS Attack, From https://www.grc.com/sn/sn-008.pdf.
839. Steve Gibson, (2002). Distributed Reflection Denial of Service Bandwidth Consumption, from https://homes.cs.washington.edu/~arvind/cs425/doc/drdos.pdf.
840. Haining Wang, Danlu Zhang, and Kang G. Shin, (2002). SYN Attack, from https://ieeexplore.ieee.org/document/1019404/.
841. Aaron Sullivan, 2001. An Audit of Active Directory Security, from https://www.symantec.com/connect/articles/audit-active-directory-security-part-2.
842. Denial of Service, from https://www.mycert.org.my/en/services/advisories/mycert/2017/main/detail/1277/index.html.
843. (2000). Denial of Service Attack in NetBIOS Services, from https://www.kb.cert.org/vuls/id/32650.
844. Wireless DoS, from https://www.cisco.com/c/en/us/td/docs/wireless/technology/wips/deployment/guide/WIPS_deployment_guide.html#pgfId-43390
845. Abhishek Singh, (2005). Demystifying Denial-Of-Service attacks, part one, from https://www.symantec.com/connect/articles/demystifying-denial-service-attacks-part-one.
846. Denial-of-service attack, from https://en.wikipedia.org/wiki/Denial-of-service_attack.
847. Kevin Poulsen, (2010). New Cyberattack Against WikiLeaks Was Weak, from https://www.wired.com/2010/11/wikileaks-attack/.
848. What is a DDoS Attack, from https://www.digitalattackmap.com/understanding-ddos/.
849. Glenn Carl and George Kesidis, (2009). Denial-of-Service Attack-Detection Techniques, https://www.evernote.com/shard/s9/note/b11a8c31-8651-4d74-acf9-1fb1b3c0f090/wishi/crazylazy#st=p&n=b11a8c31-8651-4d74-acf9-1fb1b3c0f090.
850. Distributed denial-of-service attack, from https://www.techtarget.com/searchsecurity/definition/distributed-denial-of-service-attack.
851. Taxonomies of Distributed Denial of Service Networks, Attacks, Tools, and Countermeasures, from http://www.princeton.edu/~rblee/DDoS%20Survey%20Paper_v7final.doc.
852. Glenn Carl, (2006). Denial-of-Service Attack-Detection Techniques, from https://www.computer.org/csdl/mags/ic/2006/01/w1082-abs.html.
853. Stephen M. Specht and Ruby B. Lee, (2003). Distributed Denial of Service: Taxonomies of Attacks, Tools and Countermeasures, from http://palms.ee.princeton.edu/PALMSopen/DDoS%20Final%20PDCS%20Paper.pdf.
854. Vijay C Uyyuru, Prateek Arora, and Terry Griffin, Denial of Service (DoS), from https://computerscience.engineering.unt.edu/.
855. Denial of Service (DoS), Distributed DoS (DDoS), from http://www.atiguide2000.com/security/index.php?act=view&aid=193.
856. (2007). Denial Of services [botnet] (DoS), from https://www.go4expert.com/articles/denial-services-botnet-dos-t3184/.
857. SYN Flood Attack, from https://www.cloudflare.com/learning/ddos/syn-flood-ddos-attack/.
858. Zobair Khan, (2015). Basics on DDos, from https://www.slideshare.net/kzobair/ddosbdnog.

859. Brian Prince, (2013), Multi-vector DDoS Attacks Grow in Sophistication, from http://www.securityweek.com/multi-vector-ddos-attacks-grow.

860. Dancho Danchev, (2009), Malware-infected WinRAR distributed through Google AdWords, from http://www.zdnet.com/article/malware-infected-winrar-distributed-through-google-adwords/.

861. Angad Singh, (2017), A hacker hijacked Chrome Extension to push Malware, from https://www.officialhacker.com/chrome-extension-to-push-malware/.

862. Stelian Pilci, (2015), Remove vikingwebscanner.com virus (Fake AdwCleaner), from https://malwaretips.com/blogs/vikingwebscanner-com-removal/.

863. (2016), Mobile Security Threats on The Rise as Hackers Can Launch DDoS Attacks on Their Mobile Phones, from https://www.radware.com/security/ddos-threats-attacks/cyber-attacks-in-the-palm-of-your-hand/.

864. Gurubaran S, (2018), Pulse Wave Heavy DDoS Attack to Take Down Multiple Protected Target Networks, from https://gbhackers.com/new-ddos-attack-pulse-wave/.

865. Zero-day DDoS Attack (0day DDoS attack), from https://ddos-guard.net/en/terminology/attack_type/zero-day-ddos-attack-0day-ddos-attack.

866. 35 Types of DDoS Attacks Explained, from https://mochahost.com/welcome-javapipe.

867. UDP Flood Attack, from https://www.cloudflare.com/learning/ddos/udp-flood-ddos-attack/.

868. UDP Based Amplification Attacks, from https://instasafe.com/udp-based-amplification-attacks/.

869. Sam Kottler, (2018), DDoS Incident Report, from https://github.blog/2018-03-01-ddos-incident-report/.

870. (2020), Rate Limiting, from https://en.wikipedia.org/wiki/Rate_limiting.

871. Shane Schick, (2019), TCP SACK Panic Flaw Could Compromise Production Linux Machines, from https://securityintelligence.com/news/tcp-sack-panic-flew-could-compromise-production-linux-machines/.

872. Glen Kosaka, (2019), How to Mitigate the SACK Panic DDoS Attack, from https://blog.neuvector.com/article/mitigate-sack-panic-ddos-attack.

873. (2019), TCP SACK PANIC - Kernel vulnerabilities CVE-2019-11477, CVE-2019-11478 & CVE-2019-11479, from https://access.redhat.com/security/vulnerabilities/tcpsack.

874. Carol Hildebrand, (2021), What Is a DDoS Extortion Attack?, from https://www.netscout.com/blog/what-ddos-extortion-attack.

875. What is a Ransom DDoS Attack?, from https://www.cloudflare.com/en-in/learning/ddos/ransom-ddos-attack/.

876. Extortion Attacks Are Back: How To Be Prepared, from https://www.home.neustar/blog/how-to-prepare-for-ddos-extortion-attacks.

877. Ritika Singh, (2021), What is a DDoS Extortion Attack and How do you Respond to it?, from https://securityboulevard.com/2021/07/what-is-a-ddos-extortion-attack-and-how-do-you-respond-to-it/.

878. Amir Dahan, (2021), Business as Usual for Azure Customers Despite 2.4 Tbps DDoS Attack, from https://azure.microsoft.com/en-us/blog/business-as-usual-for-azure-customers-despite-24-tbps-ddos-attack/.

879. Catalin Cimpanu, (2021), Microsoft Said It Mitigated a 2.4 Tbps DDoS Attack, from https://therecord.media/microsoft-said-it-mitigated-a-2-4-tbps-ddos-attack-the-largest-ever/.

880. Ionut Arghire, (2021), Microsoft Azure Hit by 2.4 Tbps DDoS Attack, from https://www.securityweek.com/microsoft-mitigates-24-tbps-ddos-attack-targeting-azure.

881. Denial of Service Attack: Definition, Examples, and Prevention, from https://hop.extrahop.com/resources/attacks/dos/.

882. Andreja Velimirovic, (2021), How to Prevent DDoS Attacks: 7 Tried-and-Tested Methods, from https://phoenixnap.com/blog/prevent-ddos-attacks.

883. Tim Keary, (2022), Dos vs DDoS Attacks: The Differences and How To Prevent Them, from https://www.comparitech.com/net-admin/dos-vs-ddos-attacks-differences-prevention.

884. NTP amplification DDoS attack, from https://www.cloudflare.com/learning/ddos/ntp-amplification-ddos-attack/.

885. Script ntp-monlist, from https://nmap.org/nsedoc/scripts/ntp-monlist.html.

886. Emil Kiner and Tim April, (2023), Google mitigated the largest DDoS attack to date, peaking above 398 million rps, from https://cloud.google.com/blog/products/identity-security/google-cloud-mitigated-largest-ddos-attack-peaking-above-398-million-rps.

Module 11: Session Hijacking

887. (2006), hunt(1) - Linux man page, from https://linux.die.net/man/1/hunt.

888. (2006), Web Application Attacks – Intro, from www.netprotect.ch/downloads/webguide.pdf.

889. Steps in Session Hijacking, from https://www.hackguide4u.com/2010/03/steps-in-session-hijacking.html.

890. Session Hijacking, from https://www.imperva.com/learn/application-security/session-hijacking/.

891. Adnan Anjum, Spoofing Vs Hijacking, from https://www.hackguide4u.com/2010/03/spoofing-vs-hijacking.html.
892. Lee Lawson, (2005), Session Hijacking Packet Analysis, from https://www.scribd.com/document/53979390/3d79.
893. Session hijacking attack, from https://owasp.org/www-community/attacks/Session_hijacking_attack.
894. Shray Kapoor, Session Hijacking Exploiting TCP, UDP and HTTP Sessions, from http://www.infosecwriters.com/text_resources/pdf/SKapoor_SessionHijacking.pdf.
895. David Endler, (2001), Brute-Force Exploitation of Web Application Session IDs, from https://www.cgisecurity.com/lib/SessionIDs.pdf.
896. Robert Auger, Credential and Session Prediction, from http://projects.webappsec.org/w/page/13246918/Credential%20and%20Session%20Prediction.
897. Trojan horse, from https://www.techtarget.com/searchsecurity/definition/Trojan-horse.
898. (2008), Prevention from Session Hijacking, from http://hydtechie.blogspot.com/2008/08/prevention-from-session-hijacking.html.
899. Harsh Kevadia, (2013), Session Hijacking, from https://www.slideshare.net/slideshow/session-hijacking-by-harsh-kovadiya/26648672.
900. (2009), Man-in-the-middle attack, from https://owasp.org/www-community/attacks/Man-in-the-middle_attack.
901. (2009), Man-in-the-browser attack, from https://owasp.org/www-community/attacks/Man-in-the-browser_attack.
902. Session Hijacking: A Primer, from http://www.cs.binghamton.edu/~steflik/cs455/sessionhijacking.htm.
903. (2009), Session fixation, from https://www.owasp.org/index.php?title=Session_fixation&setlang=es.
904. (2001), CERT® Advisory CA-2001-09 Statistical Weaknesses in TCP/IP Initial Sequence Numbers, from http://www.cert.org/advisories/CA-2001-09.html.
905. (2009), Connection Security and IPsec, from https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc771593(v=ws.10).
906. Laurent Joncheray, (1995), A Simple Active Attack Against TCP, from http://www.blyx.com/public/docs/security/tcp_attack.pdf.
907. (2009), IPSec Architecture, from https://learn.microsoft.com/en-us/previous-versions/bb726946(v=technet.10)?redirectedfrom=MSDN.
908. Mohit Kumar, (2012), CRIME: New SSL/TLS attack for Hijacking HTTPS Sessions, from https://thehackernews.com/2012/09/crime-new-ssltls-attack-for-hijacking.html.
909. Dennis Fisher, (2012), Crime Attack Uses Compression Ratio of TLS Requests as Side Channel to Hijack Secure Sessions, from https://threatpost.com/crime-attack-uses-compression-ratio-tls-requests-side-channel-hijack-secure-sessions-091312/77006/.
910. (2013), BEAST vs. CRIME Attack, from https://www.infosecinstitute.com/resources/hacking/beast-vs-crime-attack/#gref.
911. Amrita Mitra, (2017), What is the CRIME Attack?, from https://www.thesecuritybuddy.com/vulnerabilities/what-is-crime-attack/.
912. Agathoklis Prodromou, (2019), TLS/SSL Explained – Examples of a TLS Vulnerability and Attack, Final Part, from https://www.acunetix.com/blog/articles/tls-vulnerabilities-attacks-final-part/.
913. Nick Lewis, How can an HTTPS session get hijacked with the Forbidden attack?, from https://techtarget.com/searchsecurity/answers.
914. Dan Goodin, (2016), "Forbidden attack" makes dozens of HTTPS Visa sites vulnerable to tampering, from https://arstechnica.com/information-technology/2016/05/faulty-https-settings-leave-dozens-of-visa-sites-vulnerable-to-forgery-attacks/.
915. Mark Wyciślik-Wilson, (2016), Decade-old 'Forbidden attack' vulnerability affects HTTPS Visa sites, from https://betanews.com/2016/05/26/https-forbidden-attack/.
916. Pierluigi Paganini, (2016), Dozens of VISA HTTPS-protected sites vulnerable to Forbidden attack, from http://securityaffairs.co/wordpress/47724/breaking-news/forbidden-attack.html.
917. (2017), What is HSTS and How Do I Implement It?, from https://www.globalsign.com/en/blog/what-is-hsts-and-how-do-i-use-it/.
918. Justin Johnson, (2016), What is HSTS?, from https://blog.stackpath.com/glossary/hsts/.
919. Sjoerd Langkemper, (2017), Prevent session hijacking with token binding, from https://www.sjoerdlangkemper.nl/2017/07/05/prevent-session-hijacking-with-token-binding/.
920. Bjørn Johansen, (2017), What is a Session Donation Attack?, from https://bjornjohansen.no/wordpress-session-donation-attack.
921. Alek Amrani, Session Donation, from https://www.defcon.org/images/defcon-17/dc-17-presentations/defcon-17-alek_amrani-session_donation.pdf.
922. Ramesh Lingappa, (2018), What is Session Hijacking and How You can Stop it, from https://www.freecodecamp.org/news/session-hijacking-and-how-to-stop-it-731e3683d1ac/.
923. Man-in-the-Middle (MITM) Attacks, from https://www.rapid7.com/fundamentals/man-in-the-middle-attacks/.

924. (2019). Man-in-the-Middle Attacks and How to Avoid them, from https://www.invicti.com/blog/web-security/man-in-the-middle-attack-how-avoid/.

925. Bojana Dobran, (2019), What are Man in the Middle Attacks & How to Prevent MITM Attack with Examples, from https://phoenixnap.com/blog/man-in-the-middle-attacks-prevention.

926. Santosh Subramanya, (2021), How PetitPotam hijacks the Windows API, and what you can do about it, from https://news.sophos.com/en-us/2021/08/25/how-petitpotam-hijacks-the-windows-api-and-what-you-can-do-about-it/.

927. Ran Harel, (2021), Detecting and Mitigating the PetitPotam Attack on Windows Domains, from https://www.semperis.com/blog/petitpotam-attack-on-windows-domains/.

928. (2021), PetitPotam – NTLM Relay to AD CS, from https://pentestlab.blog/2021/09/14/petitpotam-ntlm-relay-to-ad-cs/.

929. (2021), Hacking Domain Admin 6 ways to Sunday | PetitPotam, DCSync & Golden Tickets, from https://www.youtube.com/watch?v=ZM6-SakrOtB.

930. Firefox DNS-over-HTTPS, from https://support.mozilla.org/en-US/kb/firefox-dns-over-https.

931. Brien Posey, (2020), DNS over HTTPS (DoH), from https://www.techtarget.com/searchsecurity/definition/DNS-over-HTTPS-DoH.

932. DNS over TLS vs. DNS over HTTPS | Secure DNS, from https://www.cloudflare.com/en-in/learning/dns/dns-over-tls/.

933. Password Manager, from https://www.malwarebytes.com/what-is-password-manager.

934. What is a password manager?, from https://www.zoho.com/vault/educational-content/what-is-a-password-manager.html.

935. (2020), The Ultimate Guide to Man in the Middle (MITM) Attacks and How to Prevent them, from https://doubleoctopus.com/blog/enterprise-security/the-ultimate-guide-to-man-in-the-middle-mitm-attacks-and-how-to-prevent-them/

936. Kapil Raina, (2021), Zero Trust Security Explained: Principles of the Zero Trust Model, from https://www.crowdstrike.com/cybersecurity-101/zero-trust-security/.

937. Zero Trust Security Model — What is Zero Trust?, from https://www.akamai.com/resources/zero-trust-security-model?gclid=Cj0KCQjA9OiPBhCOARIsAI0y718XijpUcXrdreIfMNSwthwzZF9M92nDOto_0EU4Q1SneLsrYEaRLPIaAbyuEALw_wcB&utm_source=google&utm_medium=cpc&utm_campaign=F-MC-52610&utm_term=zero%20trust&utm_content=India&ef_id=Cj0KCQiA9OiPBhCOARIsAI0y718XijpUcXrdreIfMNSwthwzZF9M92nDOto_0EU4Q1SneLsrYEaRLPIaAbyuEALw_wcB:G:s

938. Jennifer, (2022), Session Hijacking Tutorial for Beginner Developers, from https://pinkhatcode.com/2022/02/27/session-hijacking-tutorial-for-beginner-developers/.

939. (2022), What is Session Hijacking?, from https://intellipaat.com/blog/what-is-session-hijacking/.

940. Tomasz Andrzej Nidecki, (2020), Session Hijacking and Other Session Attacks, from https://www.acunetix.com/blog/web-security-zone/session-hijacking/.

941. Zbigniew Banach, (2019), What is Session Hijacking: Your Quick Guide to Session Hijacking Attacks, from https://www.invicti.com/blog/web-security/session-hijacking/.

942. Anas Baig, (2021), What is Session Hijacking and How Do You Prevent It?, from https://www.globalsign.com/en/blog/session-hijacking-and-how-to-prevent-it.

Module 12: Evading IDS, Firewalls, and Honeypots

943. (2006), Measuring Security Threats with Honeypot Technology, from http://www.honeynet.org/papers/individual/sane-2004.pdf.

944. (2006), SecurityFocus: Honeytokens -The Other Honeypot, from https://www.symantec.com/connect/articles/honeytokens-other-honeypot.

945. (2006), Hardware Firewalls, from http://cybercoyote.org/security/hardware.shtml.

946. (2006), Intrusion Detection System (IDS) Evasion, from http://complianceandprivacy.com/WhitePapers/iDefense-IDS-Evasion/iDefense_IDSEvasion_20060510.pdf.

947. Intrusion detection system evasion techniques, from https://en.wikipedia.org/wiki/Intrusion_detection_system_evasion_techniques.

948. Tarek S.Sobh, (2006), Wired and wireless intrusion detection system: Classifications, good characteristics and state-of-the-art, from https://www.sciencedirect.com/science/article/pii/S092054890500098X.

949. (1998), Defeating Sniffers and Intrusion Detection Systems, from http://www.phrack.org/issues.html?issue=54&id=10.

950. Adam Gowdiak, (2003), Techniques used for bypassing firewall systems, from https://www.terena.org/activities/tf-csirt/meeting9/gowdiak-bypassing-firewalls.pdf.

951. IT Infrastructure Security Plan, from https://www.sciencedirect.com/science/article/pii/B9781597490887500098.

952. What is a firewall?, from https://kb.iu.edu/data/aoru.html.

953. Wassim El-Hajj, Hazem Hajj, Zouheir Trabelsi, and Fadi Aloul, (2010), Updating snort with a customized controller to thwart port scanning, from http://www.aloul.net/Papers/faloul_scn10.pdf.

954. Satyajit, What is HoneyPot?, from https://www.securityhunk.in/2010/06/what-is-honeypot.html.
955. Ashley Poland, (2017), How to Set Up a Honey Pot, from http://www.ehow.com/how_5245821_set-up-honey-pot.html.
956. Martin Roesch, Writing Snort Rules, from https://paginas.fe.up.pt/~mgi98020/pgr/writing_snort_rules.htm.
957. Intrusion detection system evasion techniques, from https://en.wikipedia.org/w/index.php?title=Intrusion_detection_system_evasion_techniques&oldid=311670246.
958. Sumit Siddharth, (2005), Evading NIDS, revisited, from https://community.broadcom.com/home.
959. How To Access Blocked / Bypass Blocked Websites, from http://www.computingunleashed.com/how-to-access-blocked.html.
960. How do I use a Proxy Server?, from https://whatismyipaddress.com/using-proxies.
961. Steve G. Belovich, Firewall Fairytales, from http://www.iqmtm.com/PDF_presentations/ID_Firewall_Fairytales_June2010-1.pdf.
962. Vern Paxson and Mark Handley, Defending Against Network IDS Evasion, from http://www.raid-symposium.org/raid99/PAPERS/Paxson.pdf.
963. (1996), Phrack Magazine Volume Seven, Issue Forty-Nine File 06 of ..., from http://phrack.org/issues/49/6.html.
964. Frank Wiles, Quick-Tip: SSH Tunneling Made Easy, from https://www.revsys.com/writings/quicktips/ssh-tunnel.html.
965. Firewall Basics, from http://www.unixgeeks.org/security/newbie/security/firewall.html.
966. Tony Bradley, Free Intrusion Detection (IDS) and Prevention (IPS) Software, from https://www.lifewire.com/ids-and-prevention-ips-software-2487316.
967. Bypassing Firewalls, from https://flylib.com/books/en/3.500.1.95/1/.
968. Vangie Beal, Intrusion detection system, from https://www.webopedia.com/TERM/I/intrusion_detection_system.html.
969. IDS: Re: Polymorphic Shellcode detection, from https://seclists.org/focus-ids/2003/May/22.
970. Circuit-Level Gateway, from http://www.softheap.com/internet/circuit-level-gateway.html.
971. Vicomsoft Firewall Q&A, from http://www.vicomsoft.com/knowledge/reference/firewalls1.html.
972. J. Christian Smith, (2000), Covert Shells - introduction, from http://www.gray-world.net/papers/covertshells.txt.
973. Peter Kieltyka, ICMP Shell, from http://icmpshell.sourceforge.net/.
974. Mark Burnett, (2001), Running Snort on IIS Web Servers Part 2: Advanced Techniques, from http://www.securityfocus.com/infocus/1316.
975. Mark Burnett, (2001), Running Snort on IIS Web Servers Part 2: Advanced Techniques, from https://www.symantec.com/connect/articles/running-snort-part-2.
976. Niels Provos, (2003), A Virtual Honeypot Framework, from http://www.citi.umich.edu/techreports/reports/citi-tr-03-1.pdf.
977. Brittany Day, (2003), Network Intrusion Detection Using Snort, from https://linoxsecurity.com/features/features/network-intrusion-detection-using-snort.
978. Paul Innella and Oba McMillan, (2001), An Introduction to IDS, from https://www.symantec.com/connect/articles/introduction-ids.
979. Ricky M. Magalhaes, (2003), Host-Based IDS vs Network-Based IDS, from http://techgenix.com/hids_vs_nids_part1/.
980. Paul Innella, (2001), The Evolution of Intrusion Detection Systems, from https://www.symantec.com/connect/articles/evolution-intrusion-detection-systems.
981. ntsecurity.nu - ack tunneling, from http://ntsecurity.nu/papers/acktunneling/.
982. SecuriTeam™ - ACK Tunneling Trojans, from https://www.beyondsecurity.com/.
983. Mike, Firewalk, from www.blackhat.com/presentations/bh-usa-99/Route/bh-us-99-schiffman.ppt.
984. Huynh Phi Long, Chapter 06: Network Security Using Cisco IOS IPS (Part02), from http://ciscodocuments.blogspot.in/2011/05/chapter-06-network-security-using-cisco_25.html.
985. About firewalls, from https://kb.iu.edu/d/aoru.
986. Pei Guo, How to bypass a firewall, from https://cosec.bit.uni-bonn.de/fileadmin/user_upload/teaching/06ws/06ws-computer-security/49692243.pdf.
987. Frank Wiles, Quick-Tip: SSH Tunneling made easy, from https://www.revsys.com/writings/quicktips/ssh-tunnel.html.
988. Reto E. Haeni, (1997), Firewall Penetration Testing, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.42.1117&rep=rep1&type=pdf.
989. Thomas H. Ptacek and Timothy N. Newsham, (1998), Insertion, Evasion and Denial of Service: Eluding Network Intrusion Detection, from https://insecure.org/stf/secnet_ids/secnet_ids.pdf.
990. David D. Rude II and Jayson Joan, Intrusion Detection System (IDS) Evasion, from https://complianceandprivacy.com/WhitePapers/iDefense-IDS-Evasion/PPT_iDefense_IDS_Evasion_20060505.pdf.
991. Eric Peter and Todd Schiller, (2008), A Practical Guide to Honeypots, from https://www.cse.wustl.edu/~jain/cse571-09/ftp/honey/index.html#sec1.4.

992. Krishna Prasad P, (2017), Capturing Attacks on IoT Devices with a multi-purpose IoT Honeypot, from https://security.cse.iitk.ac.in/sites/default/files/15111021.pdf.

993. Nishit Majithia, (2017), Honey-System: Design, Implementation, & Attack Analysis, from https://security.cse.iitk.ac.in/sites/default/files/15111024.pdf.

994. (2015), Bypassing Modern XSS WAF Filters, from http://rajjee1337.blogspot.in/2015/09/bypassing-modern-xss-waf-filters.html.

995. (2006), Intrusion Detection System (IDS) Evasion, An iDefense Security Report, from http://www.complianceandprivacy.com/WhitePapers/iDefense-IDS-Evasion/iDefense_IDSEvasion_20060510.pdf.

996. Intrusion detection system evasion techniques, from https://en.wikipedia.org/wiki/Intrusion_detection_system_evasion_techniques.

997. Kyöstilä, Tomi, (2014), The effectiveness of evasion techniques against intrusion prevention systems, from https://aaltodoc.aalto.fi/bitstream/handle/123456789/13589/master_Ky%c3%b6stil%c3%a4_Tomi_2014.pdf?sequence=1&isAllowed=y.

998. Laurent Oudot, Thorsten Holz, (2004), Defeating Honeypots: Network Issues, Part 1, from https://www.symantec.com/connect/articles/defeating-honeypots-network-issues-part-1.

999. Laurent Oudot, Thorsten Holz, (2004), Defeating Honeypots: Network Issues, Part 2, from https://www.symantec.com/connect/articles/defeating-honeypots-network-issues-part-2.

1000. Owen Watson, (2016), Countering Attack Deception Techniques, from http://slideplayer.com/slide/4495884/.

1001. Jon Oberheide and Manish Karir, Honeyd Detection via Packet Fragmentation, from https://www.merit.edu/wp-content/uploads/2016/01/Honeyd_Detection.pdf.

1002. Examining Different Types of Intrusion Detection Systems, from https://www.dummies.com/article/home-auto-hobbies/home-improvement-appliances/safety-security/examining-different-types-of-intrusion-detection-systems-200408/.

1003. Polymorphic virus, from https://www.trendmicro.com/vinfo/us/security/definition/Polymorphic-virus.

1004. (2018), Network Design: Firewall, IDS/IPS, from https://www.infosecinstitute.com/resources/network-security-101/network-design-firewall-idsips/#gref.

1005. What is an Intrusion Prevention System?, from https://www.paloaltonetworks.com/cyberpedia/what-is-an-intrusion-prevention-system-ips.

1006. Honeypot (computing), from https://en.wikipedia.org/wiki/Honeypot_(computing).

1007. Roger Galobardes, (2018), Learn how Easy is to Bypass Firewalls using DNS Tunneling (and also how to Block it), from https://medium.com/@rogergalo/learn-how-easy-is-to-bypass-firewalls-using-dns-tunneling-and-also-how-to-block-it-3ed652f4a000.

1008. (2019), DNS Tunneling Techniques in Cyberattacks, from https://www.andreafortuna.org/2019/01/16/dns-tunneling-techniques-in-cyberattacks/.

1009. Ron Lifinski, (2018), How Hackers Use DNS Tunneling to Own Your Network, from https://www.cynet.com/attack-techniques-hands-on/how-hackers-use-dns-tunneling-to-own-your-network/.

1010. Vickie Li, (2020), Intro to Malware Detection using YARA, from https://infosecwriteups.com/intro-to-malware-detection-using-yara-eacab8373cf4.

1011. Welcome to YARA's documentation !, from https://yara.readthedocs.io/en/stable/.

1012. Cedric Pernet, (2021), How to Write YARA Rules For Improving Your Security and Malware Detection, from https://www.techrepublic.com/article/how-to-write-yara-rules-for-improving-your-security-and-malware-detection/.

1013. Neil Fox, (2021), YARA Rules Guide: Learning this Malware Research Tool, from https://www.varonis.com/blog/yara-rules.

1014. Using YARA for Malware Detection, from https://www.cisa.gov/uscert/sites/default/files/FactSheets/NCCIC%20ICS_FactSheet_YARA_S508C.pdf.

1015. Pieter Arntz, (2017), Explained: YARA Rules, from https://www.threatdown.com/blog/explained-yara-rules/.

1016. Z. Oualid, (2021), The Most Powerful WAF Evasion Technique?, from https://www.getsecureworld.com/blog/the-most-powerful-waf-evasion-techniques/.

1017. Josh Berry, (2017), Bypass WAF, from https://portswigger.net/bappstore/ae2611da3bbc4687953a1f4ba6a4e04c.

1018. X-Forwarded-For HTTP Header Security Bypass, from https://www.acunetix.com/vulnerabilities/web/x-forwarded-for-http-header-security-bypass/.

1019. (2020), What is HTML smuggling?, from https://secureteam.co.uk/2020/08/25/what-is-html-smuggling/.

1020. (2021), HTML Smuggling Surges: Highly Evasive Loader Technique Increasingly Used in Banking Malware, Targeted Attacks, from https://www.microsoft.com/security/blog/2021/11/11/html-smuggling-surges-highly-evasive-loader-technique-increasingly-used-in-banking-malware-targeted-attacks/.

1021. (2021), HTML Smuggling – A Novel Malware Deploying Technique, from https://cisomag.com/.

1022. Pedro Tavares, (2021), How Criminals Are Using Windows Background Intelligent Transfer Service, from https://www.infosecinstitute.com/resources/malware-analysis/how-criminals-are-using-windows-background-intelligent-transfer-service/.

1023. Matt Mills, (2021), Windows Function Allows You to Bypass the Firewall and Control Any PC, from https://itigic.com/windows-function-allows-to-bypass-firewall-control-any-pc.

1024. David Via and Scott Runnels, (2021), Back in a Bit: Attacker Use of the Windows Background Intelligent Transfer Service, from https://cloud.google.com/blog/topics/threat-intelligence/attacker-use-of-windows-background-intelligent-transfer-service/.

1025. Ravie Lakshmanan, (2021), Hackers Using a Windows OS Feature to Evade Firewall and Gain Persistence, from https://thehackernews.com/2021/04/hackers-using-windows-os-feature-to.html.

1026. (2020), NAC Bypass Cheatsheet, from https://redteam.coffee/woot/nac-bypass-cheatsheet.

1027. Michael Schneider, (2019), Bypassing NAC a Handy How-to Guide, from https://www.scip.ch/en/?labs.20190207.

1028. Andrew Herd, (2021), NAC Bypass Attacks, from https://hack.technicherder.com/nac-bypass-attacks/.

1029. (2021), VLAN Hopping, from https://en.wikipedia.org/wiki/VLAN_hopping.

1030. (2018), Ghostwriting for Antivirus Evasion in 2018, from https://www.digitalforensicstips.com/2018/01/ghostwriting-for-antivirus-evasion-in.html.

1031. Will Dormann, (2016), Bypassing Application Whitelisting, from https://insights.sei.cmu.edu/blog/bypassing-application-whitelisting/.

1032. Andrew Black, (2018), Application Whitelist Bypass, from https://attackiq.com/2018/05/21/application-whitelist-bypass/.

1033. Maria Korolov, (2019), 6 Ways Malware Can Bypass Endpoint Protection, from https://www.csoonline.com/article/3403860/6-ways-malware-can-bypass-endpoint-protection.html.

1034. Robert Roohparvar, (2019), 05 Ways Malware Can Bypass Endpoint Protection, from https://www.infoguardsecurity.com/05-ways-malware-can-bypass-endpoint-protection/.

1035. (2020), Bypass Endpoint With XLM Weaponization, from https://0xsp.com/offensive/red-ops-techniques/bypass-endpoint-with-xlm-weaponization.

1036. Stan Hegt, (2018), Old school: Evil Excel 4.0 Macros (XLM), from https://outflank.nl/blog/2018/10/06/old-school-evil-excel-4-0-macros-xlm/.

1037. (2021), Weaponizing XLM 4.0 Macros, from https://gitbook.seguranca-informatica.pt/untitled/phishing-with-office/pwning-xlm-4.0-macros-+-c2.

1038. (2021), Working with Excel 4.0 Macros, from https://support.microsoft.com/en-us/office/working-with-excel-4-0-macros-ba8924d4-e157-4bb2-8d76-2c07ff02e0b8.

1039. Noora Hyvärinen, (2019), Dechaining Macros and Evading EDR, from https://blog.f-secure.com/dechaining-macros-and-evading-edr/.

1040. (2019), Bypassing Parent Child / Ancestry Detections, from https://github.com/blackd3r/OSCP-Cheatsheets/blob/master/offensive-security/initial-access/phishing-with-ms-office/bypassing-malicious-macro-detections-by-defeating-child-parent-process-relationships.md.

1041. (2021), Bypassing Parent Child / Ancestry Detections, from https://www.ired.team/offensive-security/initial-access/phishing-with-ms-office/bypassing-malicious-macro-detections-by-defeating-child-parent-process-relationships.

1042. (2021), Endpoint Detection and Response: How Hackers Have Evolved, from https://www.optiv.com/insights/source-zero/blog/endpoint-detection-and-response-how-hackers-have-evolved.

1043. Hoang Bui, (2019), Bypass EDR's Memory Protection, Introduction to Hooking, from https://medium.com/@fsx30/bypass-edrs-memory-protection-introduction-to-hooking-2efb21acffd6.

1044. Brendan Ortiz, (2021), Classic API Unhooking To Bypass EDR Solutions, from https://depthsecurity.com/blog/classic-api-unhooking-to-bypass-edr-solutions.

1045. (2019), AV Bypass with Metasploit Templates and Custom Binaries, from https://www.ired.team/offensive-security/defense-evasion/av-bypass-with-metasploit-templates.

1046. Symantec Endpoint Protection Bypass + Meterpreter Pivoting, from https://cyberstruggle.org/symantec-end-point-protection-bypass-meterpreter-pivoting/.

1047. Zulon, (2020), Bypassing Symantec Endpoint Protection for Fun & Profit (Defence Evasion), from https://cognosec.com/bypassing-symantec-endpoint-protection-for-fun-profit-defense-evasion/.

1048. Scott Goetzinger, (2021), Bypassing Defenses: Symantec Endpoint Protection, from https://www.whitepaksecurity.com/blog/bypassing-defenses-symantec-endpoint-protection/.

1049. (2021), Falcon OverWatch Hunts Down Adversaries Where They Hide, from https://www.crowdstrike.com/blog/four-popular-defensive-evasion-techniques-in-2021/.

1050. Evasion Use Case Chapter 1: Introduction, from https://community.exabeam.com/s/article/Evasion-Use-Case-Chapter-1-Introduction.

1051. Samuel West, (2019), What is an Evasion Technique?, from https://www.libraesva.com/what-is-an-evasion-technique/.

1052. Alexander S. Gillis, Domain generation algorithm (DGA), from https://www.techtarget.com/searchsecurity/definition/domain-generation-algorithm-DGA.

1053. (2022), Dynamic Resolution: Domain Generation Algorithms, from https://attack.mitre.org/techniques/T1568/002/.

1054. Livia Gyongyosi, (2023), What Is Domain Generation Algorithm? Definition and Role in Malware Attacks, from https://heimdalsecurity.com/blog/what-is-domain-generation-algorithm-definition-and-role-in-malware-attacks/.

1055. (2023), Next Generation Firewall 7.0 Installation Guide, from https://help.forcepoint.com/ngfw/en-us/7.0.0/install/cov_ex-1/pg/GUID-6D2EFA89-C1C8-42F4-8A30-4976CC3E9CBF.html.

1056. Prerequisites, from https://www.manageengine.com/products/firewall/help/installation/firewall-pre.html.

1057. Our Favorite XSS Filters/IDS and how to Attack Them, from https://www.blackhat.com/presentations/bh-usa-09/VELANAVA/BHUSA09-VelaNava-FavoritXSS-SLIDES.pdf

1058. XSS Filter Evasion Cheat Sheet, from https://cheatsheetseries.owasp.org/cheatsheets/XSS_Filter_Evasion_Cheat_Sheet.html.

1059. Alon Leviev, (2023), The Pool Party You Will Never Forget: New Process Injection Techniques Using Windows Thread Pools, from https://www.safebreach.com/blog/process-injection-using-windows-thread-pools/.

1060. (2023), New PoolParty Process Injection Techniques Outsmart Top EDR Solutions, from https://thehackernews.com/2023/12/new-poolparty-process-injection.html.

1061. Lindsay Von Tish, (2023), What the Vuln: EDR Bypass with LOLBins, from https://bishopfox.com/blog/edr-bypass-with-lolbins.

1062. Mario Lobo, (2023), EDR Evasion: How Hackers Got Past Endpoint Defenses, from https://lumu.io/blog/edr-evasion/.

1063. (2022), System Binary Proxy Execution: Control Panel, from https://attack.mitre.org/techniques/T1218/002/.

1064. (2022), Control Panel, from https://dmcxblue.gitbook.io/red-team-notes-2-0/red-team-techniques/defense-evasion/t1218-signed-binary-proxy-execution/untitled-9.

1065. Eran Shimony And Omer Tsarfati, (2023), Chatting Our Way Into Creating a Polymorphic Malware, from https://www.cyberark.com/resources/threat-research-blog/chatting-our-way-into-creating-a-polymorphic-malware.

1066. Shweta Sharma, (2023), ChatGPT creates mutating malware that evades detection by EDR, from https://www.csoonline.com/article/575487/chatgpt-creates-mutating-malware-that-evades-detection-by-edr.html.

1067. Hershit Rajpal, (2022), A Detailed Guide on AMSI Bypass, from https://www.hackingarticles.in/a-detailed-guide-on-amsi-bypass/.

1068. Batuhan Sancak, (2023), Amsi Overview and Bypass Methods, from https://medium.com/@rullk3d/amsi-overview-and-bypass-methods-76b9d5896eb5.

1069. (2022), A blueprint for evading industry leading endpoint protection in 2022, from https://vanmieghem.io/blueprint-for-evading-edr-in-2022/.

1070. (2022), Disabling Event Tracing For Windows (ETW), from https://unprotect.it/technique/disabling-event-tracing-for-windows-etw/.

1071. Ahmed Eissa, (2021), Honeypots Types, Technologies, Detection Techniques, and Tools , from https://www.linkedin.com/pulse/honeypots-types-technologies-detection-techniques-tools-ahmed-eissa#:~:text=Honeypots%20Detection%20Techniques,services%20running%20on%20the%20system.

1072. Detecting Honeypots, from https://ltux.nl/mirror/honeypot/final/ch09lev1sec1.html.

Module 13: Hacking Web Servers

1073. Web Parameter Tampering, from https://owasp.org/www-community/attacks/Web_Parameter_Tampering.

1074. Meier, John D. (2010), Configuring and organizing server security information, from http://www.freepatentsonline.com/7712137.html.

1075. Internet Security, from https://en.wikipedia.org/wiki/Internet_security.

1076. Securing applications, from https://www.slideshare.net/slideshow/application-security-1831714/1831714.

1077. Robert Auger, (2009), Server Misconfiguration, from http://projects.webappsec.org/w/page/13246959/Server%20Misconfiguration.

1078. (2009), Cache Poisoning, from https://owasp.org/www-community/attacks/Cache_Poisoning.

1079. Improving Web Application Security: Threats and Countermeasures, from https://learn.microsoft.com/en-us/previous-versions/msp-n-p/ff649874(v=pandp.10)?redirectedfrom=MSDN.

1080. Rick Rosato, (2014), Best Practices for Applying Service Packs, Hotfixes and Security Patches, from https://learn.microsoft.com/en-us/previous-versions/tn-archive/cc750077(v=technet.10)?redirectedfrom=MSDN.

1081. J.D. Meier, Alex Mackman, Michael Dunner, Srinath Vasireddy, Ray Escamilla, and Anandha Murukan, (2010), Securing Your Web Server, from https://learn.microsoft.com/en-us/previous-versions/msp-n-p/ff648653(v=pandp.10)?redirectedfrom=MSDN.

1082. Web Server Security and Database Server Security, from https://www.acunetix.com/websitesecurity/webserver-security/.

1083. Windows IIS Server hardening checklist, from https://www.techtarget.com/searchsecurity/feature/Windows-IIS-server-hardening-checklist.

1084. IIS Web Server Security, from https://www.acunetix.com/websitesecurity/iis-security/.

1085. Checklist: Securing Your Web Server, from https://learn.microsoft.com/en-us/previous-versions/msp-n-p/ff648198(v=pandp.10)?redirectedfrom=MSDN.

1086. HTTP Response Splitting, Web Cache Poisoning Attacks, and Related Topics, from http://www.ouah.org/whitepaper_httpresponse.pdf.

1087. Hacking Web Servers, from https://www.scribd.com/doc/35607686/hacking-Module-11.

1088. Terms used by Microsoft to describe the various software updates released by it, from https://www.thewindowsclub.com/terms-used-by-microsoft-to-describe-the-various-software-updates-released-by-it.

1089. Directory Traversal Attacks, from https://www.acunetix.com/websitesecurity/directory-traversal/.

1090. Jason Chan, (2006), Essentials of Patch Management Policy and Practice, from http://www.patchmanagement.org/pmessentials.asp.

1091. Frank Kargl, Jörn Maier, Stefan Schlott, Michael Weber, Protecting Web Servers from Distributed Denial of Service Attacks, from http://www10.org/cdrom/papers/409/.

1092. Radu State, (2008), Hacking Web2, from http://www.aims-conference.org/issnsm-2008/01-WebHacking.pdf.

1093. Security issues affecting Apache httpd 2.0.40, from http://www.apacheweek.com/features/security-v2.0.40.

1094. Apache HTTP Server 2.2 vulnerabilities, from https://httpd.apache.org/security/vulnerabilities_22.html.

1095. Saumil Shah, (2003), One-way Web Hacking, from http://www.net-square.com/_assets/One-way_Web_Hacking.pdf.

1096. Shani, Oran, (2010), Definition: WEB-SITES DEFACEMENT, from http://www.freepatentsonline.com/y2010/0107247.html.

1097. Bodvoc, (2010), An Overview of a Web Server, from https://bodvoc.wordpress.com/2010/07/02/an-overview-of-a-web-server/.

1098. (2009), IIS 7.0 Architecture, from https://www.gandhipitesh.com/2009/05/iis-70-architecture.html.

1099. Robert Auger, Server Misconfiguration, from http://projects.webappsec.org/w/page/13246959/Server-Misconfiguration.

1100. Insecure Configuration Management, from http://www.owasp.org/index.php/Insecure_Configuration_Management.

1101. Robert Auger, HTTP Response Splitting, from http://projects.webappsec.org/w/page/13246931/HTTP-Response-Splitting.

1102. HTTP Response Splitting, from https://owasp.org/www-community/attacks/HTTP_Response_Splitting.

1103. (2005), Introduction to HTTP Response Splitting, from https://securiteam.com/securityreviews/5WP0E2KFGK.

1104. Tunneling protocol, from https://en.wikipedia.org/wiki/Tunneling_protocol.

1105. How to hack a Web Server, from https://www.guru99.com/how-to-hack-web-server.html.

1106. Siddharth Bhattacharya, (2009), Hacking A Web Site and Secure Web Server Techniques Used, from https://www.slideshare.net/slideshow/hacking-a-web-site-and-secure-web-server-techniques-used/1437440.

1107. (2014), What is the ultimate goal of hacking a webserver?, from https://security.stackexchange.com/questions/48705/what-is-the-ultimate-goal-of-hacking-a-webserver.

1108. DNS Hijacking: What is it and How it Works, from https://www.gohacking.com/dns-hijacking/.

1109. Niranjan, (2006), DNS Amplification Attack, from http://niriog.com/2006/03/28/dns-amplification-attack/.

1110. (2009), How to detect if your webserver is hacked and get alerted, from https://www.webdigi.co.uk/blog/2009/how-to-detect-if-your-webserver-is-hacked-and-get-alerted.

1111. Amit Klein, (2004), HTTP Response Splitting, Web Cache Poisoning Attacks, from http://www.ouah.org/whitepaper_httpresponse.pdf.

1112. Kevin Beaver, Top hacker tricks to exploit SQL Server systems, from https://www.techtarget.com/searchsecurity/tip/Ten-hacker-tricks-to-exploit-SQL-Server-systems?Offer=SQLwnha217.

1113. (2007), Windows IIS server hardening checklist, from https://www.techtarget.com/searchsecurity/feature/Windows-IIS-server-hardening-checklist.

1114. Web Server, from https://www.tutorialspoint.com/internet_technologies/web_servers.htm.

1115. Web server, from https://en.wikipedia.org/wiki/Web_server.

1116. Rajiv, (2015), Explain Web servers operation and general server characteristics?, from https://eduladder.com/viewquestions/975/Explain-Web-servers-operation-and-general-server-characteristics.

1117. Virtual hosting, from https://en.wikipedia.org/wiki/Virtual_hosting.

1118. Margaret Rouse, (2006), Virtual Hosting, from https://www.techtarget.com/whatis/definitions/V.
1119. Tim Fisher, (2019), What is a Web Proxy?, from https://www.lifewire.com/what-is-web-proxy-3081607.
1120. Web proxy servers, from https://en.wikipedia.org/wiki/Proxy_server#Web_proxy_servers.
1121. Addison Wesley Longman, 2003. Web Server Operation, from http://web.cs.wpi.edu/~kal/courses/awt/lab6/wwwch11servlets.PDF.
1122. Web Servers Hacking, from http://www.certiology.com/computing/certified-ethical-hacker/ethical-hacking-study-guide/how-to-hack-a-web-server.html.
1123. (2019), What is the Server Side Request Forgery Vulnerability & How to Prevent It?, from https://www.invicti.com/blog/web-security/server-side-request-forgery-vulnerability-ssrf/.
1124. Ian Muscat, (2019), What is Server Side Request Forgery (SSRF)?, from https://www.acunetix.com/blog/articles/server-side-request-forgery-vulnerability/.
1125. Server-side request forgery (SSRF), from https://portswigger.net/web-security/ssrf.
1126. Windows Patch Management Best Practices, from https://www.gfi.com/company/blog/2019/windows-patch-management.
1127. System Security Guidelines, from http://www.gswan.gov.in/PDF/Computer-System-Security-Guidelines-4-14122017.pdf.
1128. Monitor and Detect Who is Making Changes to Your Directories and Network Shares in Real-time, from https://directorymonitor.com/.
1129. (2019), DNS Hijacking: How to Identify and Protect Against it, from https://securitytrails.com/blog/dns-hijacking.
1130. (2020), 8 Patch Management Best Practices, from https://www.dnsstuff.com/patch-management-best-practices.
1131. Jeremy Rasmussen, (2019), Patch Management Best Practices, from https://cybersecurity.att.com/blogs/security-essentials/patching-frequency-best-practices.
1132. Sam Ingalls, (2021), Patch Management: Definition, Process and Best Practices, from https://www.serverwatch.com/guides/what-is-patch-management/#patch-management-best-practices.
1133. Security Patch Management Best Practices, from https://www.cybersecurity-automation.com/security-patch-management-best-practices/#Security_Patch_Management_Best_Practices.
1134. Gergely Kalman, 10 Most Common Web Security Vulnerabilities, from https://www.toptal.com/security/10-most-common-web-security-vulnerabilities.
1135. (2019), 21 Server Security Tips to Secure Your Server, from https://phoenixnap.com/kb/server-security-tips.
1136. Secure your Web Server – Best Practices To Follow, from https://apps.techglimpse.com/web-server-security/#.YkpkPChBzIU.
1137. David Bianco, (2019), 10 Ways to Prevent Computer Security Threats from Insiders, from https://www.techtarget.com/searchsecurity/feature/Ten-ways-to-prevent-insider-security-threats.
1138. Sabiha M, (2021), Apache web server, from https://www.slideshare.net/slideshow/apache-web-server-245775113/245775113#4.
1139. Smruti Panigrahi, (2020), Extension architecture for Apache Web Server, from https://docs.bmc.com/docs/middlewaremonitor/9/extension-architecture-for-apache-web-server-928605251.html.
1140. Apache HTTP Server 2.4 vulnerabilities, from https://httpd.apache.org/security/vulnerabilities_24.html.
1141. (2024), Apache: Security Vulnerabilities, CVEs, from https://www.cvedetails.com/vulnerability-list/vendor_id-45/Apache.html.
1142. (2023), Introduction to IIS Architectures, from https://learn.microsoft.com/en-us/iis/get-started/introduction-to-iis/introduction-to-iis-architecture.
1143. Premkumar M, (2019), NGINX — Architecture, from https://medium.com/@premsuryamj/nginx-architecture-9f97ef7887e2.
1144. Nginx architecture, from https://foxutech.com/apache-vs-nginx-architecture/nginx-architecture/.
1145. Apache Module mod_status, from https://httpd.apache.org/docs/2.4/mod/mod_status.html.
1146. Guru Baran, (2024), HTTP/2 Continuation Flood Attack: Single Machine Can Bring Down Server, from https://cybersecuritynews.com/http-2-continuation-attack/.
1147. Mario Teixeira, (2024), What you should know: HTTP/2 CONTINUATION Flood Vulnerability, from https://checkmarx.com/blog/what-you-should-know-http-2-continuation-flood-vulnerability/.
1148. Gil Cohen and Omri Iribar, (2023), Frontjacking: A New Attack That Threatens Reverse Proxy Servers, from https://cyesec.com/blog/frontjacking-new-attack-that-threatens-reverse-proxy-server.
1149. (2023), From Conference to Lab: The Emergence of Frontjacking, from https://blog.secureflag.com/2023/06/19/from-conference-to-lab-the-emergence-of-frontjacking/.
1150. Mudasser Hussain, (2023), Hacking Microsoft IIS: IIS Enumeration, from https://medium.com/@mudasserhussain1111/hacking-microsoft-iis-enumerating-iis-for-v-39de5a27f101.
1151. Abusing mod_userdir to enumerate user accounts, from https://www.oreilly.com/library/view/nmap-6-network/9781849517485/ch04s05.html.

1152. Ravindra Dagale, (2023), Nginxpwner - A helpful tool for Bug Bounty, from https://www.youtube.com/watch?v=7Dw53wsTPms&ab_channel=RavindraDagale.

1153. Ravindra Dagale, (2022), Exceptional Tool? Nginxpwner to Test and Run for Nginx Security and Bug Bounty, from https://systemweakness.com/exceptional-tool-nginxpwner-to-test-and-run-for-nginx-security-and-bug-bounty-f00221521689.

1154. (2023), Offensive Security Tool: Nginxpwner, from https://www.blacknatethicalhacking.com/tools/nginxpwner/.

1155. Path traversal via misconfigured NGINX alias, from https://www.acunetix.com/vulnerabilities/web/path-traversal-via-misconfigured-nginx-alias/.

1156. David Hamann, (2022), Nginx alias misconfiguration allowing path traversal, from https://davidhamann.de/2022/08/14/nginx-alias-traversal/.

1157. Mostafa, (2021), Path Traversal Via Misconfiguration Nginx, from https://mostafa-mano.medium.com/path-traversal-via-misconfiguration-nginx-68ba222137c6.

Module 14: Hacking Web Applications

1158. (2006), ISYOUR WEBSITE HACKABLE, from http://www.acunetix.com/vulnerability-scanner/wvsbrochure.pdf.

1159. (2006), The 21 Primary Classes of Web Application Threats, from www.netcontinuum.com/securityCentral/TopThreatTypes/index.cfm.

1160. IMS General Web Services Security Profile, from http://www.imsglobal.org/gws/gwsv1p0/imsgws_securityProfv1p0.html.

1161. Path Traversal and URIs, from https://phucjimy.wordpress.com/category/document-security/.

1162. Code Injection, from https://owasp.org/www-community/attacks/Code_Injection.

1163. J. Howard Beales, III, (2004), OWASP Web Application Security Vulnerabilities Top Ten List, from www.owasp.org/images/c/ce/OWASP_Top_Ten_2004.doc.

1164. Connection String Injection Attacks, from https://learn.microsoft.com/en-us/dotnet/framework/data/adonet/connection-string-builders.

1165. Connection String Parameter Pollution Attacks, from https://blackhat.com/presentations/bh-dc-10/Alonso_Chema/Blackhat-DC-2010-Alonso-Connection-String-Parameter-Pollution-wp.pdf.

1166. Session Prediction, from https://www.owasp.org/index.php?title=Session_Prediction&setlang=en.

1167. Robert Auger, (2010), Buffer Overflow, from http://projects.webappsec.org/w/page/13246916/Buffer-Overflow.

1168. Managed Web Application Firewall, from https://www.secureworks.com/resources/ds-managed-waf.

1169. Do you write secure code?, from https://www.slideshare.net/yuvalgo/do-you-write-secure-code-by-erez-metula.

1170. Web Parameter Tampering, from https://owasp.org/www-community/attacks/Web_Parameter_Tampering.

1171. (2009), Path Traversal, from https://owasp.org/www-community/attacks/Path_Traversal.

1172. LDAP Injection & BLIND LDAP Injection, from https://www.blackhat.com/presentations/bh-europe-08/Alonso-Parada/Whitepaper/bh-eu-08-alonso-parada-WP.pdf.

1173. Parameter Manipulation, from https://www.cgisecurity.com/owasp/html/ch11s04.html.

1174. (2016), Cross-site Scripting (XSS), from https://owasp.org/www-community/attacks/xss/.

1175. Robert "RSnake" Hansen, (2014), XSS Filter Evasion Cheat Sheet, from https://owasp.org/www-community/xss-filter-evasion-cheatsheet.

1176. Cross-Site Request Forgery (CSRF) Attack Lab, from http://www.cis.syr.edu/~wedu/seed/Labs_12.04/Web/Web_CSRF_Elgg/Web_CSRF_Elgg.pdf.

1177. CHRIS, (2004), Cross-Site Request Forgeries, from http://www.shiflett.org/articles/cross-site-request-forgeries.

1178. Web application Attack: DOS and DDOS attack, from http://funwhichuwant.blogspot.in/2012/10/webapplication-attack-dos-and-ddos.html.

1179. Preetish Panda, 2009, Web Application Vulnerabilities, from https://www.slideshare.net/slideshow/web-application-vulnerabilities/1238333.

1180. Yusuf Motiwala, (2007), Attacking XML Security, from https://www.slideshare.net/slideshow/attacking-xml-security/95630.

1181. Managing Web Services, from https://docs.oracle.com/cd/E19316-01/820-4335/ghbjk/index.html.

1182. Web Services Hacking And Hardening, from https://www.slideshare.net/rnewton/web-services-hacking-and-hardening.

1183. Shreeraj Shah, (2007), Advanced Web Services Hacking, from https://www.slideshare.net/slideshow/advanced-web-services-hacking/22060.

1184. Testing for HTTP Splitting/Smuggling (OWASP-DV-016), from https://www.owasp.org/index.php/Testing_for_HTTP_Exploit.

1185. Testing for SQL Wildcard Attacks (OWASP-DS-001), from https://wiki.owasp.org/index.php/Testing_for_SQL_Wildcard_Attacks_(OWASP-DS-001).

1186. Testing for DoS User Specified Object Allocation (OWASP-DS-004), from https://www.owasp.org/index.php/Testing_for_DoS_User_Specified_Object_Allocation_(OWASP-DS-004).

1187. Testing for Storing too Much Data in Session (OWASP-DS-008), from https://wiki.owasp.org/index.php/Testing_for_Storing_too_Much_Data_in_Session_(OWASP-DS-008).

1188. Testing for Naughty SOAP Attachments, from https://nilminus.wordpress.com/web-application-penetration-testing/web-services-testing/testing-for-naughty-soap-attachments/.

1189. Testing for AJAX (OWASP-AJ-002), from Testing for AJAX (OWASP-AJ-002), from https://wiki.owasp.org/index.php/Testing_for_AJAX_(OWASP-AJ-002).

1190. Common Web-Based Applications Attacks, from http://www.applicure.com/blog/owasp-top-10-2010#2._Injection_Flaws.

1191. Paper – Cross Site Scripting, from http://www.technicalinfo.net/papers/CSS.html.

1192. Robert J. Shimonski, (2002), Hacking techniques, from https://www.ibm.com/developerworks/security/library/s-crack/password_cracking.html.

1193. Sarah Granger, (2002), A Guide To Better Password Practices, from https://www.symantec.com/connect/articles/simplest-security-guide-better-password-practices.

1194. Bad Password Examples, from http://www.spy-hill.com/~myers/help/Passwords.html.

1195. Password cracker, from https://www.techtarget.com/searchsecurity/definition/password-cracker.

1196. John, HTTP Authentication: Basic and Digest Access Authentication, from https://www.ietf.org/rfc/rfc2617.txt.

1197. Authentication, Authorization, and Access Control, from https://httpd.apache.org/docs/2.4/howto/auth.html.

1198. The Cross-Site Scripting (XSS) FAQ, from https://www.cgisecurity.com/xss-faq.html.

1199. Quick Security Reference - Cross-Site Scripting.docx, from http://download.microsoft.com/download/E/E/7/EE789CF4-6A59-4832-BEDE-B018175F4610/Quick%20Security%20Reference%20-%20Cross-Site%20Scripting.docx.

1200. Web Application Penetration Testing, from https://wiki.owasp.org/index.php/Web_Application_Penetration_Testing.

1201. Jeff Orloff, The Big Website Guide to a Hacking Attack, from http://www.applicure.com/blog/big-website-guide-to-a-hacking-attack.

1202. What is Cross-Site Scripting (XSS)?, from http://www.applicure.com/blog/what-is-cross-site-scripting.

1203. LDAP Filters, from http://www.selfadsi.org/ldap-filter.htm.

1204. Paul Lee, (2002), Cross-site scripting, from https://www.ibm.com/developerworks/tivoli/library/s-csscript/.

1205. Cross Site Scripting Prevention Cheat Sheet, from https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html.

1206. Amit Klein, (2005), DOM Based Cross Site Scripting or XSS of the Third Kind, from http://www.webappsec.org/projects/articles/071105.shtml.

1207. Philip Tellis, (2010), Common Security Mistakes in Web Applications, from https://www.smashingmagazine.com/2010/10/common-security-mistakes-in-web-applications/.

1208. J.D. Meier, Alex Mackman, Michael Dunner, Srinath Vasireddy, Ray Escamilla and Anandha Murukan, (2003), Improving Web Application Security: Threats and Countermeasures, from https://learn.microsoft.com/en-us/previous-versions/msp-n-p/ff649874(v=pandp.10)?redirectedfrom=MSDN.

1209. Alex Homer, (2009), Components and Web Application Architecture, from https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-2000-server/bb727121(v=technet.10)?redirectedfrom=MSDN.

1210. Unvalidated Input, from https://wiki.owasp.org/index.php/Unvalidated_Input.

1211. Kevin Beaver, The importance of input validation, from https://www.techtarget.com/searchsecurity/definition/cyber-attack.

1212. (2010), Validating Input, from https://developer.apple.com/library/content/documentation/Security/Conceptual/SecureCodingGuide/Articles/ValidatingInput.html.

1213. Code injection, from https://en.wikipedia.org/wiki/Code_injection.

1214. Injection Prevention Cheat Sheet, from https://owasp.org/www-project-cheat-sheets/cheatsheets/Injection_Prevention_Cheat_Sheet.html.

1215. Remote file inclusion, from https://en.wikipedia.org/wiki/File_inclusion_vulnerability.

1216. Robert Auger, (2011), LDAP Injection, from http://projects.webappsec.org/w/page/13246947/LDAP%20Injection.

1217. Testing for LDAP Injection (OWASP-DV-006), from https://www.owasp.org/index.php?title=Testing_for_LDAP_Injection_(OTG-INPVAL-007)&redirect=no.

1218. Cross-site scripting, from https://en.wikipedia.org/wiki/Cross-site_scripting.

1219. DOM Based XSS, from https://owasp.org/www-community/attacks/DOM_Based_XSS.

1220. Phil Haack, (2008). CSRF Attacks and Web Forms, from https://haacked.com/archive/2009/04/02/csrf-webforms.aspx/.

1221. (2004). Cross-Site Request Forgeries, from http://shiflett.org/articles/cross-site-request-forgeries.

1222. Robert Auger, (2010). The Cross-Site Request Forgery (CSRF/XSRF) FAQ, from https://www.cgisecurity.com/csrf-faq.html.

1223. Application Denial of Service, from https://wiki.owasp.org/index.php/Application_Denial_of_Service.

1224. Broken Authentication and Session Management, from https://www.owasp.org/index.php/Broken_Authentication_and_Session_Management.

1225. (2009). Buffer Overflow, from https://owasp.org/www-community/vulnerabilities/Buffer_Overflow.

1226. Robert Auger, (2010). Brute Force, from http://projects.webappsec.org/w/page/13246915/Brute-Force.

1227. (2009). Session Prediction, from https://owasp.org/www-community/attacks/Session_Prediction.

1228. Robert Auger, (2010). XPath Injection, from http://projects.webappsec.org/w/page/13247005/XPath-Injection.

1229. (2009). XPATH Injection, from https://owasp.org/www-community/attacks/XPATH_Injection.

1230. Akshay Jindal, Web Application Attack: Injection flaws Attack, from http://funwhichuwant.blogspot.in/search?updated-max=2012-10-12T23:01:00-07:00&max-results=10&reverse-paginate=true&start=79&by-date=false.

1231. Preetish Panda, (2009). Web Application Vulnerabilities, from https://www.slideshare.net/slideshow/web-application-vulnerabilities/1238333.

1232. Gursev Singh Kalra, (2012). Attacking CAPTCHAs for Fun and Profit, from https://www.owasp.org/images/0/03/ASDC12-Attacking_CAPTCHAs_for_Fun_and_Profit.pdf.

1233. Gursev Singh Kalra, (2012). Random Security, from http://gursevkalra.blogspot.com/2012/03/captcha-re-riding-attack.html.

1234. nethicsit, (2013). Detecting Web Application Firewalls, from https://pentestlab.blog/2013/01/13/detecting-web-application-firewalls/.

1235. (2013). How To Hack: Wafw00f Tutorial – Web Application Firewall Detection Tool, from http://ultimatepeter.com/how-to-hack-wafw00f-tutorial-web-application-firewall-detection-tool.

1236. Testing for HTTP Splitting/Smuggling (OTG-INPVAL-016), from https://wiki.owasp.org/index.php/Testing_for_HTTP_Splitting/Smuggling_(OTG-INPVAL-016).

1237. Testing for SQL Wildcard Attacks (OWASP-DS-001), from https://wiki.owasp.org/index.php/Testing_for_SQL_Wildcard_Attacks_(OWASP-DS-001).

1238. Testing for DoS User Specified Object Allocation (OWASP-DS-004), from https://wiki.owasp.org/index.php/Testing_for_DoS_User_Specified_Object_Allocation_(OWASP-DS-004).

1239. Testing for Storing too Much Data in Session (OWASP-DS-008), from https://wiki.owasp.org/index.php/Testing_for_Storing_too_Much_Data_in_Session_(OWASP-DS-008).

1240. Testing for AJAX (OWASP-AJ-002), from https://www.owasp.org/index.php?title=Testing_for_AJAX_(OWASP-AJ-002)&setlang=es.

1241. GNU General Public License, from https://www.gnu.org/software/wget/manual/wget.html#GNU-Free-Documentation-License.

1242. David Larochelle, Statically Detecting Likely Buffer Overflow Vulnerabilities, http://lclint.cs.virginia.edu/usenix01.html.

1243. Dawn Song, Web Security, from http://inst.eecs.berkeley.edu/~cs161/fa08/Notes/nov10-xss.pdf.

1244. what is a cookie?, how cookie works, from http://www.perlservices.net/#two.

1245. CWE-79: Failure to Preserve Web Page Structure ('Cross-site Scripting'), from http://cwe.mitre.org/data/definitions/79.html.

1246. Caleb Sima, Security at the Next Level, from http://wp.bitpipe.com/resource/org_1015278444_799/webappwhitepaper2.pdf.

1247. Input Validation Attacks, from https://www.iesecure.in/input_validation.asp.

1248. (2002). Securing Your Web Applications: Anatomy of a Web Attack, from http://www.ebizq.net/topics/security/features/1713.html.

1249. Password Cracker, from https://www.techtarget.com/searchsecurity/definition/password-cracker?track=malware_glossary.

1250. HTTP cookie, from https://en.wikipedia.org/wiki/HTTP_cookie.

1251. Abodiford, (2014). Sensitive Data Exposure, from https://www.slideshare.net/abodiford/sensitive-data-exposure.

1252. (2017). XXE Injection Attacks – XML External Entity Vulnerability With Examples, from https://www.darknet.org.uk/2017/10/xxe-injection-attacks-xml-external-entity-vulnerability-examples/.

1253. Alex Coleman, User Authentication and Access Control in a Web Application, from https://selfteachtcoders.com/user-authentication-access-control-web-application/.

1254. Web Application Attack Trends, from https://www.prsecurity.com/upload/corporate/ww-en/analytics/Web-Application-Attack-Trends-2017-eng.pdf.

1255. What is Web Services?, from https://www.javatpoint.com/restful-web-services-what-is-web-services.

1256. Server-side JavaScript code injection, from https://portswigger.net/kb/issues/00100c00_server-side-javascript-code-injection.

1257. Server Side JS Injection, from https://ckarande.gitbooks.io/owasp-nodegoat-tutorial/content/tutorial/a1_-_server_side_js_injection.html.
1258. Joel Scambray, Vincent Liu, and Caleb Sima, (2011), Hacking Exposed Web Applications, 3rd edition, New York, McGraw-Hill.
1259. Log Injection, from https://owasp.org/www-community/attacks/Log_Injection.
1260. (2019), HTML Injection Tutorial: Types & Prevention with Examples, from https://www.softwaretestinghelp.com/html-injection-tutorial/.
1261. HTML Injection, from https://www.imperva.com/learn/application-security/html-injection/.
1262. Peter Yaworski, (2017), Web Hacking 101, from http://index-of.es/Miscellaneous/LIVRES/web-hacking-101.pdf.
1263. (2019), CRLF Injection and HTTP Response Splitting Vulnerability, from https://www.invicti.com/blog/web-security/crlf-http-header/.
1264. What are CRLF Injection Attacks, from https://www.acunetix.com/websitesecurity/crlf-injection/.
1265. Frame Injection, from https://secapps.com/vulndb/frame-injection.
1266. Ziyahan Albeniz, (2019), Frame Injection Attacks, from https://www.invicti.com/blog/web-security/frame-injection-attacks/.
1267. Frame Injection, from http://doc.cenzic.com/sadoc9x14ba847/CPL0001506.htm.
1268. Lokesh Gupta, Java AES 256 Encryption Decryption Example, from https://howtodoinjava.com/security/aes-256-encryption-decryption/.
1269. Broken Authentication and Session Management, from https://hdivsecurity.com/owasp-broken-authentication-and-session-management.
1270. JavaScript Hijacking, from https://vulncat.fortify.com/en/weakness.
1271. Amrita Mitra, (2017), What is JSON Hijacking or JavaScript Hijacking?, from https://www.thesecuritybuddy.com/vulnerabilities/what-is-json-hijacking-or-javascript-hijacking/.
1272. Mathy Vanhoef and Frank Piessens, (2015), RC4 NOMORE, from https://www.rc4nomore.com/.
1273. (2016), Timing Attacks in the Modern Web, from https://tom.vg/2016/08/browser-based-timing-attacks/.
1274. Catalin Cimpanu, (2019), New browser attack lets hackers run bad code even after users leave a web page, from https://www.zdnet.com/article/new-browser-attack-lets-hackers-run-bad-code-even-after-users-leave-a-web-page/.
1275. Chris Love, (2008), Breaking Down the MarioNet Service Worker Attack? Can Your Website Be Compromised?, from https://love2dev.com/pwa/marionet-attack/.
1276. Panagiotis Papadopoulos, Panagiotis Ilia, Michalis Polychronakis, Evangelos P. Markatos, Sotiris Ioannidis, and Giorgos Vasiliadis, Master of Web Puppets: Abusing Web Browsers for Persistent and Stealthy Computation, from https://www.ndss-symposium.org/wp-content/uploads/2019/02/ndss2019_018-2_Papadopoulos_paper.pdf.
1277. Zbigniew Banach, (2019), Clickjacking Attacks: What They are and How to Prevent Them, from https://www.invicti.com/blog/web-security/clickjacking-attacks/.
1278. What is clickjacking, from https://www.imperva.com/learn/application-security/clickjacking/.
1279. Chandan Kumar, (2019), Default Port Numbers You Need to Know as a Sysadmin, from https://geekflare.com/default-port-numbers/.
1280. (2020), List of TCP and UDP Port Numbers, from https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers.
1281. (2019), Scan Websites for Interesting Directories & Files with Gobuster, from https://null-byte.wonderhowto.com/how-to/scan-websites-for-interesting-directories-files-with-gobuster-0197226/.
1282. Enumerating directories used by popular web applications and servers, from https://nmap.org/nsedoc/scripts/http-enum.html.
1283. (2017), How to enumerate webserver directories using Nmap in Kali Linux, from https://ourcodeworld.com/articles/read/416/how-to-enumerate-webserver-directories-using-nmap-in-kali-linux.
1284. Zbigniew Banach, (2019), XSS Filter Evasion, from https://www.invicti.com/blog/web-security/xss-filter-evasion/.
1285. Do Son, Everythings do to bypass XSS filter, from https://securityonline.info/bypass-xss-filter/.
1286. Dafydd Stuttard and Marcus Pinto, (2011), The Web Application Hacker's Handbook, from http://index-of.es/EBooks/11_TheWeb%20Application%20Hackers%20Handbook.pdf.
1287. (2019), Password Reset Vulnerability (Poisoning), from https://www.acunetix.com/blog/articles/password-reset-poisoning/.
1288. Prosper Otemuyiwa, (2016), What is SAML?, from https://auth0.com/blog/how-saml-authentication-works/.
1289. Single Sign-On, from https://auth0.com/docs/sso/current.
1290. Jem Jensen, (2017), Attacking SSO: Common SAML Vulnerabilities and Ways to Find Them, from https://www.netspi.com/blog/technical-blog/web-application-penetration-testing/attacking-sso-common-saml-vulnerabilities-ways-find/.

1291. Macro Mosaic, (2018). Hack SAML Single Sign-on with Burp Suite, from https://null-byte.wonderhowto.com/how-to/hack-saml-single-sign-with-burp-suite-0184905/.

1292. Local File Inclusion (LFI), from https://xapax.gitbooks.io/security/content/local_file_inclusion.html.

1293. (2017). Web Application Penetration Testing, from https://www.exploit-db.com/docs/english/40992-web-app-penetration-testing---local-file-inclusion-(lfi).pdf.

1294. SQL Server 2019 connection strings, from https://www.connectionstrings.com/sql-server-2019/.

1295. (2015). SOAPAction Spoofing, from https://www.ws-attacks.org/index.php/SOAPAction_Spoofing.

1296. Prakash Dhatti, (2017). OWASP Top 10: Penetration Testing with SOAP Service and Mitigation, from https://blog.securelayer7.net/owasp-top-10-penetration-testing-soap-application-mitigation/.

1297. (2015). WS-Addressing spoofing, from https://www.ws-attacks.org/index.php/WS-Addressing_spoofing.

1298. Gutiérrez and Carlos A. WS-Address Spoofing, from https://books.google.as/books?id=dU4E_e8mtDcC&pg=PA222&lpg=PA222&dq=WS-Addressing+spoofing&source=bl&ots=A-YY25tLjQ&sig=ACfU3U3v6YrEwkwmeS-rMr0vgA4pSE_rOQ&hl=en&sa=X&ved=2ahUKEwiP4rzj5ObmAhWx4zgGHcieAm84ChDoATAAegQIBxAB#v=onepage&q=WS-Addressing%20spoofing&f=false.

1299. Web API, from https://developer.mescius.com/componentone/net-server-side-web-api/

1300. (2019). Web API, from https://en.wikipedia.org/wiki/Web_API.

1301. Introduction to web APIs, from https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Introduction.

1302. Nyna Malik, (2017). The Difference Between REST and SOAP APIs, from https://dzone.com/articles/difference-between-rest-and-soap-api.

1303. Saqwib Rizwan, (2017). An Introduction to RESTful APIs, from https://dzone.com/articles/an-introduction-to-restful-apis

1304. REST API and Its Component, from https://www.360logica.com/blog/rest-api-and-its-component/.

1305. Margaret Rouse, RESTful API (REST API), from https://www.techtarget.com/searchapparchitecture/definition/RESTful-API.

1306. Types of APIs, from https://rapidapi.com/blog/types-of-apis/.

1307. Matthew Guay, (2018). What Are Webhooks?, from https://zapier.com/blog/what-are-webhooks/.

1308. (2018). What are Webhooks? Easy Explanation & Tutorial, from https://snipcart.com/blog/what-are-webhooks-explained-example.

1309. (2017). Webhook vs API: What's the DIfference Between them?, from http://techtales.co/2017/08/07/webhook-vs-api-whats-difference/

1310. (2019). What is API Security?, from https://owasp.org/www-project-api-security/.

1311. Chris Romeo. (2019). OWASP API Security Top 10: Get your dev team up to speed, from https://blogs.opentext.com/category/technologies/security/.

1312. Jason Skowronski, (2019). Common API Vulnerabilities and How to Secure Them, from https://www.papertrail.com/blog/common-api-vulnerabilities-and-how-to-secure-them/.

1313. Thomas Bush, (2019). 5 Common API Vulnerabilities (and How to Fix Them), from https://nordicapis.com/5-common-api-vulnerabilities-and-how-to-fix-them/.

1314. Anand Srinivasan, (2017). Top 5 Vulnerabilities in APIs, from https://dataflog.com/read/top-5-vulnerabilities-in-apis/2876.

1315. MrBrijesh, (2015). Web API – Security Review – How to Hack an API and Get Away with it, from https://mrbrijesh.wordpress.com/2015/04/13/web-api-security-review-how-to-hack-an-api-and-get-away-with-it/.

1316. Ole Lensmar, (2014). API Security Testing – How to Hack an API and Get Away with It, from https://smartbear.com/blog/test-and-monitor/api-security-testing-how-to-hack-an-api-part-1/

1317. Daniel Tomescu, XML Based Attacks, from https://www.owasp.org/images/5/58/XML_Based_Attacks_-_OWASP.pdf.

1318. Bernard Harguindeguy, (2019). How Does Your API Security Stand Up Against the 3 Most Common Attacks?, from https://www.programmableweb.com/news/how-does-your-api-security-stand-against-3-most-common-attacks/sponsored-content/2019/01/03.

1319. What is Credential Stuffing?, from https://www.cloudflare.com/learning/bots/what-is-credential-stuffing/.

1320. (2020). Credential stuffing, from https://en.wikipedia.org/wiki/Credential_stuffing.

1321. Bernard Harguindeguy, (2019). How Does Your API Security Stand Up Against the 3 Most Common Attacks?, from https://www.programmableweb.com/news/how-does-your-api-security-stand-against-3-most-common-attacks/sponsored-content/2019/01/03.

1322. Mitchell Anicas, (2014). An Introduction to OAuth 2, from https://www.digitalocean.com/community/tutorials/an-introduction-to-oauth-2.

1323. Dhaval Kapil, (2017). Attacking the OAuth Protocol, from https://dhavalkapil.com/blogs/Attacking-the-OAuth-Protocol/.

1324. OAuth 2.0 Authorization Framework, from https://auth0.com/docs/protocols/oauth2.
1325. Gaurang Bhatnagar, (2018), OAuth Attacks, from https://www.slideshare.net/slideshow/pentesting-rest-apis-by-gaurang-bhatnagar/91933883.
1326. Kristopher Sandoval, (2018), 5 Ways to Hack an API (and How to Defend), from https://nordicapis.com/5-ways-to-hack-an-api-and-how-to-defend/.
1327. Gaurang Bhatnagar, (2019), Bypassing IDOR via Parameter Pollution, from https://0xgaurang.medium.com/case-study-bypassing-idor-via-parameter-pollution-78f7b3f9f59d.
1328. Agathoklis Prodromou, (2016), An Introduction to Web-shells – Part 1, from https://www.acunetix.com/blog/articles/introduction-web-shells-part-1/.
1329. (2020), Web shell, from https://en.wikipedia.org/wiki/Web_shell.
1330. Remote Access: WebShells, from https://www.rsa.com/resources/solution-briefs/.
1331. (2018), How to Upload a PHP Web Shell Using Weevely to Get Backdoor Access, from https://www.shellvoide.com/hacks/how-to-upload-php-web-shell-using-weevely-to-get-backdoor-access/.
1332. (2019), Web Shell, from https://attack.mitre.org/techniques/T1100/.
1333. Agathoklis Prodromou, (2016), Detection and Prevention – An Introduction to Web-Shells – Final Part, from https://www.acunetix.com/blog/articles/detection-prevention-introduction-web-shells-part-5/.
1334. (2014), Protecting Your APIs Against Attack and Hijack: Secure your enterprise applications for mobile, the cloud and open Web, from https://docs.broadcom.com/docs/protecting-your-apis-against-attack-and-hijack.
1335. API Security Architecture, from https://www.axway.com/en/products/api-management/manage-apis/security-architecture.
1336. Radware, (2019), How to Prevent Real-Time API Abuse, from https://blog.radware.com/security/applicationsecurity/2019/04/how-to-prevent-real-time-api-abuse/.
1337. (2018), Basic Steps for API Security, from https://www.imperva.com/blog/basic-steps-for-api-security.html.
1338. State of API Security, from https://www.soapui.org/learn/security/state-of-api-security.html
1339. Securing webhooks, from https://anymail.dev/en/stable/tips/securing_webhooks/.
1340. Webhook Signature Hash Validation and Security, from https://ordapad.help/docs/webhook-signature-hash-validation.
1341. Yakir Perlin, How to validate Cloudinary webhooks Signature?, from https://support.cloudinary.com/hc/en-us/articles/115001302471-How-to-validate-Cloudinary-webhooks-signature-.
1342. Best practices for using webhooks, from https://docs.stripe.com/webhooks.
1343. Webhook Best Practices, from https://apidocs.leadsquared.com/webhook-best-practices/
1344. Best Practices, from https://developer.bigcommerce.com/api-docs/getting-started/best-practices
1345. Web Application Security Testing, from https://www.techopedia.com/definition/29826/web-application-security-testing.
1346. Sherif Koussa, (2018), SAST, DAST, IAST, and RASP, from https://www.softwaresecured.com/what-do-sast-dast-iast-and-rasp-mean-to-developers/
1347. Thomas Scanlon, (2018), 10 Types of Application Security Testing Tools: When and How to Use Them, from https://insights.sei.cmu.edu/sei_blog/2018/07/10-types-of-application-security-testing-tools-when-and-how-to-use-them.html.
1348. Aditya Kakrania, (2017), Manual or Automated Application Security Testing: What's More Effective?, from https://blog.securityinnovation.com/manual-or-automated-application-security-testing-whats-more-effective.
1349. Sharon Solomon, (2015), Application Security Testing – Automated Vs Manual, From https://www.checkmarx.com/2015/05/19/application-security-testing-automated-vs-manual/.
1350. Adam Sedgewick, Murugiah Souppaya, and Karen Scarfone, (2015), Guide to Application Whitelisting, from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-167.pdf.
1351. Calyptix, (2017), Application Whitelisting: What it is and why it's good, from https://www.calyptix.com/top-threats/application-whitelisting-good/.
1352. Margaret Rouse, (2019), Application whitelisting, from https://www.techtarget.com/searchsecurity/definition/application-whitelisting.
1353. Allow or Block Access to Websites, from https://support.google.com/chrome/a/answer/7532419?hl=en.
1354. Paul Rubens, (2018), How to Prevent SQL Injection Attacks, from https://www.esecurityplanet.com/threats/how-to-prevent-sql-injection-attacks.html.
1355. Protecting Against SQL Injection, from https://www.hacksplaining.com/prevention/sql-injection.
1356. What is the SQL Injection Vulnerability & How to Prevent It?, from https://www.invicti.com/blog/web-security/sql-injection-vulnerability/.
1357. Primary Defenses, from https://cheatsheetseries.owasp.org/cheatsheets/OS_Command_Injection_Defense_Cheat_Sheet.html.

1358. Serge Truth, (2011), How to Test for Command Injection, from https://blog.securityinnovation.com/blog/2011/06/how-to-test-for-command-injection.html.

1359. David McMillen, (2016), The Importance of Thwarting Command Injection Attacks, from https://www.ibm.com/downloads/cas/ORNKZBR1.

1360. LDAP and LDAP Injection/Prevention, from https://www.geeksforgeeks.org/ldap-ldap-injectionprevention/.

1361. (2018), Understanding and Defending Against LDAP Injection Attacks, from https://ldap.com/2018/05/04/understanding-and-defending-against-ldap-injection-attacks/.

1362. (2017), Top 10 Common Web Attacks: The First Steps to Protect Your Website, from https://www.vpnmentor.com/blog/top-10-common-web-attacks/.

1363. Lucero Davalos Vizcarra, (2019), Top 10 Web Security Vulnerabilities to Watch Out for in 2019, from https://community.sap.com/t5/c-khhcw49343/Security/pd-p/495110619040672474461670911006425.

1364. Cross-Site Scripting (XSS), from https://phpsecurity.readthedocs.io/en/latest/Cross-Site-Scripting-(XSS).html

1365. A Positive XSS Prevention Model, from https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html.

1366. Directory Traversal, from https://portswigger.net/web-security/file-path-traversal .

1367. Preventing directory traversal, from https://www.hacksplaining.com/prevention/directory-traversal.

1368. (2013), Unvalidated Redirects and Forwards, from https://hdivsecurity.com/owasp-unvalidated-redirects-and-forwards.

1369. (2017), What is an Unvalidated Redirect/Forward?, from https://www.sitelock.com/blog/how-to-mitigate-unvalidated-redirects-forwards/.

1370. (2018), Safety Tips for Watering Hole Attacks, from https://www.techadvisory.org/2018/04/safety-tips-for-watering-hole-attacks/.

1371. Laurencem, (2018), 5 Ways to Defend Against Watering Hole Attacks, from https://www.itsasap.com/2018/07/31/5-ways-defend-watering-hole-attacks/.

1372. Resources that Need to be Protected from CSRF Vulnerability, from https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html.

1373. CSRF Attacks: Anatomy, Prevention, and XSRF Tokens, from https://www.acunetix.com/websitesecurity/csrf-attacks/

1374. (2006), Cookie Poisoning Prevention in ASP.NET, from https://www.techtarget.com/searchsecurity/definition/cookie-poisoning

1375. Nathan Rossiter, (2014), Common Web Application Attacks and How to Prevent Them, from https://www.business2community.com/crisis-management/common-web-application-attacks-prevent-0949592.

1376. Alex Mitchell, (2019), How Does One Defend Against Session Hijacking?, from https://hakin9.org/how-does-one-defend-against-session-hijacking/.

1377. Username Enumeration, from https://guides.codepath.com/websecurity/Username-Enumeration.

1378. Username enumeration, from https://docs.kentico.com/spaces/flyingpdf/pdfpageexport.action?pageId=68881105.

1379. Patrick Laverty, (2017), What is User Enumeration?, from https://www.rapid7.com/blog/post/2017/06/15/about-user-enumeration/.

1380. (2015), Username Enumeration Techniques and their Value, from https://www.ncsgroup.trust/uk/about-us/newsroom-and-events/blogs/2015/june/username-enumeration-techniques-and-their-value/.

1381. Abhinav Mishra, (2015), Abusing Password Reset Functionality to Steal User Data (Part – I), from https://www.tothenew.com/blog/abusing-password-reset-part-1/.

1382. John P. Mello Jr, What is Runtime Application Self-Protection (RASP)?, from https://blogs.opentext.com/category/technologies/security/.

1383. Bria Grangard, (2019), RASP 101: What is Runtime Application Self-Protection?, from https://www.rapid7.com/blog/post/2019/09/04/rasp-101-what-is-runtime-application-self-protection/.

1384. (2016), Defending Web Apps: WAFs versus RASPs, from https://warroom.rsmus.com/wafs-vs-rasps/.

1385. Govindraj Basatwar, (2019), What is Runtime Application Self-Protection (RASP)?, from https://www.appsealing.com/rasp-security-runtime-application-self-protection/.

1386. (2019), Bug bounty program, from https://en.wikipedia.org/wiki/Bug_bounty_program.

1387. (2021), OWASP Top 10 - 2021, from https://owasp.org/Top10/.

1388. Michael Stepankin, (2019), Exploiting JNDI Injections in Java, from https://www.veracode.com/blog/research/exploiting-jndi-injections-java.

1389. Oliver Moradov, (2021), Open Redirect Vulnerability: Impact, Severity, and Prevention, from https://brightsec.com/blog/open-redirect-vulnerabilities/.

1390. Open Redirection (Reflected), from https://portswigger.net/kb/issues/00500100_open-redirection-reflected.

1391. Zbigniew Banach, (2019). What is An Open Redirection Vulnerability and How To Prevent It, from https://www.invicti.com/blog/web-security/open-redirection-vulnerability-information-prevention/.

1392. (2017). Understanding and Discovering Open Redirect Vulnerabilities, from https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/understanding-and-discovering-open-redirect-vulnerabilities/.

1393. Sven Morgenroth, (2018). What Is an Open Redirection Vulnerability and How to Prevent It?, from https://deone.com/articles/what-is-an-open-redirection-vulnerability-and-how.

1394. What Are CRLF Injection Attacks, from https://www.acunetix.com/websitesecurity/crlf-injection/.

1395. Amar Zlojic, (2022). Server Side Request Forgery (SSRF) Attacks and How to Prevent Them, from https://brightsec.com/blog/ssrf-server-side-request-forgery/.

1396. Ben Dickson, (2021). Research: Hundreds of high-traffic web domains vulnerable to same-site attacks, from https://portswigger.net/daily-swig/research-hundreds-of-high-traffic-web-domains-vulnerable-to-same-site-attacks.

1397. Marco Squarcina, Mauro Tempesta, Lorenzo Veronese, Stefano Calzavara, and Matteo Maffei, (2021). Exploring Same-Site Attacks in the Modern Web, from https://canitakeyoursubdomain.name/.

1398. Shiva Mandalam, (2021). Real-world Examples Of Emerging DNS Attacks and How We Must Adapt, from https://www.paloaltonetworks.com/blog/2021/05/netsec-dns-attacks/.

1399. Marco Squarcina, Mauro Tempesta, and Lorenzo Veronese, (2021). Can I Take Your Subdomain? Exploring Same-Site Attacks in the Modern Web, from https://www.usenix.org/system/files/sec21-squarcina.pdf.

1400. (2021). What is a Pass-The-Cookie Attack?, from https://secureteam.co.uk/articles/web-application-security-articles/what-is-a-pass-the-cookie-attack.

1401. Jeff Warren, (2020). Bypassing MFA with Pass-the-Cookie, from https://stealthbits.com/blog/bypassing-mfa-with-pass-the-cookie.

1402. Bernard Brode, (2021). 4 Strategies to Mitigate Pass-the-Cookie Attacks, from https://www.tripwire.com/state-of-security/security-data-protection/strategies-to-mitigate-pass-the-cookie-attacks.

1403. George Mutone, (2020). 21 Top Hacking Extensions for Chrome, from https://cyberexperts.com/21-top-hacking-extensions-for-chrome/.

1404. Prajeet Nair, (2021). Data Exfiltration Enabled by Google Chrome Sync Extension, from https://www.bankinfosecurity.com/malicious-malware-enabled-by-google-chrome-sync-extension-a-15952.

1405. (2021). Cyber Attackers Using Google Chrome to Steal User Data and Credentials, from https://lifars.com/2021/11/cyber-attackers-using-google-chrome-to-steal-user-data-and-credentials/.

1406. (2022). Capturing HTTP Requests, from https://learning.postman.com/docs/sending-requests/capturing-request-data/capturing-http-requests/#using-the-postman-proxy.

1407. Michael Stepankin, (2021). Hidden OAuth Attack Vectors, from https://portswigger.net/research/hidden-oauth-attack-vectors.

1408. (2021). Protecting APIs with Layered Security, from https://tedspence.com/protecting-apis-with-layered-security-8c989fb5a19f.

1409. George Lawton, How to Take a Layered Approach to API Security, from https://www.traceable.ai/blog-post/how-to-take-a-layered-approach-to-api-security.

1410. Bernard Harguindegus, (2021). What is API Security?, from https://www.pingidentity.com/en/resources/blog/posts/2020/everything-need-know-api-security-2020.html.

1411. Jamie Juviler, (2021). 8 API Security Best Practices to Protect Sensitive Data, from https://blog.hubspot.com/website/api-security.

1412. Debbie Walkowski and Shahnawaz Backer, (2020). Securing APIs: 10 Ways to Keep Your Data and Infrastructure Safe, from https://www.f5.com/labs/articles/education/securing-apis--10-best-practices-for-keeping-your-data-and-infra.

1413. Carole Kennel and Aleksandr Nartovich, (2022). API security: 12 Essential Best Practices, from https://blog.axway.com/api-security/api-security-best-practices.

1414. Nedim Maric, (2021). 12 API Security Best Practices You Must Know, from https://brightsec.com/blog/api-security-best-practices/.

1415. Lucy Kerner, (2021). Critical API Security Risks: 10 Best Practices, from https://techbeacon.com/security/critical-api-security-risks-10-best-practices.

1416. Peter Balatazar, (2021). 10 API Security Best Practices, from https://nordicapis.com/10-api-security-best-practices/.

1417. How to Secure Webhooks: 5-Step Checklist, from https://hookdeck.com/webhooks/guides/webhooks-security-checklist#encrypt-all-data.

1418. Fienny Angelina, (2020). Best Practice to Secure your WebHooks, from https://code.likeagirl.io/best-practice-to-secure-your-webhooks-618684813723.

1419. Amir Chatrbahr, (2020). Webhooks Done Right, from https://medium.com/prospa-engineering/webhooks-done-right-676d4e74578a.

1420. (2020). Building Webhooks Into Your Application: Guidelines and Best Practices, from https://workos.com/blog/building-webhooks-into-your-application-guidelines-and-best-practices.

1421. Best Practices for Webhooks, from https://apaleo.dev/guides/webhook/best-practices.html#guarantees-and-error-handling.

1422. SQL Injection Prevention Cheat Sheet, from https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html.
1423. What is SQL Injection (SQLi) and How to Prevent It, from https://www.acunetix.com/websitesecurity/sql-injection.
1424. Sam Ingalls, (2021), How to Prevent SQL Injection Attacks in 2022, from https://www.esecurityplanet.com/threats/how-to-prevent-sql-injection-attacks/.
1425. Emil Drkusic, (2021), Learn SQL: How to prevent SQL injection attacks, from https://www.sqlshack.com/learn-sql-how-to-prevent-sql-injection-attacks/.
1426. LDAP Injection Prevention Cheat Sheet, from https://cheatsheetseries.owasp.org/cheatsheets/LDAP_Injection_Prevention_Cheat_Sheet.html.
1427. Danko Kovacic, (2021), Local File Inclusion (LFI): Understanding and Preventing LFI Attacks, from https://brightsec.com/blog/local-file-inclusion-lfi/.
1428. Remote file inclusion (RFI), from https://www.imperva.com/learn/application-security/rfi-remote-file-inclusion/.
1429. Richard Barrus, (2018), File Inclusion Vulnerabilities and Defenses Against Them, from https://www.pivotpointsecurity.com/blog/file-inclusion-vulnerabilities/.
1430. Vickie Li, (2021), Common Vulnerabilities in Java and How to Fix Them, from https://securityboulevard.com/2021/11/common-vulnerabilities-in-java-and-how-to-fix-them/.
1431. Log Injection-Tampering-Forging, from https://capec.mitre.org/data/definitions/93.html.
1432. What Are CRLF Injection Attacks, from https://www.acunetix.com/websitesecurity/crlf-injection.
1433. (2022), XSS and CRLF injection prevention with HST-2, from https://xmldocumentation.bloomreach.com/library/concepts/web-application/xss-and-crlf-injection-prevention-with-hst-2.html.
1434. Preritpathak, (2020), CRLF Injection Attack, from https://www.geeksforgeeks.org/crlf-injection-attack.
1435. Sudip Sengupta, (2021), Broken Access Control and How to Prevent It, from https://crashtest-security.com/broken-access-control-prevention/.
1436. OWASP Top 10 Vulnerabilities, from https://snyk.io/learn/owasp-top-10-vulnerabilities/.
1437. Muzaffer Pasha, Top 5 Ways To Protect Against Data Exposure, from https://www.traceable.ai/blog-post/top-5-ways-to-protect-against-data-exposure.
1438. Borislav Kiprin, (2021), Your Guide to Sensitive Data Exposure (Fuzzing) & How to Fix It, from https://crashtest-security.com/sensitive-data-exposure/.
1439. Cynthia Konan, Insufficient Due Diligence as a Security and Privacy Issue, from https://repository.stcloudstate.edu/cgi/viewcontent.cgi?article=1118&context=msia_etds.
1440. (2018), The OWASP Top 10: Broken Authentication and Session Management, from https://www.sitelock.com/blog/owasp-top-10-broken-authentication-session-management/.
1441. Diego Poza, (2020), What Is Broken Authentication?, from https://auth0.com/blog/what-is-broken-authentication/.
1442. Borislav Kiprin, (2021), What Is Insecure Deserialization and How to Prevent It, from https://crashtest-security.com/insecure-deserialization/.
1443. Deserialization Cheat Sheet, from https://cheatsheetseries.owasp.org/cheatsheets/Deserialization_Cheat_Sheet.html.
1444. Admir Dodar, (2021), Deserialization: How it Works and Protecting Your Apps, from https://brightsec.com/blog/deserialization/.
1445. (2021), Insecure Deserialization in Web Application, from https://cybersecurityresearch.tech/insecure-deserialization-web-application/.
1446. Javan Rasokat, (2019), How to Prevent Insufficient Logging and Monitoring, from https://medium.com/@javan.rasokat/owasp-appsensor-logging-and-monitoring-2518712ce0fe.
1447. (2019), Insufficient Logging and Monitoring, from https://www.wallarm.com/what/insufficient-logging-monitoring.
1448. (2021), OWASP Top 10 in 2021: Security Logging and Monitoring Failures Practical Overview, from https://www.immuniweb.com/blog/OWASP-security-logging-and-monitoring-failures.html.
1449. Amar Zlojic, (2022), Server Side Request Forgery (SSRF) Attacks and How to Prevent Them, from https://brightsec.com/blog/ssrf-server-side-request-forgery/.
1450. Marc Dahan, (2022), Server-side Request Forgery (SSRF) Attacks and How to Prevent Them, from https://www.comparitech.com/blog/information-security/server-side-request-forgery-attacks/.
1451. Ian Muscat, (2022), What is Server-Side Request Forgery (SSRF)?, from https://www.acunetix.com/blog/articles/server-side-request-forgery-vulnerability/.
1452. Ajay Ohri, (2021), SSRF (Server-Side Request Forgery): An Easy Guide For 2021, from https://www.jigsawacademy.com/blogs/cyber-security/ssrf/.
1453. Clickjacking Defense Cheat Sheet, from https://cheatsheetseries.owasp.org/cheatsheets/Clickjacking_Defense_Cheat_Sheet.html.

1454. Andrea Chiarelli, (2020), Clickjacking Attacks and How to Prevent Them, from https://auth0.com/blog/preventing-clickjacking-attacks/#Prevent-Clickjacking-Attacks.

1455. Tomasz Andrzej Nidecki, (2019), Clickjacking – What is It and How To Defend Yourself, from https://www.acunetix.com/blog/web-security-zone/defend-against-clickjacking-attacks/#:~:text=%20How%20to%20Defend%20against%20Clickjacking%20Attacks%20,Step%204%3A%20Scan%20regularly%20%2Bwith%20Acunetix%29%20More%20.

1456. JSON Hijacking (aka JavaScript Hijacking), from https://capec.mitre.org/data/definitions/111.html.

1457. JavaScript Hijacking: Vulnerable Framework, from https://vulncat.fortify.com/en/detail?id=desc.config.java.javascript_hijacking_vulnerable_framework.

1458. Ben Diamant, (2019), The Client Side Battle Against JavaScript Attacks Is Already Here, from https://medium.com/swlh/the-client-side-battle-against-javascript-attacks-is-already-here-656f3602c1f2.

1459. (2021), Client-side Attacks: What They Are and How to Prevent Them, from https://www.ensighten.com/blog/client-side-website-data-theft-javascript.

1460. Avoiding User Enumeration, from https://www.hacksplaining.com/prevention/user-enumeration.

1461. Patrick Laverty, (2019), What Is User Enumeration?, from https://www.rapid7.com/blog/post/2017/06/15/about-user-enumeration.

1462. Kumar Chandrakant, (2019), Preventing Username Enumeration Attacks with Spring Security, from https://www.baeldung.com/spring-security-enumeration-attacks.

1463. Username Enumeration, from https://www.virtuesecurity.com/kb/username-enumeration.

1464. Forgot Password Cheat Sheet, from https://cheatsheatseries.owasp.org/cheatsheets/Forgot_Password_Cheat_Sheet.html.

1465. (2024), Upgrade Header Smuggling, from https://book.hacktricks.xyz/pentesting-web/h2c-smuggling.

1466. Business logic vulnerabilities, from https://portswigger.net/web-security/logic-flaws.

1467. Magecart, from https://www.imperva.com/learn/application-security/magecart/.

1468. (2024), What is Magecart?, from https://sansec.io/what-is-magecart.

1469. (2024), Cross-site WebSocket hijacking, from https://portswigger.net/web-security/websockets/cross-site-websocket-hijacking.

1470. (2024), WebSocket Attacks, from https://book.hacktricks.xyz/pentesting-web/websocket-attacks.

1471. Ivan Lee, (2024), What is Magecart Attack? How To Prevent It?, from https://www.wallarm.com/what/what-is-magecart-attack-how-to-prevent-it#.

1472. What Is Magecart?, from https://www.akamai.com/glossary/what-is-magecart.

1473. Do Son, (2021), STEWS: Security Testing and Enumeration of Websockets, from https://securityonline.info/stews-security-testing-and-enumeration-of-websockets/#google_vignette.

1474. Huzaifa Tahir, (2020), Methods to Bypass Rate Limit, from https://huzaifa-tahir.medium.com/methods-to-bypass-rate-limit-5185e6c67ecd.

1475. (2024), Rate Limit Bypass, from https://book.hacktricks.xyz/pentesting-web/rate-limit-bypass.

1476. Keshav Malik, (2020), Bypassing Rate Limit like a PRO!, from https://infosecwriteups.com/bypassing-rate-limit-like-a-pro-5f3e40250cd3c.

1477. Prasad Pathak, (2024), LEARN365: Day 1 | 2FA Bypass Techniques | Part 1, from https://medium.com/@_prasad4/learn365-day-1-2fa-bypass-techniques-part-1-ebc9731a756f.

1478. Harsh Bothra, (2021), Bypassing the Protections — MFA Bypass Techniques for the Win, from https://www.cobalt.io/blog/bypassing-the-protections-mfa-bypass-techniques-for-the-win.

1479. (2024), Testing for WebSockets security vulnerabilities, from https://portswigger.net/web-security/websockets#intercepting-and-modifying-websocket-messages.

1480. (2023), Manipulating WebSocket handshakes with Burp Suite, from https://www.youtube.com/watch?v=NI2TM4-id2w&ab_channel=PortSwigger.

1481. (2023), Manipulating WebSocket messages with Burp Suite, from https://www.youtube.com/watch?v=Wcd1Lskvj4c&ab_channel=PortSwigger.

1482. (2023), OWASP Top 10 API Security Risks – 2023, from https://owasp.org/API-Security/editions/2023/en/0x11-t10/.

1483. Binary to base64: Convert between bytes and base64, from https://cryptii.com/pipes/binary-to-base64.

1484. ASCII Text to Hex Code Converter, from https://www.rapidtables.com/convert/number/ascii-to-hex.html.

Module 15: SQL Injection

1485. [2008). Preventing SQL Injections in Online Applications: Study, Recommendations and Java Solution Prototype Based on the SQL DOM, from http://mirror.kioss.undip.ac.id/pustaka-bebas/library-sw-hw/linux-1/security/WebGoat/OWASP/OWASP-AppSecEU08-Janot.pdf.

1486. San-Tsai Sun, (2007). Classification of SQL Injection Attacks, from http://courses.ece.ubc.ca/412/term_project/reports/2007-fall/Classification_of_SQL_Injection_Attacks.pdf.

1487. Victor Chapela, Advanced SQL Injection, from https://www.slideshare.net/amiable_indian/advanced-sql-injection.

1488. [2012). Advanced SQL Injection, from http://blogs.pages.ir/1341.

1489. Dmitry Evteev, (2009). Advanced SQL Injection, from https://www.slideshare.net/slideshow/advanced-sql-injection-eng/2532493.

1490. [2005). SQL injection, from https://www.techtarget.com/searchsoftwarequality/definition/SQL-injection.

1491. What is SQL Injection?, from https://www.secpoint.com/sql-injection.html.

1492. Rise in SQL Injection Attacks Exploiting Unverified User Data Input, from https://learn.microsoft.com/en-us/security-updates/SecurityAdvisories/2008/954462.

1493. [2006). Injection Protection, from https://learn.microsoft.com/en-us/previous-versions/sql/legacy/aa224806(v=sql.80)?redirectedfrom=MSDN.

1494. Understanding SQL Injection, from https://tools.cisco.com/security/center/resources/sql_injection.

1495. SQL Injection, from https://owasp.org/www-community/attacks/SQL_Injection.

1496. San-Tsai Sun, Ting Han Wei, and Stephen Liu, Sheung Lau, (2007). Classification of SQL Injection Attacks, from http://courses.ece.ubc.ca/412/term_project/reports/2007-fall/Classification_of_SQL_Injection_Attacks.pdf.

1497. Krzysztof Kotowicz, (2010). SQL injection. Complete walkthrough (not only) for PHP developers, from https://www.slideshare.net/kkotowicz/sql-injection-complete-walktrough-not-only-for-php-developers.

1498. [2007). INTRODUCTION, from http://isea.nitk.ac.in/publications/web.pdf.

1499. Dmitry Evteev, (2009). Advanced SQL Injection, from http://www.ptsecurity.com/download/PT-devteev-Advanced-SQL-Injection-ENG.zip.

1500. Cameron Hotchkies, (2004). Blind SQL Injection Automation Techniques, from http://www.blackhat.com/presentations/bh-usa-04/bh-us-04-hotchkies/bh-us-04-hotchkies.pdf.

1501. SQL Injection, from https://learn.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/ms161953(v=sql.105)?redirectedfrom=MSDN.

1502. SQL INJECTION, from http://www.authorstream.com/Presentation/useful-155975-sql-injection-hacking-computers-22237-education-ppt-powerpoint/.

1503. Ferruh Mavituna, (2007). SQL Injection Cheat Sheet, from https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/.

1504. K. K. Mookhey and Nilesh Burghate, (2004). Detection of SQL Injection and Cross-site Scripting Attacks, from https://www.symantec.com/connect/articles/detection-sql-injection-and-cross-site-scripting-attacks.

1505. Debasish Das, Utpal Sharma, and D.K. Bhattacharyya, (2010). An Approach to Detection of SQL Injection Attack Based on Dynamic Query Matching, from https://www.ijcaonline.org/journal/number25/pxc387766.pdf.

1506. [2010). Quick Security Reference: SQL Injection, from http://download.microsoft.com/download/E/E/7/EE709CF4-6A59-4832-8EDE-B81817SF461C/Quick%20Security%20Reference%20-%20SQL%20Injection.docx.

1507. Ferruh Mavituna, One Click Ownage, Adventures of a lazy pentester, from https://www.owasp.org/images/8/8e/One_Click_Ownage-Ferruh_Mavituna.pdf.

1508. Alexander Kornbrust, (2009). ODTUG - SQL Injection Crash Course for Oracle Developers, from http://www.red-database-security.com/wp/ODW2009_sql_crashcourse_for_developers.pdf.

1509. William G.J. Halfond, Jeremy Viegas, and Alessandro Orso, (2006). A Classification of SQL Injection Attack Techniques and Countermeasures, from https://faculty.cc.gatech.edu/~orso/papers/halfond.viegas.orso.ISSSE06.presentation.pdf.

1510. Injection Flaws - OWASP, from https://www.owasp.org/index.php/Injection_Flaws.

1511. Victor Chapela, (2005). Advanced SQL Injection, from http://www.owasp.org/images/7/74/Advanced_SQL_Injection.ppt.

1512. [2010). SQL Injection, from https://learn.microsoft.com/en-us/previous-versions/sql/sql-server-2008/ms161953(v=sql.100)?redirectedfrom=MSDN.

1513. Blind SQL Injection, from http://www.evilsql.com/main/page1.php.

1514. SQL Injection, from https://www.w3schools.com/sql/sql_injection.asp.

1515. SQL Injection Cheat Sheet & Tutorial: Vulnerabilities & How to Prevent SQL Injection Attacks, from https://www.veracode.com/security/sql-injection.

1516. Types of SQL Injection (SQLi), from https://www.acunetix.com/websitesecurity/sql-injection2/.

1517. Everything You Need to Know About SQL Injection Attacks & Types, SQLi Code Example, Variations, Vulnerabilities & More, from http://www.firewall.cx/general-topics-reviews/web-application-vulnerability-scanners/1207-how-sql-injection-attacks-work-examples.html.

1518. Hack2Secure, (2017). Understanding SQL Injection Attacks, from https://www.hack2secure.com/blogs/understanding-sql-injection-attacks.

1519. Using Comments to Simplify SQL Injection, from https://www.sqlinjection.net/comments/.

1520. SQL Injection Cheat Sheet, from https://www.invicti.com/blog/web-security/sql-injection-cheat-sheet/#InlineComments.

1521. (2017). SQL Injection Tutorial, from https://www.w3resource.com/sql/sql-injection/sql-injection.php.

1522. Nuno Loureiro, (2010). Advanced SQL Injection: Attacks, from https://www.slideshare.net/slideshow/advanced-sql-injection-attacks/5767626.

1523. Types of SQL Injection Attacks, from http://hwang.cisdept.cpp.edu/swanew/Text/SQL-Injection.htm.

1524. Time-Based Blind SQL Injection using Heavy Query, from https://www.sqlinjection.net/heavy-query/.

1525. (2017). SQL Injection Bypassing WAF, from https://www.owasp.org/index.php/SQL_Injection_Bypassing_WAF.

1526. (2013). WAF SQL evasion using HPF technique, from https://security.stackexchange.com/questions/34498/waf-sql-evasion-using-hpf-technique.

1527. (2009). HTTP Parameter Fragmentation (HPF) is one of the methods to bypass security filters in web applications, from http://blog.ptsecurity.com/2009/12/http-parameter-fragmentation-hpf-is-one.html.

1528. Luca Carettoni and Stefano di Paola, HTTP Parameter Pollution, from https://www.owasp.org/images/b/ba/AppsecEU09_CarettoniDiPaola_v0.8.pdf.

1529. Steve Friedl, (2017). SQL Injection Attacks by Example, from http://www.unixwiz.net/techtips/sql-injection.html.

1530. Simone Quatrini and Marco Rondini, "Blind Sql Injection with Regular Expressions Attack", from https://www.exploit-db.com/docs/english/17397-blind-sql-injection-with-regular-expressions-attack.pdf.

1531. (2010). Hacking website using SQL injection -step by step guide, from https://breakthesecurity.cysecurity.org/2010/12/hacking-website-using-sql-injection-step-by-step-guide.html.

1532. Chandrakant Nal, How to Find Admin Login Panel of a Website, from http://www.darksite.co.in/2013/04/how-to-find-admin-login-panel-of.html.

1533. SQL Injection techniques, from https://www.oratechinfo.co.uk/sql_injection.html.

1534. PL/SQL Attacks, from https://www.sqlinjection.net/advanced/pl-sql/.

1535. PL/SQL Security Vulnerabilities and Language Overview, from https://www.checkmarx.com/plsql-security-vulnerabilities-platform-overview/.

1536. MacGyver, (2013). Create server backdoors using SQL injection, from http://macgyverdev.blogspot.in/2013/02/create-server-backdoors-using-sql.html.

1537. (2012). Creating Backdoors Using SQL Injection, from https://www.infosecinstitute.com/resources/hacking/backdoor-sql-injection/#gref.

1538. (2011). Leveraging a shell from SQL injection, from https://security.stackexchange.com/questions/6935/leveraging-a-shell-from-sql-injection.

1539. SQL Injection Attack, from https://shodhganga.inflibnet.ac.in/bitstream/10603/123504/7/chapter%202.pdf.

1540. Advanced evasion techniques for defeating SQL injection input validation mechanisms, from https://jeannotclement.wordpress.com/2011/01/16/advanced-evasion-technique/.

1541. Evasion Techniques, from https://linux.nl/mirror/apachesecurity/0596007248/apachesc-CHP-10-SECT-8.html.

1542. Intrusion detection system evasion techniques, from https://en.wikipedia.org/wiki/Intrusion_detection_system_evasion_techniques.

1543. Do Son, (2019). SQL injection: 9 Ways to Bypass Web Application Firewall, from https://securityonline.info/sql-injection-9-ways-bypass-web-application-firewall/.

1544. SQL Injection Bypass Cheat Sheet, from http://www.0xby.com/873.html.

1545. System Tables and Views, from https://docs.oracle.com/database/timesten-18.1/TTSYS/systemtables.htm#TTSYS722.

1546. Grant Tables, from https://dev.mysql.com/doc/refman/8.0/en/grant-tables.html.

1547. (2017). Remove references to undocumented system tables, from https://docs.microsoft.com/en-us/sql/sql-server/install/remove-references-to-undocumented-system-tables?view=sql-server-2014.

1548. (2017). sys.syslogins (Transact-SQL), from https://learn.microsoft.com/en-gb/sql/relational-databases/system-compatibility-views/sys-syslogins-transact-sql?view=sql-server-2017.

1549. Victor Chapela, (2005), Advanced SQL Injection, from https://www.owasp.org/images/7/74/Advanced_SQL_Injection.ppt.
1550. sys.sysdatabases (Transact-SQL), from https://learn.microsoft.com/en-us/sql/relational-databases/system-compatibility-views/sys-sysdatabases-transact-sql?view=sql-server-2017.
1551. sys.sysobjects (Transact-SQL), from https://learn.microsoft.com/en-us/sql/relational-databases/system-compatibility-views/sys-sysobjects-transact-sql?view=sql-server-2017.
1552. sys.syscolumns (Transact-SQL), from https://learn.microsoft.com/en-us/sql/relational-databases/system-compatibility-views/sys-syscolumns-transact-sql?view=sql-server-2017.
1553. sys.sysservers (Transact-SQL), from https://learn.microsoft.com/en-us/sql/relational-databases/system-compatibility-views/sys-sysservers-transact-sql?view=sql-server-2017.
1554. xp_cmdshell Server Configuration Option, from https://learn.microsoft.com/en-us/sql/database-engine/configure-windows/xp-cmdshell-server-configuration-option?view=sql-server-2017.
1555. (2012), SQL Injection through HTTP Headers, from https://www.infosecinstitute.com/resources/application-security/sql-injection-http-headers/#gref.
1556. Nikos Danopoulos, (2019), X Forwarded for SQL Injection, from https://outpost24.com/blog/X-forwarded-for-SQL-injection.
1557. Header Field Definitions, from https://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html.
1558. (2019), Intrusion Prevention, from https://fortiguard.com/encyclopedia/ips/41512.
1559. Exfiltrating Data From MS SQL Server Via DNS, from http://pentestmonkey.net/blog/mssql-dns.
1560. Mike Saunders, (2018), SQLi Data Exfiltration via DNS, from https://www.redsiege.com/blog/2018/11/sqli-data-exfiltration-via-dns/.
1561. (2014), DNS Exfiltration with SQL Injection, from https://blog.safebuff.com/2016/05/14/DNS-exfiltration-with-SQL-Injection/.
1562. SQL Injection: Vulnerabilities & How to Prevent SQL Injection Attacks, from https://www.veracode.com/security/sql-injection.
1563. (2018), How to Avoid Detection & Bypass Defenses, from https://null-byte.wonderhowto.com/how-to/sql-injection-101-avoid-detection-bypass-defenses-0184918/.
1564. Service overview and network port requirements for Windows, from https://learn.microsoft.com/en-us/troubleshoot/windows-server/networking/service-overview-and-network-port-requirements.
1565. Thomas Sermpinis, Web Application Hacking Advanced SQL Injection and Data Store Attacks, Vol 12, from Book.
1566. Tomasz Andrzej Nidecki, (2020), NoSQL Injections and How to Avoid Them, from https://www.acunetix.com/blog/web-security-zone/nosql-injections/.
1567. Zbigniew Banach, (2020), What is NoSQL Injection and How Can You Prevent It?, from https://www.invicti.com/blog/web-security/what-is-nosql-injection/.
1568. Howard Poston, (2020), What is NoSQL injection?, from https://www.infosecinstitute.com/resources/application-security/what-is-nosql-injection/.
1569. (2021), SQL Injection, from https://learn.microsoft.com/en-us/sql/relational-databases/security/sql-injection?view=sql-server-ver15.
1570. Understanding SQL Injection, from https://sec.cloudapps.cisco.com/security/center/resources/sql_injection.html.
1571. Evan Klein, (2019), How to Defend Your Business Against SQL Injections, from https://logz.io/blog/defend-against-sql-injections/.
1572. How to Protect Against SQL Injection Attacks, from https://security.berkeley.edu/education-awareness/how-protect-against-sql-injection-attacks.
1573. SQL Injection Prevention Cheat Sheet, from https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html.
1574. Ed Pollack, (2019), SQL Injection: Detection and prevention, from https://www.sqlshack.com/sql-injection-detection-and-prevention/.
1575. Borislav Kiprin, (2021), What Is SQL Injection? Types, Examples, Prevention, from https://crashtest-security.com/sql-injections/.
1576. Jack Pincombe, (2022), Abusing JSON-Based SQL, from https://www.imperva.com/blog/abusing-json-based-sql/.
1577. (2024), sys.objects (Transact-SQL), from https://learn.microsoft.com/en-us/sql/relational-databases/system-catalog-views/sys-objects-transact-sql?view=sql-server-ver16.
1578. (2023), MySQL LOAD_FILE() function, from https://www.w3resource.com/mysql/string-functions/mysql-load_file-function.php.

Module 16: Hacking Wireless Networks

1579. Jari Arkko, Vesa Torvinen, Aki Niemi, (2002), HTTP Authentication with EAP, from http://www.arkko.com/publications/draft-torvinen-http-eap-01.txt.
1580. Peter Loshin, (2019), Defending against the most common wireless network attacks, from https://www.techtarget.com/searchsecurity/feature/A-list-of-wireless-network-attacks.

1581. Ajay Kumar Gupta, (2010), Comment: Rogue Access Point Setups on Corporate Networks, from https://www.infosecurity-magazine.com/opinions/comment-rogue-access-point-setups-on-corporate/.

1582. Bluetooth Security Risks and Tips to Prevent Security Threats, from https://www.brighthub.com/computing/smb-security/articles/30045.aspx.

1583. Cisco Unified Wireless Network Architecture—Base Security Features, from https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Mobility/secwlandg20/sw2dg/ch4_2_SPMb.pdf.

1584. Chris Weber and Gary Bahadu, (2009), Wireless Networking Security, from https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-xp/bb457019(v=technet.10)?redirectedfrom=MSDN.

1585. (2006), How to Cheat at Securing a Wireless Network, from https://www.sciencedirect.com/science/article/pii/B9781597490870500572.

1586. Lisa Phifer, (2005), Eliminating interference thru Wi-Fi spectrum analysis, from https://searchmobilecomputing.techtarget.com/tip/Eliminating-interference-thru-Wi-Fi-spectrum-analysis.

1587. Understanding WiFi Hotspots... from https://www.scambusters.org/wifi.html.

1588. (2009), How 802.11 Wireless Works, from https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc757419(v=ws.10)?redirectedfrom=MSDN.

1589. TKIP (Temporal Key Integrity Protocol), from https://www.tech-faq.com/tkip-temporal-key-integrity-protocol.html.

1590. Kevin Beaver and Peter T. Davis, Understanding WEP Weaknesses, from https://www.dummies.com/category/articles/networking-33581/.

1591. Eric Geier, (2010), 7 Things Hackers Hope You Don't Know, from https://www.esecurityplanet.com/views/article.php/3891710/7-Things-Hackers-Hope-You-Dont-Know.htm.

1592. Rogue Wireless Access Point, from https://www.tech-faq.com/rogue-wireless-access-point.html.

1593. How to War Drive, from https://www.wikihow.com/War-Drive.

1594. ALFRED LOO, (2009), Security Threats of Smart Phones and Bluetooth, from http://www.aaronfrench.com/coursefiles/ucommerce/Loo_2009.pdf.

1595. Prabhaker Mateti, Hacking Techniques in Wireless Networks, from https://web1.cs.wright.edu/~pmateti/InternetSecurity/Lectures/WirelessHacks/Mateti-WirelessHacks.htm.

1596. Bradley Mitchell, (2020) Wired vs. Wireless Networking, from https://www.lifewire.com/wired-vs-wireless-networking-816352.

1597. Bradley Mitchell, (2019), Wireless Standards - 802.11b 802.11a 802.11g and 802.11n, from https://www.lifewire.com/wireless-standards-802-11a-802-11b-g-n-and-802-11ac-816553.

1598. Wi-Fi Protected Access, from https://www.techtarget.com/searchmobilecomputing/definition/Wi-Fi-Protected-Access.

1599. WPA (Wi-Fi Protected Access), From https://www.tech-faq.com/wpa-wi-fi-protected-access.shtml.

1600. Paul Arana, (2006), Benefits and Vulnerabilities of Wi-Fi Protected Access 2 (WPA2), from https://cs.gmu.edu/~yhwang1/INFS612/Sample_Projects/Fall_06_GPN_6_Final_Report.pdf.

1601. Wireless Hacking, from https://www.darknet.org.uk/category/wireless-hacking/.

1602. Gary Wollenhaupt, How Cell Phone Jammers work, from https://electronics.howstuffworks.com/cell-phone-jammer1.htm.

1603. Brian R. Miller and Booz Allen Hamilton, (2002), Issues in Wireless security, from https://www.acsac.org/2002/case/wed-c-330-Miller.pdf.

1604. Jonathan Hassell, (2004), Wireless Attacks and Penetration Testing, from https://www.symantec.com/connect/articles/wireless-attacks-and-penetration-testing-part-1-3.

1605. Martin Beck and TU-Dresden, (2008), Practical attacks against WEP and WPA, from http://dl.aircrack-ng.org/breakingwepandwpa.pdf.

1606. Chris Hurley, Finding cloaked access points, (Chapter 9), from https://books.google.co.in/books?id=wGJhD9spE3wC&pg=PA333&lpg=PA333&dq=cloaked+access+point&source=bl&ots=ZDkM5ykDNV&sig=1sLKlx-1ZcqkhUdr1WpFaqYcyI&hl=en&ei=V8R2Ss35Oo2e6gP59viqCw&sa=X&oi=book_result&ct=result&redir_esc=y#v=onepage&q=cloaked%20access%20point&f=false.

1607. (2002), Wireless Scanning Wardriving / Warchalking, from https://www.it-observer.com/wireless-scanning-wardriving-warchalking.html.

1608. Wireless Network, from http://www.hackingthesunverse.com/information-security/attack-vs-defense/attack-vs-defense-on-an-organizational-scale/5-wireless-network.

1609. Michael Roche, (2007), Wireless Attack Tools, from https://www.cse.wustl.edu/~jain/cse571-07/ftp/wireless_hacking.pdf.

1610. Protecting your wireless network from hacking, from http://www.businessknowledgesource.com/technology/protecting_your_wireless_network_from_hacking_025027.html.

1611. Agustina, J. V. Peng Zhang, and Kantola, (2003), Performance evaluation of GSM handover traffic in a GPRS/GSM network, from https://ieeexplore.ieee.org/document/1214113?isnumber=27298&arnumber=1214113&count=217&index=21.

1612. Service set identifier, from https://www.techtarget.com/searchmobilecomputing/definition/service-set-identifier.

1613. Antenna Cabling Guide, from http://wireless.gumph.org/content/3/12/011-antenna-cabling.html.

1614. The Wireless Intrusion detection system, from http://www.forum-intrusion.com/widz_design.pdf.

1615. Humphrey Cheung, (2005), How To Crack WEP - Part 1: Setup & Network Recon, from https://www.tomsguide.com/tag/security.

1616. Humphrey Cheung, (2005), How To Crack WEP - Part 2: Performing the Crack, from https://www.tomsguide.com/us/how-to-crack-wep,review-459.html.

1617. Humphrey Cheung, (2005), How To Crack WEP - Part 3: Securing your WLAN, from https://www.tomsguide.com/us/how-to-crack-wep,review-471.html.

1618. Chris Weber and Gary Behadur, (2009), Wireless Networking Security, from https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-xp/bb457019(v=technet.10)?redirectedfrom=MSDN.

1619. (2009), How 802.11 Wireless Works, from https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc757419(v=ws.10)?redirectedfrom=MSDN.

1620. Brandon Teska, (2008), How To Crack WPA / WPA2, from https://www.smallnetbuilder.com/wireless/wireless-howto/30278-how-to-crack-wpa--wpa2.

1621. A Comprehensive Review of 802.11 Wireless LAN Security and the Cisco Wireless Security Suite, from https://www.cisco.com/c/en/us/products/collateral/wireless/aironet-1200-access-point/prod_white_paper09186a00800b469f.html.

1622. (2006), How To Crack WEP and WPA Wireless Networks, from http://121space.com/index.php?showtopic=3376.

1623. (2009), How to prevent wireless DoS attacks, from https://www.techtarget.com/searchsecurity/feature/How-to-prevent-wireless-DoS-attacks.

1624. Jim Geier, (2003), Denial of Service a Big WLAN issue, from https://www.esecurityplanet.com/networks/types-of-ddos-attacks/.

1625. Peter Loshin, (2009), A list of wireless network attacks, from https://www.techtarget.com/searchsecurity/feature/A-list-of-wireless-network-attacks.

1626. Lisa Phifer, (2009), A wireless network vulnerability assessment checklist, from https://www.techtarget.com/searchsecurity/features.

1627. Lisa Phifer, (2009), Hunting for rogue wireless devices, from https://searchsecurity.techtarget.com/feature/Hunting-for-rogue-wireless-devices.

1628. PreciousJohnDoe, List of Wireless Network Attacks, from http://www.brighthub.com/computing/smb-security/articles/53949.aspx.

1629. Security Disciplines for Objective 3: Detection and Recovery, from https://oja.ojp.gov/sites/g/files/xyckuh186/files/media/document/wirelesssecurity.pdf.

1630. Jim Geier, How to: Sniff Wireless Packets with Wireshark, from http://www.wireless-nets.com/resources/tutorials/sniff_packets_wiroshark.html.

1631. Laurent Oudot, (2004), Wireless Honeypot Countermeasures, from https://www.symantec.com/connect/articles/wireless-honeypot-countermeasures.

1632. (2009), Fragmentation Attack, from http://www.aircrack-ng.org/doku.php?id=fragmentation.

1633. Andrei A. Mikhailovsky, Konstantin V. Gavrilenko, and Andrew Vladimirov, (2004), The Frame of Deception: Wireless Man-in-the-Middle Attacks and Rogue Access Points Deployment, from http://www.informit.com/articles/article.aspx?p=353735&seqNum=7.

1634. Renee Oricchio, How to Surf Safely on Public Wi-Fi, from https://www.inc.com/telecom/articles/200707/wifi.html.

1635. What is WiFi, from https://www.scambusters.org/wifi.html.

1636. Trishna Panse and Prashant Panse, (2013), A Survey on Security Threats and Vulnerability attacks on Bluetooth Communication, from http://www.ijcsit.com/docs/Volume5%204/Vol4Issue5/ijcsit20130405121.pdf.

1637. How to Bluejack, from https://www.wikihow.com/Bluejack.

1638. John Padgette and Karen Scarfone, (2012), Guide to Bluetooth Security (Draft), from https://csrc.nist.gov/csrc/media/publications/sp/800-121/rev-1/final/documents/draft-sp800-121_rev1.pdf.

1639. Nateq Be-Nazir Ibn Minar and Mohammed Tanque, (2012), BLUETOOTH SECURITY THREATS AND SOLUTIONS: A SURVEY, from http://airccse.org/journal/ijdps/papers/0112ijdps10.pdf.

1640. Lisa Phifer, Wireless network troubleshooting: Connectivity, from https://www.techtarget.com/searchnetworking/tip/Wireless-network-troubleshooting-Connectivity.

1641. (2017), What You Should Know About the 'KRACK' WiFi Security Weakness, from https://krebsonsecurity.com/2017/10/what-you-should-know-about-the-krack-wifi-security-weakness/.

1642. Lily Hay Newman, (2017), The 'Secure' Wi-Fi Standard has a Huge, Dangerous Flaw, from https://www.wired.com/story/krack-wi-fi-wpa2-vulnerability/.

1643. Steve Tilson, (2017), WPA2 Key Reinstallation Attack (KRACK) Vulnerability Detection Dashboard, from https://www.tenable.com/sc-dashboards/wpa2-key-reinstallation-attack-krack-vulnerability-detection-dashboard.

1644. Thomas Brewster, (2017), Update Every Device -- This KRACK Hack Kills Your Wi-Fi Privacy, from https://www.forbes.com/sites/thomasbrewster/2017/10/16/krack-attack-breaks-wifi-encryption/#3d9b893e2ba9.

1645. Paul Ducklin, (2017), Wi-Fi at risk from KRACK attacks – here's what to do, from https://news.sophos.com/en-us/category/serious-security/.

1646. Charlie Osborne and Zack Whittaker, (2017), Here's every patch for KRACK Wi-Fi vulnerability available right now, from http://www.zdnet.com/article/here-is-every-patch-for-krack-wi-fi-attack-available-right-now/.

1647. Michael Heller, (2017), KRACK WPA2 flaw might be more hype than risk, from https://searchsecurity.techtarget.com/news/450428414/KRACK-WPA2-vulnerability-might-be-more-hype-than-risk.

1648. (2017), KRACK WPA Vulnerability - Key Reinstallation AttaCK, from https://aircrack-ng.blogspot.in/2017/10/krack-wpa-vulnerability-key.html.

1649. Attacks on EAP Protocols, from http://etutorials.org/Networking/Wireless+lan+security/Chapter+6.+Wireless+Vulnerabilities/Attacks+on+EAP+Protocols/.

1650. Mateusz Buczkowski, (2018), Introduction to Wi-Fi Security, from https://www.grandmetric.com/ended-wpa3-wi-fi-security-evolution/.

1651. Wireless Security Protocols: WEP, WPA, WPA2 and WPA3, from https://www.cyberpunk.rs/wireless-security-protocols-wep-wpa-wpa2-and-wpa3.

1652. Penny Hoelscher, (2018), What is WPA3, is it secure and should I use it?, from https://www.compantech.com/blog/information-security/what-is-wpa3/.

1653. Discover Wi-Fi Security, from https://www.wi-fi.org/discover-wi-fi/security.

1654. (2018), WPA3 Explained, from https://medium.com/@reliancegcs/wpa3-explained-wi-fi-is-getting-major-security-update-2b6dca8f3aff.

1655. (2020), Wi-Fi Protected Access, from https://en.wikipedia.org/wiki/Wi-Fi_Protected_Access.

1656. Mark Wyciślik-Wilson, aLTEr: Hackers can spy on your 4G browsing sessions thanks to LTE flaws, from https://betanews.com/2018/06/30/alter-lte-vulnerability/.

1657. (2018), aLTEr: POC Exploit of LTE Layer Two, from https://www.zimperium.com/blog/alter-poc-exploit-lte-layer-two/.

1658. Nicole Lorenz, (2018), aLTEr: Attack every smartphone via LTE, from https://www.avira.com/en/blog/alter-attack-every-smartphone-via-lte.

1659. Aaron Woland, (2018), Protecting iOS against the aLTEr attacks, from https://www.networkworld.com/article/3287149/protecting-ios-against-the-alter-attacks.html.

1660. (2013), Wireless Attacks Unleashed, from https://www.infosecinstitute.com/resources/hacking/wireless-attacks-unleashed/#gref.

1661. Elahe Fazeldehkordi and Oluwatobi Ayodeji Akanbi, (2010), Wormhole Attack, from https://www.sciencedirect.com/topics/computer-science/wormhole-attack.

1662. Julián Ramírez Gómez, Héctor Fernando Vargas Montoya, and Álvaro León Henao1, (2019), Implementing a Wormhole Attack on Wireless Sensor Networks with XBee S2C Devices, from https://dialnet.unirioja.es/descarga/articulo/6939721.pdf.

1663. (2015), Finding WPS enabled Wifi Networks with Kali Linux Wash, from https://www.hackingtutorials.org/wifi-hacking-tutorials/wps-wifi-networks-with-kali-linux-wash/?prettyPhoto.

1664. Brian Sak and Jiumudi Raghu Ram, (2016), Mastering Kali Linux Wireless Pentesting, Birmingham, Packt Publishing Ltd, from https://github.com/volym3ad/BOOKS/blob/master/Mastering%20Kali%20Linux%20Wireless%20Pentesting.pdf.

1665. (2008), ARP Spoofing, from https://openmaniak.com/ettercap_arp.php.

1666. Zaid Sabih, Creating Fake Access Points with the MANA Toolkit, from https://www.oreilly.com/library/view/learn-ethical-hacking/9781788622059/7b4ce7c7-1a2d-4dc6-8754-799ee9e4d7ba.xhtml.

1667. Arnav Tripathy, (2019), How to Make a Fake Access Point with Mana-Toolkit, from https://arnavtripathy98.medium.com/how-to-make-a-fake-access-point-with-mana-toolkit-2464c1843d1e.

1668. Balaji N, (2018), Wi-Jacking – New Wifi Attack Allow Accessing Millions of Neighbour's WiFi Without Cracking, from https://gbhackers.com/wi-jacking-wifi-attack/.

1669. Catalin Cimpanu, (2018), Google Fixes Chrome Issue that Allowed Theft of WiFi Logins, from https://www.zdnet.com/article/google-fixes-chrome-issue-that-allowed-theft-of-wifi-logins/.

1670. Gurubaran S, (2019), Pentesting & Crack WPA/WPA2 WiFi Passwords with Wifiphisher by Jamming the WiFi, from https://gbhackers.com/crack-wpawpa2-kali-linux-tutorial/.

1671. Kody, (2015), Get Anyone's Wi-Fi Password Without Cracking Using Wifiphisher, from https://null-byte.wonderhowto.com/how-to/hack-wi-fi-get-anyones-wi-fi-password-without-cracking-using-wifiphisher-0165154/.

1672. Tomáš Foltýn, (2019), WPA3 Flaws May Let Attackers Steal Wi-Fi Passwords, from https://www.welivesecurity.com/2019/04/11/wpa3-flaws-steal-wifi-passwords/.

1673. Catalin Cimpanu, (2019), Dragonblood vulnerabilities disclosed in WiFi WPA3 standard, from https://www.zdnet.com/article/dragonblood-vulnerabilities-disclosed-in-wifi-wpa3-standard/.

1674. Dan Goodin, (2019), Serious flaws leave WPA3 vulnerable to hacks that steal Wi-Fi passwords, from https://arstechnica.com/information-technology/2019/04/serious-flaws-leave-wpa3-vulnerable-to-hacks-that-steal-wi-fi-passwords/.

1675. Michael Peters, (2019), Dragonblood Vulnerabilities Discovered in WPA3 WiFi Standard, from https://securityboulevard.com/2019/04/dragonblood-vulnerabilities-discovered-in-wpa3-wifi-standard/.

1676. Sergiu Gatlan, (2019), WPA3 Wi-Fi Standard Affected by New Dragonblood Vulnerabilities, from https://www.bleepingcomputer.com/news/security/wpa3-wi-fi-standard-affected-by-new-dragonblood-vulnerabilities/.

1677. Pierluigi Paganini, (2019), WPA3 Attacks Allow Hackers to Hack Wi-Fi Password, from https://securityaffairs.co/wordpress/83653/hacking/wpa3-security-flaws.html.

1678. Daniele Antonioli, Nils Ole Tippenhauer, and Kasper B. Rasmussen, The KNOB is Broken: Exploiting Low Entropy in the Encryption Key Negotiation of Bluetooth BR/EDR, from https://www.usenix.org/conference/usenixsecurity19/presentation/antonioli.

1679. Doug Lynch, (2019), KNOB Attack exploits Bluetooth spec flaw to spy on device connections, from https://www.xda-developers.com/knob-attack-bluetooth-flaw/.

1680. Michael Heller, (2019), KNOB attack puts all Bluetooth devices at risk, from https://www.techtarget.com/searchsecurity/news/252468914/KNOB-attack-puts-all-Bluetooth-devices-at-risk.

1681. Daniele Antonioli, (2019), About the KNOB Attack, from https://knobattack.com.

1682. Mathy Vanhoef and Eyal Ronen, Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd, from https://papers.mathyvanhoef.com/dragonblood.pdf.

1683. PravinKarthik, (2021), Wireless InterChip Privilege Escalation Attack, from https://thecyberthrone.in/2021/12/27/wireless-interchip-privilege-escalation-attack/.

1684. Jiska Classen, Francesco Gringoli, Michael Hermann, and Matthias Hollick, (2021), Attacks on Wireless Coexistence: Exploiting Cross-Technology Performance Features for Inter-Chip Privilege Escalation, from https://www.x-mol.net/paper/article/1470351175419355136.

1685. GNSS SYSTEMS, from https://novatel.com/an-introduction-to-gnss/chapter-1-gnss-overview/section-1.

1686. (2016), A Survey and Analysis of the GNSS Spoofing Threat and Countermeasures, from https://www.researchgate.net/publication/301798786_A_Survey_and_Analysis_of_the_GNSS_Spoofing_Threat_and_Countermeasures.

1687. Mark L. Psiaki and Todd E. Humphreys, GNSS Spoofing and Detection, from https://radionavlab.ae.utexas.edu/images/stories/files/papers/gnss_spoofing_detection.pdf.

1688. Antonio De Maio, Global Navigation Satellite System GNSS Spoofing, from https://www.emsopedia.org/entries/global-navigation-satellite-system-gnss-spoofing/.

1689. (2018), 'ALTER' Attack Allows Hackers to Steal Data Via LTE, from https://blog.koddos.net/alter-attack-allows-hackers-to-steal-data-via-lte.

1690. (2020), Securing Wireless Networks, from https://www.cisa.gov/uscert/ncas/tips/ST05-003.

1691. Rajat Bhargava, (2019), The 4 Best Practices for WiFi Network Security – JumpCloud, from https://jumpcloud.com/blog/best-practices-for-wifi-security.

1692. Nitesh Malviya, (2021), Wireless Attacks and Mitigation, from https://www.infosecinstitute.com/resources/network-security-101/wireless-attacks-and-mitigation/.

1693. Developing Wireless Security Best Practices, from https://sourcedaddy.com/windows-xp/developing-wireless-security-best-practice.html.

1694. Monitored Bands, from https://nutsaboutnets.com/docs/rational-waves/monitored-bands/.

Module 17: Hacking Mobile Platforms

1695. Android framework for exploitation, from http://www.xysec.com/afe_manual.pdf.

1696. Sarah Perez, (2010), How to Hack Your Android Phone (and Why You Should Bother), from https://readwrite.com/how_to_hack_your_android_phone/.

1697. (2016), OWASP Mobile Top 10, from https://www.owasp.org/index.php/Projects/OWASP_Mobile_Security_Project_-_Top_Ten_Mobile_Risks.

1698. Security Threat Report 2014, from https://www.sophos.com/en-us/content/security-threat-report.

1699. wiseGEEK, What is Mobile Phone Spam?, from https://www.wisegeek.com/what-is-mobile-phone-spam.htm.

1700. Murugiah Souppaya and Karen Scarfone, (2013), Guidelines for Managing the Security of Mobile Devices in the Enterprise, from https://csrc.nist.gov/csrc/media/publications/sp/800-124/rev-1/final/documents/draft_sp800-124-rev1.pdf.

1701. Michael Cooney, (2012), 10 common mobile security problems to attack, from https://www.pcworld.com/article/2010278/10-common-mobile-security-problems-to-attack.html.

1702. Shruti Dhapola, (2014), Android is most hacked mobile OS: Here's how to protect your phone, from https://www.firstpost.com/tech/news-analysis/android-malware-increasing-tips-protect-phone-3647981.html.

1703. Network Spoofer - Hacking networks from Android, from http://www.hackedexistence.com/project-networkspoofer.html.

1704. iOS jailbreaking, From https://en.wikipedia.org/wiki/IOS_jailbreaking#Types_of_jailbreaks.

1705. Jeff Benjamin, (2011), Untethered Jailbreak vs. Tethered Jailbreak vs. SemiTethered Jailbreak — What's the Difference?, from https://www.idownloadblog.com/2011/10/22/untethered-jailbreak-vs-tethered-jailbreak-vs-semitethered-jailbreak/.

1706. Jailbreak Your iPhone, iPad And iPod Touch, from https://www.iphoneinformer.com/jailbreak/.

1707. Lisa Phifer, (2013), BYOD security strategies: Balancing BYOD risks and rewards, from https://www.techtarget.com/searchsecurity/features.

1708. Stephan Chenette, (2013), Building Custom Android Malware for Penetration Testing, from https://www.slideshare.net/slideshow/building-custom-android-malware-brucon-2013/26636208.

1709. Al Berg, Best practices for protecting handhelds from mobile malware, from https://searchsecurity.techtarget.com/tip/Best-practices-for-protecting-handhelds-from-mobile-malware.

1710. Sam Bakken, (2017), Defense in Depth: A Layered Approach to Mobile Security with MDM, MAM & Mobile App Vetting, from https://www.nowsecure.com/blog/2017/12/12/defense-in-depth-a-layered-approach-to-mobile-security-with-mdm-mam-mobile-app-vetting/.

1711. (2017), Anatomy of an Android, from https://www.sophos.com/en-us/medialibrary/PDFs/other/sophos-anatomy-of-an-android-infographic.pdf.

1712. Pierluigi Paganini, (2016), Researchers hack WhatsApp accounts through SS7 protocol, from https://securityaffairs.co/wordpress/47179/hacking/hacking-ss7-protocol.html.

1713. Samuel Gibbs, (2016), SS7 hack explained: what can you do about it?, from https://www.theguardian.com/technology/2016/apr/19/ss7-hack-explained-mobile-phone-vulnerability-snooping-texts-calls.

1714. Secure your network from SS7 attacks, from https://www.sinch.com/products/telco/signaling-firewall/?cn-reloaded=1.

1715. Simjacker, from https://www.enea.com/info/simjacker/.

1716. Shouvik Das, (2019), Your Data, Location Might be Tracked with This SIM Card Flaw, Without Your Knowledge, from https://www.news18.com/news/tech/your-data-location-might-be-tracked-with-this-sim-card-flaw-without-your-knowledge-2306879.html.

1717. Mohit Kumar, (2019), New SIM Card Flaw Lets Hackers Hijack Any Phone Just by Sending SMS, from https://thehackernews.com/2019/09/simjacker-mobile-hacking.html.

1718. (2019), Device Administration Overview, from https://developer.android.com/work/device-admin.

1719. Ashraf Iftekhar, (2018), How To Test Android Application Security Using Drozer?, from https://medium.com/@ashrafrizvi3006/how-to-test-android-application-security-using-drozer-edc002c5dcac.

1720. Sumit Bhattacharya, Android Penetration Tools Walkthrough Series: Drozer, from https://www.infosecinstitute.com/resources/penetration-testing/android-penetration-tools-walkthrough-series-drozer/#gref.

1721. Romansh Yadav, (2019), Drozer! The Game Changer Tool for Android Pen Testing, from https://blog.securelayer7.net/drozer-the-game-changer-tool-for-android-pentesting/.

1722. Man-in-the-Disk: A New Attack Surface for Android Apps, from https://blog.checkpoint.com/security/man-in-the-disk-a-new-attack-surface-for-android-apps/.

1723. Margaret Rouse, (2019), Man-in-the-Disk (MITD) Attack, from https://www.techtarget.com/iotagenda/definition/man-in-the-middle-attack-MitM.

1724. Mohammed Tahir, (2019), Latest attack on Android known as "Spearphone Attack", from https://cyberops.in/blog/spearphone-attack/.

1725. Swati Khandelwal, (2019), New Attack Lets Android Apps Capture Loudspeaker Data Without Any Permission, from https://thehackernews.com/2019/07/android-side-channel-attacks.html.

1726. Abeerah Hashim, (2019), Spearphone Attack Allows Android Apps to Listen to Your Loudspeaker Conversations, from https://latesthackingnews.com/2019/07/20/spearphone-attack-allows-android-apps-to-listen-to-your-loudspeaker-conversations/.

1727. Connor Jones, (2019), Android phones vulnerable to advanced SMS phishing attacks, from https://www.itpro.co.uk/security/34334/android-phones-vulnerable-to-advanced-sms-phishing-attacks.

1728. Ravie Lakshmanan, (2019), Hackers are now attacking Android users with advanced SMS phishing techniques, from https://thenextweb.com/news/hackers-are-now-attacking-android-users-with-advanced-sms-phishing-techniques.

1729. Aman Bhardwaj, (2019), SSL Pinning: Introduction & Bypass for Android, from https://niiconsulting.com/checkmate/2019/04/ssl-pinning-introduction-bypass-for-android/.

1730. Balaji N, (2019), Bypassing and Disabling SSL Pinning on Android to Perform Man-in-the-Middle Attack, from https://gbhackers.com/bypass-ssl-pinning/.

1731. Gurubaran S, (2019), Newly Discovered Tap 'n Ghost Attack Let Hackers to Remotely Control Android Smartphones, from https://gbhackers.com/tap-n-ghost-attack-remotely-smartphones/

1732. (2019), Newly discovered Tap 'n Ghost attack can be used to target Android devices, from https://cyware.com/news/newly-discovered-tap-n-ghost-attack-can-be-used-to-target-android-devices-fc81c323.

1733. Sven Taylor, (2018), How to Secure Your Android Device in 5 Simple Steps, from https://restoreprivacy.com/secure-android-device/.

1734. Michael Simon, (2019), How to Secure, Protect, and Completely Lock Down Your Android Phone, from https://www.pcworld.com/article/403176/secure-android-phone.html.

1735. Steven J. and Vaughan-Nichols, (2018), The 10 best ways to secure your Android phone, from https://www.zdnet.com/article/the-ten-best-ways-to-secure-your-android-phone/.

1736. (2020), iOS jailbreaking, from https://en.wikipedia.org/wiki/IOS_jailbreaking#Types_of_jailbreaks.

1737. Roy Iarchy, (2018), iOS Trustjacking – A Dangerous New iOS Vulnerability, from https://symantec-enterprise-blogs.security.com/feature-stories/ios-trustjacking-dangerous-new-ios-vulnerability.

1738. Joshua Long, (2018), iOS Trustjacking: How Attackers can Hijack Your iPhone, from https://www.intego.com/mac-security-blog/ios-trustjacking-how-attackers-can-hijack-your-phone/.

1739. (2018), iOS Trustjacking Protection with EMM, from https://arsenb.wordpress.com/2018/04/25/ios-trustjacking-protection-with-emm/.

1740. Lewis Painter, (2019), iPhone Security Tips: How to Protect Your Phone from Hackers, from https://www.macworld.com/article/668652/iphone-security-tips-how-to-protect-your-phone-from-hackers.html.

1741. (2019), 5 Easy Ways to Protect Your iPhone and Privacy in 2020 FREE, from https://www.vpnmentor.com/blog/protect-privacy-iphone/.

1742. Ken Hess, (2014), 10 BYOD policy guidelines for a secure work environment, from https://techtalk.gfi.com/10-byod-policy-guidelines-for-a-secure-work-environment/.

1743. OWASP Mobile Top 10, from https://owasp.org/www-project-mobile-top-10/#tab=Top_10_Mobile_Controls.

1744. Tampering and Reverse Engineering, from https://mobile-security.gitbook.io/mobile-security-testing-guide/general-mobile-app-testing-guide/0x04c-tampering-and-reverse-engineering.

1745. Srinivas, Introduction to Reverse Engineering, from https://www.infosecinstitute.com/resources/reverse-engineering/android-hacking-and-security-part-18-introduction-to-reverse-engineering/#gref.

1746. (2021), How Does One Time Password Hijacking Work?, from https://www.teampassword.com/blog/how-does-one-time-password-hijacking-work#:~:text=OTP%20via%20SMS%20Hijacking,has%20several%20serious%20security%20drawbacks.&text=While%20many%20services%20offer%20password,restrict%20access%20of%20the%20account.

1747. Chris Kalis, (2022), The Rising Risk of OTP Hijacking and SIM Swap Attacks and How Behavioral Biometrics Helps Thwart These Attacks, from https://securityboulevard.com/2022/01/the-rising-risk-of-otp-hijacking-sim-swap-attacks-and-how-behavioral-biometrics-helps-thwart-these-attacks/.

1748. (2020), Don't Use SMS For 2FA. Here is Why, from https://www.finextra.com/blogposting/18645/dont-use-sms-for-2fa-here-is-why#:~:text=Studies%20are%20finding%20that%20the,the%20cell%20phone%20service%20provider.

1749. (2021), How to Guard Against Webcam Attacks, from https://home.sophos.com/en-us/security-news/2021/webcam-attacks.

1750. David Cook, (2020), Hackers can Access Your Mobile and Laptop Cameras and Record You – Cover Them Up Now, from https://theconversation.com/hackers-can-access-your-mobile-and-laptop-cameras-and-record-you-cover-them-up-now-135933.

1751. Pierluigi Paganini, (2019), CVE-2019-2234 Flaws in Android Camera Apps Exposed Millions of Users Surveillance, from https://securityaffairs.co/wordpress/94089/hacking/cve-2019-2234-android-camera-apps-flaws.html.

1752. Mic Johnson, (2021), What is Camfecting and What Can You Do About It?, from https://latesthackingnews.com/2021/12/09/what-is-camfecting-and-what-can-you-do-about-it/.

1753. Egle Juodyte, (2022), How to Tell If Your Laptop Camera Has Been Hacked, from https://nordvpn.com/blog/tell-if-laptop-camera-hacked/.

1754. Erez Yalon, (2019), How Attackers Could Hijack Your Android Camera to Spy on You, from https://checkmarx.com/blog/how-attackers-could-hijack-your-android-camera/.

1755. (2019). A new Android vulnerability (CVE-2019-2234) allows attackers to hijack Camera App, from https://www.andreafortuna.org/2019/11/22/a-new-android-vulnerability-cve-2019-2234-allows-attackers-to-hijack-camera-app/.

1756. Lindsey O'Donnell, (2019), Google Discloses Android Camera Hijack Hack, from https://threatpost.com/google-android-camera-hijack-hack/150409/.

1757. (2019). Android Vulnerability Allows Hackers to Access Camera (CVE-2019-2234), from https://www.defenxor.com/blog/android-vulnerability-allows-hackers-to-access-camera-cve-2019-2234/.

1758. Cecilia Duong, (2021), Camfecting: How Hackers Attack By Gaining Access to Your Webcam, from https://www.unsw.edu.au/newsroom/news/2021/10/camfecting--how-hackers-attack-by-gaining-access-to-your-webcam.

1759. (2021). Metasploit Basics, Part 13: Exploiting Android Mobile Devices (Updated), from https://www.hackers-arise.com/post/2018/07/06/metasploit-basics-part-13-exploiting-android-mobile-devices.

1760. Irfan Shakeel, (2020), Hacking Android Phone Remotely Using Metesploit, from https://irfanshakeel.medium.com/hacking-android-phone-remotely-using-metasploit-43ccf0fbe9b8.

1761. Irfan Shakeel, (2020), How to Hack an Android Phone Using Metasploit Msfvenom in Kali Linux, from https://opswatacademy.com/.

1762. Priyanshu Sahay, (2019), iOS Penetration Testing- Cycript A Runtime Manipulation- Part 2, from https://hackersonlineclub.com/ios-penetration-testing-cycript-a-runtime-manipulation/.

1763. Damian Malarczyk, (2019), Swift Native method swizzling, from https://www.guardsquare.com/blog/swift-native-method-swizzling.

1764. Abhimuralidharan, (2017). Method swizzling in iOS swift, from https://abhimuralidharan.medium.com/method-swizzling-in-ios-swift-1f38edaf984f.

1765. (2022). Keychain Data, from https://ios.pentestglobal.com/file-system/keychain-data.

1766. Allyson O'Malley, (2018). iOS Pentesting Tools Part 3: Frida and Objection, from https://www.allysonomalley.com/2018/12/20/ios-pentesting-tools-part-3-frida-and-objection/.

1767. (2022). Data Storage on Android, from https://mobile-security.gitbook.io/mobile-security-testing-guide/android-testing-guide/0x05d-testing-data-storage.

1768. (2022). Android Keystore System, from https://developer.android.com/privacy-and-security/keystore.

1769. (2021). Keychain Data Protection, from https://support.apple.com/en-in/guide/security/secb0694df1a/web.

1770. Chris Brook, (2020), The Ultimate Guide to BYOD Security: Overcoming Challenges, Creating Effective Policies, and Mitigating Risks to Maximize Benefits, from https://digitalguardian.com/blog/ultimate-guide-byod-security-overcoming-challenges-creating-effective-policies-and-mitigating.

1771. Lyle Del Vecchio, Bring Your Own Device (BYOD) Security Best Practices, from https://planergy.com/blog/byod-security-best-practices/.

1772. BYOD Security: Threats, Security Measures and Best Practices, from https://www.hysolate.com/learn/byod/byod-security-threats-security-measures-and-best-practices/.

1773. (2022). Objection Tutorial, from https://book.hacktricks.xyz/mobile-apps-pentesting/android-app-pentesting/frida-tutorial/objection-tutorial.

1774. (2020). How to Hook Android Native Methods With Frida, from https://prevul.com/blog/how-hook-android-native-methods-frida-noob-friendly.

1775. 5 Mobile Security Threats You Can Protect Yourself From, from https://us.norton.com/blog/mobile/types-of-common-mobile-threats-and-what-they-can-do-to-your-phone.

1776. Michael Holmes, (2021), 12 Ways to Protect Your Smartphone from Cyber Attacks, from https://smallbiztrends.com/2013/01/protect-smartphone-cyber-attack.html.

1777. Jovi Umawing, (2018), 10 Ways to Protect Your Android Phone, from https://blog.malwarebytes.com/101/2018/03/10-ways-to-protect-your-android-phone/.

1778. Sia Smith, (2022), How To Secure Your iOS Device in 2022, from https://www.uplarn.com/how-to-secure-your-ios-device.

1779. Brandon Vigliarolo, (2020), How to Protect Your Privacy on an iOS Device, from https://www.techrepublic.com/topic/apple/.

1780. Learn About Privacy Settings and Controls., from https://www.apple.com/privacy/control.

1781. (2021). Encryption and Data Protection Overview, from https://support.apple.com/en-in/guide/security/sece3bee0835/1/web/1.

1782. Securing Your iOS Device, from https://oit.ncsu.edu/it-security/mobile/ios/device.

1783. (2022). Mobile Device Security Guidelines, from https://www.technology.pitt.edu/mobile-device-security-guideline.

1784. Mobile Device Security and Usage Guideline, from https://www.cmu.edu/iso/governance/guidelines/mobile-device.html.

1785. John Powers, (2021), The Ultimate Guide to Mobile Device Security in the Workplace, from https://www.techtarget.com/searchmobilecomputing/The-ultimate-guide-to-mobile-device-security-in-the-workplace.

1786. (2022). Mobile Security, from https://en.wikipedia.org/wiki/Mobile_security#Countermeasures.

1787. (2023). IT threat evolution in Q3 2023. Mobile statistics, from https://securelist.com/it-threat-evolution-q3-2023-mobile-statistics/111224/.

1788. (2024). The mobile malware threat landscape in 2023, from https://securelist.com/mobile-malware-report-2023/111964/.

1789. Anand Chandrashaker, Why Hackers Love Your Mobile Devices and What You Can Do About It, from https://www.infosysbpm.com/blogs/business-transformation/key-reasons-why-hackers-target-mobile-devices-and-how-to-stop-them.html#:~:text=A%20mobile%20device%20used%20for,find%20data%20and%20download%20it.

1790. Android Architecture, from https://mas.owasp.org/MASTG/Android/0x05a-Platform-Overview/#android-architecture.

1791. Device administration overview, from https://developer.android.com/work/device-admin#:~:text=Policies%20supported%20by%20the%20Device%20Administration%20API,&text=Requires%20that%20devices%20ask%20for%20PIN%20or%20passwords,&text=Set%20the%20required%20number%20of,have%20at%20least%20six%20characters.

1792. Joseph, (2023), How to FRP Bypass Lock on Any Android Phone: Easy Guide, from https://cellularnews.com/guides/how-to-frp-bypass-lock-on-any-android-phone-easy-guide/#google_vignette.

1793. Jerry Cook, (2024), D&G Password Unlocker: Latest Review, Free Download & Tutorial, from https://www.ultfone.com/unlock-android/dg-password-unlocker.html.

1794. (2022). Android app for network / vulnerability scanning: zANTI | mobile penetration testing, from https://www.youtube.com/watch?v=Q2Hfde54HTM&ab_channel=ClicksandBits.

1795. (2024), MASTG-TECH-0100: Logging Sensitive Data from Network Traffic, from https://mas.owasp.org/MASTG/techniques/android/MASTG-TECH-0100/.

1796. Jessica Molden, (2024), Android Banking Trojan Strikes as Fake Google Chrome Browser, Threatening Users' Financial Security, from https://www.pcmatic.com/blog/moment-android-banking-trojan-strikes-as-fake-google-chrome-browser-threatening-users-financial-security/.

1797. Ivan Nikolskiy, (2024), SeaShell: iOS 16/17 Remote Access, from https://medium.com/@entys080/seashell-ios-16-17-remote-access-41cc3366019d#:~:text=This%20vulnerability%20in%20CoreTrust%2C%20designated,again%20in%20the%20final%20release.

1798. (2023). SeaShell - iOS 16/17 Remote Access Fresh iOS post-exploitation tool, from https://blog.entysec.com/2023-12-31-seashell-ios-malware/.

1799. MASTG-TECH-0052: Accessing the Device Shell, from https://mas.owasp.org/MASTG/techniques/ios/MASTG-TECH-0052/#remote-shell.

1800. Pieter Arntz, (2024), GoldPickaxe Trojan steals your face!, from https://www.malwarebytes.com/blog/news/2024/01/goldpickaxe-trojan-steals-your-face.

1801. (2024), Chinese-Linked LightSpy iOS Spyware Targets South Asian iPhone Users, from https://thehackernews.com/2024/04/chinese-linked-lightspy-ios-spyware.html.

1802. (2024), About Apple threat notifications and protecting against mercenary spyware, from https://support.apple.com/en-in/102174.

1803. Laura French, (2024), New macOS malware SpectralBlur ID'd as North Korean backdoor, from https://www.scmagazine.com/news/new-macos-malware-spectralblur-ides-as-north-korean-backdoor.

1804. Stephen Shankland, (2022), Pegasus Spyware and Citizen Surveillance: Here's What You Should Know, from https://www.cnet.com/tech/mobile/pegasus-spyware-and-citizen-surveillance-what-you-need-to-know/.

Module 18: IoT and OT Hacking

1805. Margaret Rouse, (2016), Internet of Things (IoT), from https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT.

1806. Bernadette Johnson, How the Internet of Things Works, from https://computer.howstuffworks.com/internet-of-things.htm.

1807. (2016), The Pros and Cons of IoT, from http://www.humavox.com/blog/pros-cons-iot/.

1808. (2015), How IoT Works – An Overview of the Technology Architecture, from https://www.embitel.com/blog/embedded-blog/how-iot-works-an-overview-of-the-technology-architecture-2.

1809. Internet of Things: Explained, From http://www.carritech.com/news/internet-of-things/.

1810. Dr. Gaurav Bajpai, Middleware for Internet of Things, from http://wireless.ictp.it/rwanda_2015/presentations/Middleware_IoT.pdf.

1811. M2M/IoT Sector Map, from http://www.beechamresearch.com/article.aspx?id=4.

1812. Vasanth Ganesan, (2016), Video meets the Internet of Things, from https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/video-meets-the-internet-of-things.

1813. (2015), 11 Internet of Things (IoT) Protocols You Need to Know About, from https://www.rs-online.com/designspark/eleven-internet-of-things-iot-protocols-you-need-to-know-about.

1814. Sean Schneider, (2013), Understanding The Protocols Behind The Internet Of Things, from https://www.electronicdesign.com/technologies/iot/article/21798493/understanding-the-protocols-behind-the-internet-of-things.

1815. Internet of things, from https://en.wikipedia.org/wiki/Internet_of_things#Trends_and_characteristics.

1816. Anupama Kaushik, (2016), IOT-An Overview, from https://www.ijarcce.com/upload/2016/march-16/IJARCCE%20264.pdf.

1817. Karen Rose, Scott Eldridge, Lyman Chapin, (2015), The Internet of Things: An Overview, from https://www.internetsociety.org/wp-content/uploads/2017/08/ISOC-IoT-Overview-20151221-en.pdf.

1818. Bruce Byfield, (2016), The Internet of Things: 7 Challenges, from https://www.datamation.com/data-center/the-internet-of-things-7-challenges.html.

1819. Aritra Sarkhel, (2016), 5 challenges to Internet of Things, from https://tech.economictimes.indiatimes.com/news/internet/5-challenges-to-internet-of-things/52700940.

1820. Robbie Mitchell, (2015), 5 challenges of the Internet of Things, from https://blog.apnic.net/2015/10/20/5-challenges-of-the-internet-of-things/.

1821. Azita Esmaili, (2015), The opportunities and threats of the Internet of Things, from https://www.techworld.com/startups/opportunities-threats-of-internet-of-things-3598417/.

1822. Charlie Ashton, (2015), Is IoT a Threat or an Opportunity for Service Providers?, from https://www.sdxcentral.com/articles/contributed/iot-threat-opportunity-service-providers-charlie-ashton/2015/06/.

1823. Avantika Monnappa, (2018), TOGAF and the Internet of Things, from https://www.simplilearn.com/togaf-applications-in-internet-of-things-iot-article.

1824. Tessel Renzenbrink, (2014), Internet of Things Poses an Unprecedented Privacy Risk, from https://www.elektormagazine.com/articles/internet-of-things-poses-an-unprecedented-privacy-risk.

1825. (2016), Top IoT Vulnerabilities, from https://www.owasp.org/index.php/Top_IoT_Vulnerabilities.

1826. (2015), IoT Attack Surface Areas, from https://www.owasp.org/index.php/IoT_Attack_Surface_Areas.

1827. Jaibeer Malik, (2017), IOT Security: Attack surface areas, Vulnerabilities & Considerations, from https://jaibeermalik.wordpress.com/2017/01/05/iot-security-attack-surface-areas-vulnerabilities-considerations/.

1828. Security for IoT, from https://www.infineon.com/cms/en/applications/smart-card-and-security/iot-security/?redirId=59655#!overview.

1829. Masato Terada, Naoko, and Naoko Ohnishi, (2017), HIRT-PUB16003: Cyber-attacks Using IoT Devices, https://www.hitachi.com/hirt/publications/hirt-pub16003/index.html.

1830. APNIC, (2017), IoT - the Next Wave of DDoS Threat Landscape, from https://www.slideshare.net/apnic/iot-the-next-wave-of-ddos-threat-landscape?qid=b1d633e5-2d40-4151-b3ec-91d93be094e9&v=&b=&from_search=6.

1831. Jaikumar Vijayan, (2014), Target attack shows danger of remotely accessible HVAC systems, from https://www.computerworld.com/article/2487452/cybercrime-hacking/target-attack-shows-danger-of-remotely-accessible-hvac-systems.html.

1832. Paul Roberts, (2012), FBI Issued Alert over July Attack on HVAC System, from https://securityledger.com/2012/12/fbi-issued-alert-over-july-attack-on-hvac-system/.

1833. Erez Metula, (2016), Hacking The IoT (Internet of Things) - PenTesting RF Operated Devices, from https://www.owasp.org/images/2/29/AppSecIL2016_HackingTheIoT-PenTestingRFDevices_ErezMetula.pdf.

1834. Jerry Hildenbrand, (2017), Let's talk about Blueborne, the latest Bluetooth vulnerability, from https://www.androidcentral.com/lets-talk-about-blueborne-latest-bluetooth-vulnerability.

1835. (2017), The Attack Vector "BlueBorne" Exposes Almost Every Connected Device, from https://www.armis.com/blueborne/.

1836. Blueborne Attack Threatens IoT Devices, from https://www.pindrop.com/blog/blueborne-attack-threatens-iot-devices/.

1837. Kim Zetter, (2016), Inside the Cunning, Unprecedented Hack of Ukraine's Power Gnd, from https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/.

1838. Taylor Edginton, Find the IP Address of any IoT Device, from https://www.pingplotter.com/wisdom/article/finding-my-smart-outlet.html.

1839. (2017) Hacking Zigbee Devices, from https://blog.attify.com/hack-iot-devices-zigbee-sniffing-exploitation/.

1840. (2016), IoT Devices Easily Hacked to be Backdoors: Experiment, from https://www.securityweek.com/iot-devices-easily-hacked-be-backdoors-experiment.

1841. Scott Craig, (2016), Telnet: An Attacker's Gateway to the IoT, from https://securityintelligence.com/telnet-an-attackers-gateway-to-the-iot.

1842. (2016), Emulating and Exploiting Firmware binaries – Offensive IoT Exploitation series, from https://www.infosecinstitute.com/resources/hacking/emulating-and-exploiting-firmware-binaries-offensive-iot-exploitation-series/.

1843. OWASP Top 10 Internet of Things Vulnerability Categories, from https://www.owasp.org/images/8/8e/Infographic-v1.jpg.

1844. (2016), IoT Framework Assessment, from https://www.owasp.org/index.php/IoT_Framework_Assessment.

1845. (2019), MQTT v5.0 now an official OASIS standard, from http://mqtt.org/.

1846. (2020). MQTT. from https://en.wikipedia.org/wiki/MQTT.

1847. (2020). Constrained Application Protocol, from https://en.wikipedia.org/wiki/Constrained_Application_Protocol.

1848. CoAP. from https://coap.technology/.

1849. (2019). 6LoWPAN. from https://en.wikipedia.org/wiki/6LoWPAN.

1850. (2020). Edge computing. from https://en.wikipedia.org/wiki/Edge_computing.

1851. (2020). ANT (network). from https://en.wikipedia.org/wiki/ANT_(network).

1852. (2019). OMA LWM2M. From https://en.wikipedia.org/wiki/OMA_LWM2M.

1853. (2020). IoT Standards and Protocols, from https://www.postscapes.com/internet-of-things-protocols/.

1854. (2020). Google Fuchsia, from https://en.wikipedia.org/wiki/Google_Fuchsia.

1855. Rita Sharma, Top 10 Challenges Enterprises Face in IoT Implementation, from https://www.finoit.com/blog/enterprise-challenges-in-iot/.

1856. OWASP Internet of Things, from https://owasp.org/www-project-internet-of-things/#tab=IoT_Attack_Surface_Areas.

1857. Internet of Things (IoT) Threats, from https://appsec-labs.com/iot_threats/#toggle-id-5.

1858. (2019). IoT Application Security Challenges and Solutions, from https://www.iotforall.com/iot-application-security/.

1859. Ryan Kh, (2018), Assessing the Severity of SQL Injection Threats to IoT Security, from https://www.smartdatacollective.com/assessing-severity-sql-injection-threats-iot-security/.

1860. Anand Srinivasan, (2017), Understanding SDR-Based Attacks on IoT, from https://dataflog.com/read/understanding-sdr-based-attacks-on-iot/3735.

1861. Nitesh Malviya, IoT Radio Communication Attack, from https://www.infosecinstitute.com/resources/hacking/iot-radio-communication-attack/#gref.

1862. Robert Keim, (2017), Introduction to Software-Defined Radio, from https://www.allaboutcircuits.com/technical-articles/introduction-to-software-defined-radio/.

1863. Rene Millman, (2018), Hackers Could Use Web-based Attacks to Take Over IoT Devices, from https://internetofbusiness.com/hackers-could-use-web-based-attacks-to-take-over-iot-devices/.

1864. (2019), Discovering and Hacking IoT Devices Using Web-Based Attacks, from https://www.invicti.com/blog/web-security/discovering-hacking-iot-devices-using-web-based-attacks/.

1865. Gunes Acar, Danny Y. Huang, Frank Li, Arvind Narayanan, and Nick Feamster, (2018). Fast Web-based Attacks to Discover and Control IoT Devices, from https://freedom-to-tinker.com/2018/06/21/fast-web-based-attacks-to-discover-and-control-iot-devices/.

1866. Gunes Acar, Danny Huang, Frank Li, Arvind Narayanan, and Nick Feamster, Web-based Attacks on Local IoT Devices, from https://conferences.sigcomm.org/sigcomm/2018/files/slides/iot/paper_3.1.pdf.

1867. Margaret Rouse, (2008), DNS Rebinding Attack, from https://www.techtarget.com/searchsecurity/definition/exploit.

1868. Kobus Marneweck, (2019), The Role of Physical Security in IoT, from https://community.arm.com/arm-community-blogs/b/internet-of-things-blog/posts/the-role-of-physical-security-in-iot.

1869. Shivam Bhasin and Debdeep Mukhopadhyay, (2016), Fault Injection Attacks, from https://pdfs.semanticscholar.org/0ae1/a6e055383e64011fa639e42f9294d11c3639.pdf.

1870. Hezam Akram Abdul-Ghani, Dimitri Konstantas, and Mohammed Mahyoub, (2018), A Comprehensive IoT Attacks Survey Based on a Building-blocked Reference Model, from https://thesai.org/Downloads/Volume9No3/Paper_49-A_Comprehensive_IoT_Attacks_Survey.pdf.

1871. (2019). 2019 Internet Security Threat Report, from https://www.broadcom.com/support/security-center.

1872. Karen Taylor, Mark Steedman, Amen Sanghera, and Matthew Thanner, (2018), Medtech and the Internet of Medical Things, from https://www2.deloitte.com/content/dam/Deloitte/global/Documents/Life-Sciences-Health-Care/gx-lshc-medtech-iomt-brochure.pdf.

1873. Dr Leonie Maria Tanczer, Dr Ine Steenmans, Dr Irina Brass, and Dr Madeline Carr, (2018), Networked World Risks and Opportunities in the Internet of Things, from https://discovery.ucl.ac.uk/id/eprint/10063068/1/InterconnectedWorld2018.pdf.

1874. Cyrill Brunschwiler, (2016), Software Defined Radio (SDR) and Decoding On-off Keying (OOK), from https://blog.compass-security.com/2016/09/software-defied-radio-sdr-and-decoding-on-off-keying-ook/.

1875. Aaron Guzman and Aditya Gupta, Defining Firmware Analysis Methodology, from https://play.google.com/books/reader?id=rEFPDwAAQBAJ&hl=en&pg=GBS.PA73.w.2.0.0.

1876. OWASP Internet of Things, from https://owasp.org/www-project-internet-of-things/#Things_to_check_for_once_the_file_system_is_mounted_or_extracted.

1877. Industrial IoT: Threats and Countermeasures, from https://www.rambus.com/iot/industrial-iot/.

1878. Internet of Things (IoT) security: 9 ways you can help protect yourself, from https://us.norton.com/blog/iot/what-is-the-internet-of-things.

1879. Cujo AI, (2018), Five Key Security Tips to Avoid an IoT Hack, from https://www.helpnetsecurity.com/2018/08/14/avoid-iot-hack/.

1880. Common Attacks on IoT Devices, from https://elinux.org/images/f/f8/Common-Attacks-on-IoT-Devices-Christina-Quast.pdf.

1881. Lyndon Sutherland, (2017), The Weaponization of IoT Devices, from https://www.ibm.com/downloads/cas/6MLEALKV.

1882. Jeff Day, Roger Shepherd, Paul Kearney and Richard Storer, (2018), Best Practice Guides, from https://www.iotsecurityfoundation.org/wp-content/uploads/2019/03/Best-Practice-Guides-Release-1.2.1.pdf.

1883. OWASP's Top 10 IoT Vulnerabilities, from https://www.deviceauthority.com/blog/owasp-s-top-10-iot-vulnerabilities.

1884. Fredric Paul, (2019), Top 10 IoT vulnerabilities, from https://www.networkworld.com/article/3332032/top-10-iot-vulnerabilities.html.

1885. (2016), IoT Framework Assessment, from https://www.owasp.org/index.php/IoT_Framework_Assessment.

1886. Calum McClelland, (2019), IoT Device Management: What is it and Why Do You Need it?, from https://www.iotforall.com/what-is-iot-device-management/.

1887. (2019), Using Oracle Internet of Things Asset Monitoring Cloud Service, from https://docs.oracle.com/en/cloud/saas/iot-asset-cloud/iotaa/operations-center.html#GUID-25C1A99C-85OD-40E6-864D-04CC6CE295C4.

1888. (2020), Operational Technology, from https://en.wikipedia.org/wiki/Operational_Technology.

1889. Lauren Horwitz, OT networks and IT networks are closely intertwined, from https://www.cisco.com/c/en/us/products/security/ot-networks.html.

1890. Operational Technology (OT) – Definitions and Differences with IT, from https://www.i-scoop.eu/industry-4-0/operational-technology-ot/.

1891. Graham Williamson, (2015), OT, ICS, SCADA – What's the difference?, from https://www.kyrgingercole.com/blog/williamson/ot-ics-scada-whats-the-difference.

1892. OT Definition - Operational Technology, from http://www.commtunnel.com/ot-definition/.

1893. About Industrial Networks, from https://www.oreilly.com/library/view/industrial-network-security/9780124201149/B9780124201149000022/B9780124201149000022.xhtml#B9780124201149000022.

1894. Mohamed Babikir, (2018), Convergence of IT and OT in Energy and Manufacturing, from https://www.digitalistmag.com/cio-knowledge/2018/11/05/convergence-of-it-ot-in-energy-manufacturing-06192743.

1895. Tim Sowell, (2015), OT/IT Convergence "What does it mean in the Industrial World?", from http://operationalevolution.blogspot.com/2015/02/otit-convergence-what-does-it-mean-in.html.

1896. Bridging the Gap Between Operational Technology and Information Technology, from https://www.avnet.com/wps/wcm/connect/onesite/90fb0b8d-33a4-4039-97De-91bea619456f/pa-eurotechot-it-whitepaper-inc0360043-0416-en.pdf?MOD=AJPERES&CVID=lFRKUrv&id=1489688438797.

1897. Beginners: What is Industrial IoT (IIoT), from https://www.youtube.com/watch?v=6MNDxRU3yzE.

1898. The Purdue model for industrial control systems, from https://subscription.packtpub.com/book/networking_and_servers/9781788395168/1/ch01lvl1sec10/the-purdue-model-for-industrial-control-systems.

1899. (2019), Blueprint for Securing Industrial Control Systems, from https://www.checkpoint.com/downloads/products/cp-industrial-control-ics-security-blueprint.pdf.

1900. Ethernet-to-the-Factory 1.2 Design and Implementation Guide, from https://www.cisco.com/c/en/us/td/docs/solutions/Verticals/EttF/EttFDIG/ch2_EttF.html.

1901. Rick Peters, (2019), Key Findings on the State of Operational Technology and Cybersecurity, from https://www.csoonline.com/article/3392579/key-findings-on-the-state-of-operational-technology-and-cybersecurity.html#:~:targetText=Cybersecurity%20Risks%20for%20Operational%20Technology&targetText=The%20most%20common%20types%20of,spyware%2C%20and%20mobile%20security%20breaches.

1902. (2018), Identifying the Risks to Operational Technology, from https://www.horn-it.com/2018/09/identifying-the-risks-to-operational-technology/.

1903. (2018), Why a Unified Approach to IT and OT Network Security is Critical, from https://www.infosecurityeurope.com/en-gb.html?vs=636628570667070000.

1904. (2019), The IIoT Attack Surface: Threats and Security Solutions, from https://www.trendmicro.com/vinfo/us/security/news/internet-of-things/-the-iot-attack-surface-threats-and-security-solutions.

1905. Keith Stouffer, Victoria Pillitteri, Suzanne Lightman, Marshall Abrams, and Adam Hahn, (2015), Guide to Industrial Control Systems (ICS) Security, from https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-82r2.pdf.

1906. Industrial Control System, from https://www.trendmicro.com/vinfo/in/security/definition/industrial-control-system.

1907. Arthur Gervais, (2012), Security Analysis of Industrial Control Systems, from http://nordsecmob.aalto.fi/en/publications/theses_2012/gervais-arthur_thesis.pdf.

1908. Pascal Ackerman, The industrial control system architecture, from https://www.oreilly.com/library/view/industrial-cybersecurity/9781788395151/4e1de95d-dcb4-44bd-b845-4706bc4cd682.xhtml.

1909. Distributed control system, from https://www.oreilly.com/library/view/industrial-cybersecurity/9781788395151/9116dbe5-4ade-43fc-9ea4-1adcea6cdac1.xhtml.

1910. Everything You Need to Know About Distributed Control System, from https://www.elprocus.com/distributed-control-system-features-and-elements/.

1911. Pascal Ackerman, Supervisory Control and Data Acquisition, from https://www.oreilly.com/library/view/industrial-cybersecurity/9781788395151/b99736a0-1e9d-4b70-9ea8-b6a76142ffd0.xhtml.

1912. SCADA Overview, from https://research.aurainfosec.io/scada-penetration-testing/#scada-overview.

1913. What is a PLC System – Different Types of PLCs with Applications, from https://www.elprocus.com/programmable-logic-controllers-and-types-of-plcs/.

1914. PLC- Industrial Applications of Programmable Logic Controller, from https://www.mobileautomation.com.au/plc-industrial-application/.

1915. Basic Process Control Systems, From http://sti-automation.net/basic-process-control-systems/.

1916. (2015), Basic Process Control System (BPCS), from http://thamilkash.blogspot.com/2015/06/identify-basic-process-control-system.html.

1917. Process Automation Protocols, from https://www.oreilly.com/library/view/industrial-cybersecurity/9781788395151/f1cac712-ca9d-4361-a4a6-715a4bb1d6b6.xhtml.

1918. Miguel Herrero Collantes and Antonio López Padilla, (2015), Protocols and Network Security in ICS Infrastructures , from https://www.incibe.es/extfrontinteco/img/File/intecocert/ManualesGuias/incibe_protocol_net_security_ics.pdf.

1919. (2016), Communication Network Dependencies for ICS/SCADA Systems, from https://www.enisa.europa.eu/publications/ics-scada-dependencies/at_download/fullReport.

1920. (2018), Industrial Protocols, from https://uploads-ssl.webflow.com/57b990500c7bcb623821b928/5ab757d71f3370e024d6184c_ProtocolUpdateDataSheet.pdf.

1921. Operational Technology and Security, from https://trustcentral.com/use-cases/operational-technology-ot-and-iiot/.

1922. Pascal Ackerman, Communication Protocols in The Enterprise Zone, from https://www.oreilly.com/library/view/industrial-cybersecurity/9781788395151/0af70163-af69-40d1-8ac2-fdfe65e2d3b1.xhtml.

1923. Glenn Schatz, (2016), Wireless IoT Network Protocols, from https://www.link-labs.com/blog/complete-list-iot-network-protocols.

1924. (2019), List of Automation Protocols, from https://en.wikipedia.org/wiki/List_of_automation_protocols.

1925. LoRa Network Protocol and Long Range Wireless IoT, from https://www.postscapes.com/long-range-wireless-iot-protocol-lora/

1926. ICS Protocols, from https://resources.infosecinstitute.com/category/certifications-training/ics-scada/ics-protocols/#gref.

1927. Sercos II, from https://www.kunbus.com/sercos-ii.html.

1928. S7 Communication (S7comm), from https://wiki.wireshark.org/S7comm.

1929. Siemens S7 Protocol, from http://sattimino.sourceforge.net/s7_protocol.html.

1930. (2015), Cyber Savvy: Securing Operational Technology Assets, from https://www.pwc.com.au/pdf/securing-operational-technology-assets.pdf.

1931. (2016), Critical Infrastructure and SCADA/ICS Cybersecurity Vulnerabilities and Threats, from https://www.checkpoint.com/downloads/products/top-10-cybersecurity-vulnerabilities-threat-for-critical-infrastructure-scada-ics.pdf.

1932. Top 10 Operational Technology Security Threats, from https://bayshorenetworks.com/.

1933. Eduard Kovacs, (2018), Vulnerability Exposes Rockwell Controllers to DoS Attacks, from https://www.securityweek.com/vulnerability-exposes-rockwell-controllers-dos-attacks.

1934. Brian Gorenc and Fritz Sands, Hacker Machine Interface, from https://documents.trendmicro.com/assets/wp/wp-hacker-machine-interface.pdf.

1935. (2019), Singapore's Operational Technology Cybersecurity Masterplan 2019, from https://www.csa.gov.sg/~/media/csa/documents/publications/ot_masterplan/csa_ot_masterplan.pdf.

1936. Galina Antova, (2017), Rethinking Vulnerability Disclosures in Industrial Control Systems, from https://www.darkreading.com/vulnerabilities-threats.

1937. Irfan Shakeel, (2016), Destructive Security Flaws in Industrial Control Systems, from https://resources.infosecinstitute.com/destructive-security-flaws-industrial-control-systems/#gref.

1938. (2018), Side-Channel Attacks Put Critical Infrastructure at Risk, from https://www.icscybersecurityconference.com/side-channel-attacks-put-critical-infrastructure-at-risk/.

1939. Eduard Kovacs, (2018), ICS Devices Vulnerable to Side-Channel Attacks: Researcher, from https://www.securityweek.com/ics-devices-vulnerable-side-channel-attacks-researcher.

1940. Dr. Siv Hilde Houmb, (2018), How to Hack Programmable Logic Controllers, from https://www.controldesign.com/control/plcs-pacs/article/11310029/how-to-hack-programmable-logic-controllers.

1941. Ali Abbasi and Majid Hashemi, (2016), Ghost in the PLC: Designing an Undetectable Programmable Logic Controller Rootkit via Pin Control Attack, from https://research.utwente.nl/en/publications/ghost-in-the-plc-designing-an-undetectable-programmable-logic-con.

1942. (2019), Attacks Against Industrial Machines via Vulnerable Radio Remote Controllers: Security Analysis and Recommendations, from https://www.trendmicro.com/vinfo/us/security/news/vulnerabilities-and-exploits/attacks-against-industrial-machines-via-vulnerable-radio-remote-controllers-security-analysis-and-recommendations.

1943. Bruce Sussman, (2019), Industrial Cybersecurity: RF Vulnerability, from https://www.secureworld.io/industry-news/industrial-cybersecurity-risk-study.

1944. (2018), An Introduction to Operational Technology and it's Security: 5 Key Facts, from https://www.vsec.infinigate.co.uk/blog/operational-technology-security-ransomware-threats.

1945. Marcel Kisch, (2017), What Do Recent Attacks Mean for OT Network Security?, from https://securityintelligence.com/what-do-recent-attacks-mean-for-ot-network-security/.

1946. (2017), Hardware Hacking in Industrial Control Systems, from https://www.incibe-cert.es/en/blog/hardware-hacking-industrial-control-system.

1947. (2017), SCADA Hacking: Hacking the Schneider Electric TM221 Modicon PLC using modbus-cli, from https://www.hackers-arise.com/post/2017/03/28/scada-hacking-hacking-the-schneider-electric-tm221-modicon-plc-using-modbus-cli.

1948. (2018), SCADA Hacking: Exploiting SCADA/ICS Systems with the Command Line Tool, modbus-cli, from https://www.hackers-arise.com/post/2018/03/22/scada-hacking-exploiting-scadaics-systems-with-the-command-line-tool-modbus-cli.

1949. William Grove, (2019), OT Networks Saw Attacks Continue to Rise in 2018, from https://blog.skyboxsecurity.com/ot_networks_2018_threats/.

1950. An Executive Guide to Cyber Security for Operational Technology, from https://www.ge.com/fr/sites/www.ge.com.fr/files/en-executive-guide-to-cyber-security-for-operational-technology-whitepaper.pdf.

1951. Dr. J.M. Ceron, Dr. J.J.Chromik, Dr. R.J.C. Santanna and Prof. dr. ir. A. Pras, (2019), Online Discoverability and Vulnerabilities of ICS/SCADA Devices in the Netherlands, from https://securitydelta.nl/media/com_hsd/report/250/document/wodc-report-scada-final.pdf.

1952. (2011), Common Cybersecurity Vulnerabilities in Industrial Control Systems, from https://www.cisa.gov/sites/default/files/recommended_practices/DHS_Common_Cybersecurity_Vulnerabilities_ICS_2010.pdf.

1953. Adrian Booth, Aman Dhingra, Sven Heiligtag, Mahir Nayfeh, and Daniel Wallance, (2019), Critical Infrastructure Companies and the Global Cybersecurity Threat, from https://www.mckinsey.com/capabilities/risk-and-resilience/our-insights/critical-infrastructure-companies-and-the-global-cybersecurity-threat.

1954. Lauren Gibbons Paul, (2018), Making Sense of the ICS Cybersecurity Market, from https://www.automationworld.com/home/article/13318353/making-sense-of-the-ics-cybersecurity-market.

1955. Open Source Honeypots That Detect Threats for Free, from https://www.smokescreen.io/practical-honeypots-a-list-of-open-source-deception-tools-that-detect-threats-for-free/.

1956. Shakir, (2020), IoT Security-Part 14 (Introduction To And Identification of Hardware Debug Ports), from https://payatu.com/static/iot-master-class-dep/iot-master-class-pdf/c14.pdf.

1957. Cristian Vences, (2020), Hardware Hacking 101: Glitching into Privileged Shells, from https://www.riverloopsecurity.com/blog/2020/10/hw-101-glitching/.

1958. (2020), Technique: Glitching U-Boot (or Other Bootloaders) by Shorting the NAND Flash, from https://www.mcafee.com/enterprise/en-us/assets/misc/ins-glitching-uboot.pdf.

1959. Brett Lischalk, (2017), Nand Glitching Wink Hub For Root, from https://www.brettlischalk.com/posts/nand-glitching-wink-hub-for-root.

1960. NAND Glitching, from https://www.oreilly.com/library/view/iot-penetration-testing/9781787280571/583faf86-b88b-4530-8c3b-9a70dfc7d9d8.xhtml.

1961. Cristian Vences, (2021), Revisiting Glitching, from https://www.riverloopsecurity.com/blog/2021/09/introducing-flash-dash/.

1962. Vincent Lee, (2020), How to Just Emulate it with QEMU, from https://www.zerodayinitiative.com/blog/2020/5/27/mindshare-how-to-just-emulate-it-with-qemu.

1963. QEMU User Space Emulator, from https://www.qemu.org/docs/master/user/main.html#other-binaries.

1964. Eross-msft, Dominicbetts, Philmea, Robinsh, and BryanLa, (2021), Security best practices for Internet of Things (IoT), from https://learn.microsoft.com/en-us/azure/iot/iot-overview-security.

1965. (2019). 8 IoT Security Best Practices, to Keep the Hackers Away from your Industrial/Enterprise Assets!, from https://www.embitel.com/blog/embedded-blog/8-iot-security-best-practices-to-keep-the-hackers-away-from-your-industrial-enterprise-assets.

1966. The Ultimate IoT Security Best Practices Guide, from https://pages.awscloud.com/rs/112-TZM-766/images/IoT_Security_Best_Practices_Guide_design_v3.1.pdf.

1967. Mary K. Pratt, (2021), Bolster Physical Defenses With IoT Hardware Security, from https://www.techtarget.com/iotagenda/tip/Bolster-physical-defenses-with-IoT-hardware-security.

1968. Internet of Things Security Challenges and Best Practices, from https://www.kaspersky.com/resource-center/preemptive-safety/best-practices-for-iot-security.

1969. (2022), ATT&CK® for Industrial Control Systems, from https://collaborate.mitre.org/attackics/index.php/Main_Page.

1970. Erin Anderson, (2020), MITRE ATT&CK for ICS Matrix: What It Is and How Its Used, from https://www.industrialdefender.com/mitre-attack-for-ics-what-it-is-how-its-used/

1971. Alessandro Di Pinto and Yiannis Stavrou, (2020), Your Guide to the MITRE ATT&CK Framework for ICS, from https://www.nozominetworks.com/blog/your-guide-to-the-mitre-attack-framework-for-ics/.

1972. Ravie Lakshmanan, (2022), U.S. Warns of APT Hackers Targeting ICS/SCADA Systems with Specialized Malware, from https://thehackernews.com/2022/04/us-warns-of-apt-hackers-targeting.html.

1973. (2022), Pipedream: New Malware Designed to Attack Industrial Control Systems Identified, from https://www.continuitycentral.com/index.php/news/technology/7236-pipedream-new-malware-designed-to-attack-industrial-control-systems-identified.

1974. Daniel Kapellmann Zafra, Raymond Leong, Chris Sistrunk, Ken Proska, Corey Hildebrandt, Keith Lunden, and Nathan Brubaker, (2022), INDUSTROYER.V2: Old Malware Learns New Tricks, from https://cloud.google.com/blog/topics/threat-intelligence/industroyer-v2-old-malware-new-tricks/.

1975. Matt Hubbard, (2021), How to Implement Zero Trust in an ICS Environment, from https://www.armis.com/blog/how-to-implement-zero-trust-in-an-ics-environment/.

1976. Dell Rodilles, (2021), 5 Steps to Realize a Zero Trust Enterprise in Critical Infrastructure, from https://www.paloaltonetworks.com/blog/2021/10/zero-trust-enterprise-in-critical-infrastructure/.

1977. Sachin Shah, (2021), Implementing Zero-Trust in an ICS environment, from https://www.tcpro.com/business-transformation/31933/five-business-benefits-of-digital-transaction-management

1978. Larry Cashdollar, (2023), Updated Kmsdx Binary Shows KmsdBot Is Targeting the IoT Landscape, from https://www.akamai.com/blog/security-research/updated-kmsdbot-binary-targeting-iot#:~:text=KmsdBot%20targets%20IoT&text=Our%20research%20into%20this%20ever,downloaded%20from%20the%20C2%20server.

1979. Chao Lei, Zhibin Zhang, and Cecilia Hu, (2023), Old Wine in the New Bottle: Mirai Variant Targets Multiple IoT Devices, from https://unit42.paloaltonetworks.com/mirai-variant-ic1h9/.

1980. (2023), Zigbee Hacking: How to Perform A Replay Attack, from https://www.youtube.com/watch?v=_NyOQCi8wRo&ab_channel=HackerAssociate

1981. Ray Felch, (2020), How To Replay RF Signals Using SDR, from https://www.blackhillsinfosec.com/how-to-replay-rf-signals-using-sdr/.

1982. Shakir, (2023), IoT Security-Part 14 (Guide To Hardware Debug Ports: Overview And Identification), from https://payatu.com/blog/iot-security-part-14-introduction-to-and-identification-of-hardware-debug-ports/.

1983. Mashav Sapir, Uri Katz, Noam Moshe, Sharon Brizinov, and Amir Preminger, (2022), Evil PLC Attack: Weaponizing PLCS, from https://web-assets.claroty.com/resource-downloads/team82-evil-plc-attack-research-paper-1661285586.pdf.

1984. Ravie Lakshmanan, (2022), New Evil PLC Attack Weaponizes PLCs to Breach OT and Enterprise Networks, from https://thehackernews.com/2022/08/new-evil-plc-attack-weaponizes-plcs-to.html.

1985. (2022), Evil PLC: The Silent Threat, from https://www.incibe.es/en/incibe-cert/blog/evil-plc-silent-threat.

1986. (2024), Unpacking the Blackjack Group's Fuxnet Malware, from https://claroty.com/team82/research/unpacking-the-blackjack-groups-fuxnet-malware.

1987. Ken Proska, Daniel Kapellmann Zafra, Keith Lunden, Corey Hildebrandt, Rushikesh Nandedkar, and Nathan Brubaker, (2023), COSMICENERGY: New OT Malware Possibly Related To Russian Emergency Response Exercises, from https://cloud.google.com/blog/topics/threat-intelligence/cosmicenergy-ot-malware-russian-response.

1988. Daniel Kapellmann Zafra, Raymond Leong, Chris Sistrunk, Ken Proska, Corey Hildebrandt, Keith Lunden, and Nathan Brubaker, (2022), INDUSTROYER.V2: Old Malware Learns New Tricks, from https://cloud.google.com/blog/topics/threat-intelligence/industroyer-v2-old-malware-new-tricks.

1989. Soumen Burma, (2024), Unveiling Abyss Locker: The Rapid Rise of a Menacing Ransomware Threat, from https://www.seqrite.com/blog/unveiling-abyss-locker-the-rapid-rise-of-a-menacing-ransomware-threat/.

1990. Nate Nelson, (2023), Feds: Beware AvosLocker Ransomware Attacks on Critical Infrastructure, from https://www.darkreading.com/ics-ot-security/feds-beware-avoslocker-ransomware-attacks-critical-infrastructure.

Module 19: Cloud Computing

1991. (2013), Cloud Computing Vulnerability Incidents: A Statistical Overview, from https://cloudsecurityalliance.org/download/cloud-computing-vulnerability-incidents-a-statistical-overview/.

1992. (2013), Introduction to Cloud Computing, from https://www.slideshare.net/ProfEdge/introduction-to-cloud-computing-23970527.

1993. Alok Tripathi and Abhinav Mishra, (2011), Cloud Computing Security Considerations, from https://www.semanticscholar.org/paper/Cloud-computing-security-considerations-Tripathi-Mishra/fd710d62f8db9621d97ab00acf1bb8e8d28e06b2.

1994. Kazi Zunnurhain and Susan V. Vrbsky, Security Attacks and Solutions in Clouds, from http://salsahpc.indiana.edu/CloudCom2010/Poster/cloudcom2010_submission_98.pdf.

1995. Chimere Barron, Huiming Yu and Justin Zhan (2013), Cloud Computing Security Case Studies and Research, from http://www.iaeng.org/publication/WCE2013/WCE2013_pp1287-1291.pdf.

1996. Keiko Hashizume, David G Rosado, Eduardo Fernández-Medina and Eduardo B Fernandez (2013), An analysis of security issues for cloud computing, from https://jisajournal.springeropen.com/articles/10.1186/1869-0238-4-5.

1997. Ian Mitchell and John Alcock, Cloud Security The definitive guide to managing risk in the new ICT landscape, from https://www.fujitsu.com/global/Images/WBOC-2-Security.pdf.

1998. Michael Cobb, (2007), What security issues can arise from unsynchronized system clocks?, from https://www.techtarget.com/searchsecurity/answers.

1999. Network Time Synchronization, from https://endruntechnologies.com/products/ntp-time-servers/network-time-synchronization.

2000. Man in the Cloud (MITC) Attacks, from https://www.imperva.com/products/web-application-firewall-waf/.

2001. Martin Gontovnikas, (2018), What is Identity as a Service (IDaaS)?, from https://auth0.com/blog/identity-as-a-service-in-2018/.

2002. Multi-Cloud, from https://www.vmware.com/topics/glossary/content/multi-cloud.html.

2003. (2019), Multicloud, from https://en.wikipedia.org/wiki/Multicloud.

2004. Rich Caldwell, (2019), Pros and Cons of a Multi Cloud Strategy, from https://centricconsulting.com/blog/pros-and-cons-of-a-multi-cloud-strategy/.

2005. Jignesh Solanki, 6 Multi-Cloud Architecture Designs for an Effective Cloud Strategy, from https://www.simform.com/blog/multi-cloud-architecture/.

2006. (2020), Cloud storage, from https://en.wikipedia.org/wiki/Cloud_storage.

2007. Laxmi Ashrit, What is Cloud Storage – Architecture, Types, Advantages & Disadvantages, from https://electricalfundablog.com/cloud-storage-architecture-types/.

2008. Basic Cloud Storage Architecture Information Technology Essay, from https://www.uniassignment.com/essay-samples/information-technology/basic-cloud-storage-architecture-information-technology-essay.php.

2009. Liz Alton, (2019), 4 Ways AI Is Improving Cloud Computing, from https://community.connection.com/4-ways-ai-is-improving-cloud-computing/.

2010. Vishal Bhatia, The Role of Artificial Intelligence in Cloud Computing, from https://www.goodfirms.co/blog/role-of-ai-in-cloud-computing.

2011. Neeru Jain, (2019), Top 7 Benefits of Using AI in Cloud Computing, from https://www.whizlabs.com/blog/benefits-of-ai-in-cloud-computing/.

2012. Bob O'Donnell, (2016), Virtual Reality and the Cloud Belong Together, from https://www.vox.com/2016/4/6/11585914/virtual-reality-in-the-cloud.

2013. (2019), What is Containers as a service (CaaS)?, from https://www.ibm.com/consulting/cloud-managed.

2014. Murugiah Souppaya, John Morello, and Karen Scarfone, (2017), Application Container Security Guide, from https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-190.pdf.

2015. Sten Pittet, What is a Container?, from https://www.atlassian.com/microservices/cloud-computing/containers-vs-vms.

2016. Pethuru Raj, Jeeva S. Chelladhurai, and Vinod Singh, (2015), Containerization vs Virtualization – An Introduction to Docker, from https://devm.io/docker/containerization-vs-virtualization-docker-introduction-120662.

2017. How is containerization different from virtualization?, from https://www.techopedia.com/7/31288/technology-trends/how-is-containerization-different-from-virtualization.

2018. Roderick Bauer, (2018), What's the Diff: VMs vs Containers, from https://www.backblaze.com/blog/vm-vs-containers/.

2019. (2020), Docker (software), from https://en.wikipedia.org/wiki/Docker_(software).

2020. Docker overview, from https://docs.docker.com/guides/docker-overview/.

2021. Docker Containers, from https://www.aquasec.com/wiki/display/containers/Docker+Containers.
2022. Avi, (2019), Docker Architecture and its Components for Beginner, from https://geekflare.com/docker-architecture/.
2023. Docker Architecture, from https://www.aquasec.com/wiki/display/containers/Docker+Architecture.
2024. Swarm mode overview, from https://docs.docker.com/engine/swarm/.
2025. What is Docker Swarm, from https://www.aquasec.com/wiki/display/containers/Docker+Containers#DockerContainers-DOCKERSWARM.
2026. (2019), Designing a Microservices Architecture with Docker Containers, from https://www.sumologic.com/blog/microservices-architecture-docker-containers/.
2027. Asad Faizi, (2019), Microservices Orchestration with Kubernetes, from https://faun.pub/microservices-orchestration-with-kubernetes-1cbb737cfa46.
2028. Mark Church, Marlon Ruiz, Andrew Seifert, and Trapier Marshall, Docker Swarm Reference Architecture: Exploring Scalable, Portable Docker Container Networks, from https://success.docker.com/article/networking.
2029. (2018), Docker Networking, from https://github.com/kyhau/docker-notebook/blob/master/docker-networking.md.
2030. Saurabh Kulshrestha, (2018), Docker Networking - Explore How Containers Communicate With Each Other, from https://medium.com/edureka/docker-networking-1a7d65e89013.
2031. Isaac Eldridge, (2018), What Is Container Orchestration?, from https://newrelic.com/blog/best-practices.
2032. Container Orchestration, from https://avinetworks.com/glossary/container-orchestration/.
2033. (2019), What is Kubernetes, from https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/.
2034. Kubernetes Architecture 101, from https://www.aquasec.com/wiki/display/containers/Kubernetes+Architecture+101.
2035. (2020), Kubernetes Components, from https://kubernetes.io/docs/concepts/overview/components/.
2036. Guillermo Velez, (2019), Kubernetes vs. Docker: A Primer, from https://cloudnativenow.com/topics/cloudnativedevelopment/kubernetes-vs-docker-a-primer/.
2037. Jim Armstrong, Docker And Kubernetes? I thought you were competitors?, from https://www.docker.com/blog/top-questions-docker-kubernetes-competitors-or-together/.
2038. Amir Ierbi, (2017), 8 Docker security rules to live by, from https://www.infoworld.com/article/3154711/8-docker-security-rules-to-live-by.html.
2039. (2018), Security Challenges Related to Containers, from https://www.ariacybersecurity.com/container-security-challenges-blog/.
2040. Christopher Tozzi, (2018), 3 Container Security Advantages and 3 Security Challenges, from https://cloudnativenow.com/topics/cloudnativesecurity/3-container-security-advantages-and-3-security-challenges/.
2041. Paul Castro, Vatche Ishakian, Vinod Muthusamy, and Aleksander Slominski, (2019), The Rise of Serverless Computing, from https://cacm.acm.org/research/the-rise-of-serverless-computing/.
2042. (2020), Serverless Computing, from https://en.wikipedia.org/wiki/Serverless_computing.
2043. David Prothero, Serverless Architecture, from https://www.twilio.com/docs/glossary/what-is-serverless-architecture.
2044. Why use Serverless Computing? | Pros and Cons of Serverless, from https://www.cloudflare.com/learning/serverless/why-use-serverless/.
2045. Jignesh Solanki, Serverless Architecture: A Comprehensive Guide, from https://www.simform.com/serverless-architecture-guide/.
2046. (2014), Cloud Top 10 Security Risks, from https://www.owasp.org/index.php/Category:OWASP_Cloud_%E2%80%90_10_Project.
2047. Shankar Babu, Chebrolu, Vinay Bansal, and Pankaj Telang, Top 10 Cloud Risks That Will Keep You Awake at Night, from https://www.owasp.org/images/4/47/Cloud-Top10-Security-Risks.pdf.
2048. OWASP Top 10 (2017) Interpretation for Serverless, from https://www.owasp.org/images/5/5c/OWASP-Top-10-Serverless-Interpretation-en.pdf.
2049. Lance Whitney, (2019), How to Prevent the Top 11 Threats in Cloud Computing, from https://www.techrepublic.com/article/how-to-prevent-the-top-11-threats-in-cloud-computing/.
2050. Chester Avey, (2019), 7 Key Cybersecurity Threats to Cloud Computing, from https://cloudacademy.com/blog/key-cybersecurity-threats-to-cloud-computing/.
2051. Rakesh Soni, (2019), The Rise of Cloud Computing Threats: How to protect your cloud customers from security risks, from https://customerthink.com/the-rise-of-cloud-computing-threats-how-to-protect-your-cloud-customers-from-security-risks/.
2052. (2019), Container Security: Examining Potential Threats to the Container Environment, from https://www.trendmicro.com/vinfo/us/security/news/security-technology/container-security-examining-potential-threats-to-the-container-environment.
2053. Container Vulnerabilities and Threats, from https://www.aquasec.com/wiki/display/containers/Container+Vulnerabilities+and+Threats.

2054. (2019). Cloud Container Vulnerabilities Soar, According to Report, from https://www.globenewswire.com/news-release/2019/07/24/1886967/0/en/Cloud-Container-Vulnerabilities-Soar-According-to-Report.html.

2055. Anurag Kahol, (2019), Beware the man in the cloud: How to protect against a new breed of cyberattack, from https://www.helpnetsecurity.com/2019/01/21/mitc-attack/.

2056. Adrian Nish and Tom Rowles, (2017), APT10 - OPERATION CLOUD HOPPER, from https://baesystemsai.blogspot.com/2017/04/apt10-operation-cloud-hopper_3.html.

2057. Nathaniel Richmond, (2019), Operation Cloud Hopper Case Study, from https://insights.sei.cmu.edu/sei_blog/2019/03/operation-cloud-hopper-case-study.html.

2058. Jeremy Kirk, (2019), Cloud Hopper: Major Cloud Services Victims Named, from https://www.bankinfosecurity.com/cloud-hopper-major-cloud-services-victims-named-a-12695.

2059. (2018), Cryptojacking Attacks – Securonix Security Advisory (SSA), from https://www.securonix.com/web/wp-content/uploads/2018/08/cryptojacking_security_advisory.pdf.

2060. Charlie Osborne, (2018), Cryptojacking Attacks Surge Against Enterprise Cloud Environments, from https://www.zdnet.com/article/cryptojacking-attacks-surge-against-enterprise-cloud-environments/.

2061. Trenton Baker, (2018), Mobile and Cloud Cryptojacking Skyrockets, from https://cyberfortress.com/?utm_source=www.keepitsafe.com&utm_medium=website_redirect&utm_campaign=website_sunsetting.

2062. Tara Seals, (2019), 'Cloudborne' IaaS Attack Allows Persistent Backdoors in the Cloud, from https://threatpost.com/cloudborne-iaas-attack-cloud/142223/.

2063. Rene Millman, (2019), Bare metal flaw allows hackers to put backdoors into cloud servers, from https://www.itpro.com/network-security/33106/bare-metal-flaw-allows-hackers-to-put-backdoors-into-cloud-servers.

2064. Maria Deutscher, New Cloudborne vulnerability exposes cloud servers to potential hacking, from https://siliconangle.com/2019/02/26/new-cloudborne-vulnerability-potentially-exposes-cloud-servers-hacking/.

2065. Kelly Sheridan, (2019), Cloudborne: Bare Metal Cloud Servers Vulnerable to Attack, from https://www.darkreading.com/cloud-security/-cloudborne-bare-metal-cloud-servers-vulnerable-to-attack.

2066. Aditya K Sood and Rohan Jalil, (2018), Cloudifying Threats—Understanding Cloud App Attacks and Defenses, from https://www.issa.org/journal/archives/2018/Volume-1/Pages/cloudifying-threats-understanding-cloud-app-attacks-and-defenses.aspx?utm_referrer=.

2067. Metadata Spoofing Attack, from https://books.google.as/books?id=x4e6BQAAQBAJ&pg=PA182&lpg=PA182&dq=Cloud+Malware+Injection+Attack&source=bl&ots=xv-egwmdJ9&sig=ACfU3U1HzGBMIcKJu-SwIOzFsLjGUlvEw&hl=en&sa=X&ved=2ahUKEwjSwsmigoXmAhVYB3MBHe64A944KBDoATAFegQICBAB#v=onepage&q=metadata%20spoofing&f=false.

2068. What is an HTTP Flood DDoS attack?, from https://www.cloudflare.com/learning/ddos/http-flood-ddos-attack/.

2069. (2017), Detection of HTTP Flooding Attacks in Cloud Using Dynamic Entropy Method, from https://www.researchgate.net/publication/321185428_Detection_of_HTTP_Flooding_Attacks_in_Cloud_Using_Dynamic_Entropy_Method.

2070. Anna Bryk, (2018), Cloud Computing: A New Vector for Cyber Attacks, from https://www.apriorit.com/dev-blog/523-cloud-computing-cyber-attacks.

2071. (2019), Top 5 Cloud Computing Security Issues; and How they are used by Hackers, from https://www.cloudmanagementinsider.com/top-5-cloud-computing-security-issues-and-strategies-used-by-hackers/.

2072. Warwick Ashford, (2018), Hackers increasingly Targeting Cloud Infrastructure, from https://www.computerweekly.com/news/252444716/Hackers-increasingly-targeting-cloud-infrastructure.

2073. (2018), A Practical Guide to Testing the Security of Amazon Web Services (Part 1: AWS S3), from https://blog.mindedsecurity.com/2018/09/a-practical-guide-to-testing-security.html.

2074. Working with Amazon S3 Buckets, from https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingBucket.html.

2075. Rohan Chavan, (2019), Finding and Testing MisConfigured S3 Buckets, from https://rohanchavan.medium.com/finding-and-testing-misconfigured-s3-buckets-d77992c4b5cd.

2076. Rmorrill, (2012), Google hacking Amazon Web Services Cloud front and S3, from https://it.toolbox.com/blogs/rmorrill/google-hacking-amazon-web-services-cloud-front-and-s3-011613.

2077. Matthew Palmer, How Does Kubernetes Use etcd?, from https://matthewpalmer.net/kubernetes-app-developer/articles/how-does-kubernetes-use-etcd.html.

2078. Spencer Gietzen, Enumerating AWS Roles through 'AssumeRole', from https://rhinosecuritylabs.com/aws/assume-worst-aws-assume-role-enumeration/.

2079. Jay Jani, (2019), S3 Bucket Misconfiguration: From Basics to Pawn, from https://bugbountypoc.com/s3-bucket-misconfiguration-from-basics-to-pawn/.

2080. Spencer Gietzen, Cloud Breach: Compromising AWS IAM Credentials, from https://rhinosecuritylabs.com/aws/aws-iam-credentials-get-compromised/.

2081. Vince Lujan, (2017), What is AWS IAM?, from https://jumpcloud.com/blog/what-is-aws-iam/.

2082. Vitaly Simonovich and Ori Naker, (2019), Hundreds of Vulnerable Docker Hosts Exploited by Cryptocurrency Miners, from https://www.imperva.com/blog/hundreds-of-vulnerable-docker-hosts-exploited-by-cryptocurrency-miners/.

2083. Riyaz Walikar, (2019), Getting Shell and Data Access in AWS by Chaining Vulnerabilities, from https://blog.appsecco.com/getting-shell-and-data-access-in-aws-by-chaining-vulnerabilities-7630fa57c7ed.

2084. Riyaz Walikar, (2019), An SSRF, Privileged AWS Keys and the Capital One Breach, from https://blog.appsecco.com/an-ssrf-privileged-aws-keys-and-the-capital-one-breach-4c3c2cded3af.

2085. Spencer Gietzen, AWS IAM Privilege Escalation – Methods and Mitigation, from https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation/.

2086. Setu, (2018), What is AWS Post exploitation?, from https://cloudsecops.com/aws-post-exploitation-part-1/

2087. Spencer Gietzen, Google Cloud Platform (GCP) Bucket Enumeration and Privilege Escalation, from https://rhinosecuritylabs.com/gcp/google-cloud-platform-gcp-bucket-enumeration/.

2088. Pawel Rzepa, (2018), Playing with CloudGoat part 2: Fooling AWS CloudTrail and Getting Persistent Access, from https://rzepsky.medium.com/playing-with-cloudgoat-part-2-fooling-cloudtrail-and-getting-persistence-access-6a1257bb3f7c.

2089. (2019), Cloud Computing Security Considerations, from https://www.cyber.gov.au/publications/cloud-computing-security-considerations.

2090. Top 6 Considerations for Cloud Security and Data Protection, from https://www.techtarget.com/searchstorage/ironMountainCloud/Top-6-Considerations-For-Cloud-Security-and-Data-Protection.

2091. (2018), Moving to the Cloud – Cloud Security Considerations, from https://cloudcheckr.com/cloud-security/moving-cloud-security/.

2092. Gerry Grealish, Six Key Security Considerations for Responsible Cloud Migration, from https://docs.broadcom.com/doc/six-key-considerations-for-responsible-cloud-migration-en.

2093. Cynthia Harvey, (2017), Cloud Security: 11 Best Practices, from https://www.esecurityplanet.com/cloud/cloud-security-best-practices.html.

2094. Jason Meilleur, (2019), The Growing Dangers of Cyber Attacks and the Need for Cloud Security, from https://www.360visibility.com/the-growing-dangers-of-cyber-attacks-and-the-need-for-cloud-security/.

2095. (2019), 19 Cloud Security Best Practices for 2019, from https://www.mcafee.com/blogs/.

2096. Matt Miller, (2018), Cloud Security Best Practices, from https://www.beyondtrust.com/blog/entry/cloud-security-best-practices.

2097. Rani Osnat, (2018), Top Docker Security Best Practices, from https://blog.aquasec.com/docker-security-best-practices.

2098. (2017), Major Risks for Core Components of Container Technologies and their Countermeasures NIST SP 800 190, from https://www.hack2secure.com/blogs/major-risks-for-core-components-of-container-technologies-and-their-countermeasures-nist-sp-800-190.

2099. Devdatta Mulgund, (2019), 8 Best Practices for Application Container Security, from https://securityintelligence.com/posts/8-best-practices-for-application-container-security/.

2100. Docker_Security_Cheat_Sheet.md, from https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/Docker_Security_Cheat_Sheet.md.

2101. Jeff Hale, (2019), Top 20 Docker Security Tips, from https://towardsdatascience.com/top-20-docker-security-tips-81c41dd06f57.

2102. (2019), Understanding Docker Container Escapes, from https://blog.trailofbits.com/2019/07/19/understanding-docker-container-escapes/.

2103. (2019), Understanding and Preparing for Container security threats, from https://cyware.com/news/understanding-and-preparing-for-container-security-threats-26031da9.

2104. Docker Security Resources, from https://www.aquasec.com/wiki/display/containers/Docker+Security+Resources.

2105. Sean Michael Kerner, (2019), Serverless Cloud Security: How to Secure Serverless Computing, from https://www.esecurityplanet.com/cloud/serverless-computing/.

2106. Liran Tal and Guy Podjarny, (2019), 10 Serverless Security Best Practices, from https://snyk.io/blog/10-serverless-security-best-practices/.

2107. Bianca Soare, (2019), What is the Zero Trust Model?, from https://heimdalsecurity.com/blog/what-is-the-zero-trust-model/.

2108. What is a Cloud Firewall? What is Firewall-as-a-service (FWaaS)?, from https://www.cloudflare.com/learning/cloud/what-is-a-cloud-firewall.

2109. What is Firewall as a Service (FWaaS)?, from https://www.zenarmor.com/docs/network-security-tutorials/what-is-firewall-as-a-service-fwaas.

2110. Firewall as a Service (FWaaS), from https://www.fortinet.com/resources/cyberglossary/firewall-as-a-service-fwaas.

2111. What is Firewall as a Service?, from https://www.zscaler.com/resources/security-terms-glossary/what-is-firewall-as-a-service.
2112. Firewalls as a Service (FWaaS): The Future of Network Security, from https://oreta.com.au/firewalls-as-a-service-fwaas-the-future-of-network-security.
2113. Anything as a Service (Xaas), from https://timesofcloud.com/cloud-tutorial/xaas/.
2114. Natallia Sakovich, Everything-as-a-Service (Xaas): Definition and Examples, from https://www.sam-solutions.com/blog/everything-as-a-service-xaas-definition-and-examples/.
2115. Forrest Stroud, (2013), Xaas – Anything-As-A-Service, from https://www.webopedia.com/definitions/anything-as-a-service-xaas/.
2116. What is XaaS (anything as a service)?, from https://www.netapp.com/knowledge-center/what-is-anything-as-a-service-xaas/#:~:text=%E2%80%9CAnything%20as%20a%20service%E2%80%9D%20(,a%20service%20over%20the%20Internet.
2117. Ryan Squires, (2018), Everything-as-a-Service (Xaas): From Software to Property, from https://umpcloud.com/blog/xaas.
2118. (2018), Everything-as-a-Service: Have Federal CIOs Found Their Holy Grail?, from https://www.eglobaltech.com/post/everything-as-a-service-have-federal-cios-found-their-holy-grail.
2119. Alam Mohammed, What are the Core Cloud Computing Concepts?, from https://www.finsliqblog.com/cloud-computing/what-are-the-core-concepts-of-cloud-computing/.
2120. What is Desktop as a Service (DaaS)?, from https://www.vmware.com/topics/glossary/content/desktop-as-a-service.html?resource=cat-1178904d40#cat-1178904440.
2121. Bill Detwiler, (2020), Top Desktop as a service (DaaS) Providers: Amazon, Citrix, Microsoft, VMware, and more, from https://www.techrepublic.com/article/top-desktop-as-a-service-daas-providers-amazon-citrix-microsoft-vmware-and-more/.
2122. Desktop as a Service (DaaS), from https://www.citrix.com/glossary/what-is-desktop-as-a-service-daas.html.
2123. Tahani Khaled, (2018), What is DaaS?, from https://blog.thinprint.com/what-is-daas/.
2124. Christopher Fanchi, (2022), What Is Mobile Backend As A Service (MBaaS)?, from https://backendless.com/what-is-mobile-backend-as-a-service-mbaas/.
2125. What is the Machines as a Service Business Model?, from https://www.exonn.com/en/blog/2019/04/26/what-is-the-machines-as-a-service-business-model/#:~:text=%20What%20is%20the%20Machines%20as%20a%20Service,definitely%20something%20both%20machine%20manufacturers%20and...%20More%20.
2126. Danielle Collins, (2022), What is Machine as a Service (MaaS) and What Are Its Benefits?, from https://www.motioncontroltips.com/what-is-machine-as-a-service-what-are-its-benefits/.
2127. (2022), The Advantages and disadvantages of "Machine-as-a-Service", from https://blog.prophetic-technology.com/the-advantages-and-disadvantages-of-machine-as-a-service.
2128. Google Distributed Cloud, from https://cloud.google.com/distributed-cloud.
2129. What is Distributed Cloud Computing?, from https://www.stackpath.com/edge-academy/what-is-distributed-cloud-computing.
2130. What is a Distributed Cloud?, from https://www.vmware.com/topics/glossary/content/distributed-cloud.html.
2131. Michael Isberto, (2021), What is the Difference between a Multi Cloud and Poly Cloud Strategy?, from https://www.colocationamerica.com/blog/poly-cloud-vs-multi-cloud#:~:text=A%20poly%2Dcloud%20is%20a,between%20multicloud%20and%20poly%2Dcloud.
2132. (2019), Multi Cloud/Poly Cloud, from https://abs.sogeti.com/multi-cloud-poly-cloud/.
2133. Lee Atchison, Chapter 1. What Is Polycloud?, from https://www.oreilly.com/library/view/what-is-polycloud/9781098104634/ch01.html.
2134. Fog Computing vs. Cloud Computing for IoT Projects, from https://www.sam-solutions.com/blog/fog-computing-vs-cloud-computing-for-iot-projects/#:~:text=The%20definition%20may%20sound%20like,able%20to%20provide%20instant%20connections.
2135. (2020), Difference between Cloud, Fog and Edge Computing in IoT, from https://www.digiteum.com/cloud-fog-edge-computing-iot/.
2136. Tim Keary, (2018), What is Fog Computing?, from https://www.itpro.com/fog-computing/.
2137. (2020), Difference Between Cloud Computing and Fog Computing, from https://www.geeksforgeeks.org/difference-between-cloud-computing-and-fog-computing/.
2138. Rahul Hirve, (2017), What is Fog Computing? Why Fog Computing Trending Now?, from https://medium.com/yeello-digital-marketing-platform/what-is-fog-computing-why-fog-computing-trending-now-7a6bdfd73ef.
2139. Brandon Butler, (2018), What is fog computing? Connecting the cloud to things, from https://www.networkworld.com/article/3243111/what-is-fog-computing-connecting-the-cloud-to-things.html.
2140. Brien Posey, (2021), What is fog computing?, from https://www.techtarget.com/iotagenda/definition/fog-computing-fogging.
2141. Jagreet Kaur, (2022), Edge Computing and its Impact on IoT, from https://www.xenonstack.com/blog/edge-computing/.

2142. Real-Life Use Cases for Edge Computing, from https://innovationatwork.ieee.org/real-life-edge-computing-use-cases/.
2143. (2020), What is Series (#10): What is Edge Computing?, from https://nicolasavindpassinger.com/what-is-series-edge-computing.
2144. Alison DeNisco-Rayome, (2018), Ten Scenarios Where Edge Computing Can Bring New Value, from https://www.zdnet.com/article/10-scenarios-where-edge-computing-can-bring-new-value/.
2145. 10 Edge Computing Use Case Examples, from https://stlpartners.com/edge-computing/10-edge-computing-use-case-examples/.
2146. (2017), Cloud, Fog and Edge Computing – What's the Difference?, from https://www.winsystems.com/cloud-fog-and-edge-computing-whats-the-difference/.
2147. Oriol Rius, Cloud, Edge, and Fog Computing – Practical Application for Each, from https://www.e-zigurat.com/innovation-school/blog/cloud-edge-fog-computing-practical-applications/.
2148. (2020), Difference between Cloud Computing and Grid Computing, from https://www.geeksforgeeks.org/difference-between-cloud-computing-and-grid-computing/.
2149. Difference between Cloud Computing and Grid Computing, from https://www.javatpoint.com/cloud-computing-vs-grid-computing.
2150. Grid Computing Vs Cloud Computing, from https://techvidvan.com/tutorials/grid-computing-vs-cloud-computing/.
2151. Grid Computing Vs Cloud Computing – Top 13 Factors of Difference, from https://data-flair.training/blogs/grid-computing-vs-cloud-computing/.
2152. Aaron Nordhoff, (2020), What is a Cluster? An Overview of Clustering in the Cloud, from https://www.capitalone.com/tech/cloud/what-is-a-cluster/.
2153. Computer Clusters, from https://www.virtana.com/glossary/what-is-a-cluster/.
2154. Jason Hoffman, What is Cluster Computing and How It Is Different From Cloud Computing?, from https://wisdomplexus.com/blogs/what-is-cluster-computing/.
2155. Ryan Davis, (2021), Insecure API Cloud Computing: The Causes and Solutions, from https://www.extrahop.com/company/blog/2020/insecure-apis-cloud-computing-cause-solutions.
2156. Matt Cauthorn, (2020), Insecure API Cloud Computing: The Causes and Solutions, from https://www.optiv.com/insights/discover/blog/insecure-api-cloud-computing-causes-and-solutions.
2157. Ajoy Kumar, (2020), How the CDN Cache Poisoning Vulnerability Leads to DoS Attacks?, from https://www.thecoderworld.com/how-the-cdn-cache-poisoning-vulnerability-leads-to-dos-attacks.
2158. (2022), CDN Cache Poisoning Enables DoS Attacks On Cloud Apps, from https://roxxcloud.com/cdn-cache-poisoning-enables-dos-attacks-on-cloud-apps.
2159. Swati Khandelwal, (2019), New Cache Poisoning Attack Lets Attackers Target CDN Protected Sites, from https://thehackernews.com/2019/10/cdn-cache-poisoning-dos-attack.html.
2160. Catalin Cimpanu, (2019), CPDoS Attack Can Poison CDNs to Deliver Error Pages Instead of Legitimate Sites, from https://www.zdnet.com/article/cpdos-attack-can-poison-cdns-to-deliver-error-pages-instead-of-legitimate-sites.
2161. Sergei Shevchenko, (2020), Cloud Snooper Attack Bypasses AWS Security Measures, from https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/sophoslabs-cloud-snooper-report.pdf.
2162. Madhavanti M, (2021), Spotting and Stopping Cloud Attacks: AWS Server Hack, from https://www.manageengine.com/log-management/cyber-security/aws-cloud-snooper-attack.html.
2163. Kelly Jackson Higgins, (2020), Cloud Snooper' Attack Circumvents AWS Firewall Controls, from https://www.darkreading.com/cloud/ cloud-snooper-attack-circumvents-aws-firewall-controls.
2164. (2020), Cloud Snooper, an Advanced Targeted Attack that Allows Malware to Communicate Across Firewalls, from https://www.expresscomputer.in/news/cloud-snooper-an-advanced-targeted-attack-that-allows-malware-to-communicate-across-firewalls/50165/.
2165. Understanding Golden SAML Forgery Attacks, from https://www.qomplx.com/qomplx-knowledge-understanding-golden-saml-forgery-attacks/#how-golden-saml-attacks-work.
2166. (2021), Detection and Hunting of Golden SAML Attack, from https://www.sygnia.co/threat-reports-and-advisories/golden-saml-attack/.
2167. Shaked Reiner, Golden SAML: Newly Discovered Attack Technique Forges Authentication to Cloud Apps, from https://www.cyberark.com/resources/threat-research-blog/golden-saml-newly-discovered-attack-technique-forges-authentication-to-cloud-apps.
2168. Jay Chen, Aviv Sasson, and Ariel Zelivansky, (2021), Hildegard: New TeamTNT Cryptojacking Malware Targeting Kubernetes, from https://unit42.paloaltonetworks.com/hildegard-malware-teamtnt/.
2169. Paolo Passeri, (2022), The Exploitation of Cloud Services Continues in 2022, from https://www.infosecurity-magazine.com/blogs/exploitation-cloud-services-2022/.
2170. Sean Metcalf, (2020), What is Azure Active Directory?, from https://adsecurity.org/?p=4211.
2171. Introduction to the Instance Metadata Service, from https://hackingthe.cloud/aws/general-knowledge/intro_metadata_service/.

2172. (2019). Get Access/Secret Key from EC2 Instance Metadata, from https://www.youtube.com/watch?v=nhqTCrmrx4A.
2173. Retrieve instance metadata, from https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instancedata-data-retrieval.html.
2174. Stefano Chierici, (2022). Vulnerable AWS Lambda function – Initial Access in Cloud Attacks, from https://sysdig.com/blog/exploit-mitigate-aws-lambdas-mitre/.
2175. What is a Shadow Admin?, from https://docs.cyberark.com/Product-Doc/OnlineHelp/CEM/Latest/en/Content/CloudAdmin/kd_shadow-admin-per-platform.htm#Tab2.
2176. Asaf Hecht, (2020), DIY: Hunting Azure Shadow Admins Like Never Before, from https://www.cyberark.com/resources/threat-research-blog/diy-hunting-azure-shadow-admins-like-never-before-2.
2177. Asaf Hecht, (2020), Fantastic Cloud Shadow Admins and where to Find, from https://cfp.hacktest.ca/hacktest-2020/talk/7J8ZEW/.
2178. Shay Siksik, Privilege Escalation and Lateral Movement on Azure – Part 1, from https://www.sincyber.com/privilege-escalation-and-lateral-movement-on-azure-part-1/.
2179. Jeff Petters, (2020), What is SAML and How Does it Work?, from https://www.varonis.com/blog/what-is-saml/.
2180. Russell Jones, (2021), How SAML 2.0 Authentication Works?, from https://goteleport.com/blog/how-saml-authentication-works/.
2181. Christine Mikolajczak, (2017), An Introduction to SAML (Security Assertion Markup Language), from https://www.secureauth.com/blog/an-introduction-to-saml-security-assertion-markup-language/.
2182. Josh Fruhlinger, (2021), SAML explained: How this open standard enables single sign on, from https://www.csoonline.com/article/3232355/what-is-saml-how-it-works-and-how-it-enables-single-sign-on.html.
2183. (2019), SAML 2.0: Technical Overview, from https://www.youtube.com/watch?v=5vppXbpv-5k.
2184. What is Cloud Networking?, from https://www.vmware.com/topics/glossary/content/cloud-networking.
2185. What is a Virtual Private Cloud (VPC)?, From https://www.checkpoint.com/cyber-hub/cloud-security/what-is-vpc-virtual-private-cloud/.
2186. What is a Virtual Private Cloud (VPC)?, from https://www.cloudflare.com/learning/cloud/what-is-a-virtual-private-cloud/.
2187. VPC With Public and Private Subnets (NAT), from https://docs.aws.amazon.com/vpc/latest/userguide/vpc-example-private-subnets-nat.html.
2188. What's the Difference Between a Public and Private Subnet in an AWS VPC?, from https://chrisguitarguy.com/2017/09/28/public-private-subnet-differences-aws-vpc/.
2189. Mohamed Jawad, AWS Transit Gateway, from https://blogs.tensult.com/2019/01/02/aws-transit-gateway/.
2190. Fernando Hönig, (2016), Save Money with AWS VPC Endpoints, from https://medium.com/nubego/how-to-save-money-with-aws-vpc-endpoints-9bac8ae1319c.
2191. Debasish Pramanik, (2021), Data Loss Prevention Techniques for Cloud Computing Security – Reaping Benefits & Minimizing Risk, from https://www.cloudcodes.com/blog/data-loss-prevention-for-cloud-computing.html.
2192. Oleksandr Bushkovskyi, Cloud Computing Security Risks in 2021, and How to Avoid Them, from https://theappsolutions.com/blog/development/cloud-security-risks/.
2193. Deniz Mustafa, (2021), Cloud Security: Risks and Countermeasures, from https://www.securiwiser.com/blog/cloud-security-risks-and-countermeasures/.
2194. Bea Potter, (2020), 6 Ways to Prevent a Data Breach, from https://cloudacademy.com/blog/ways-to-prevent-a-data-breach/.
2195. Abuse And Nefarious Use Of Cloud Computing Information Technology Essay, from https://www.uniassignment.com/essay-samples/information-technology/abuse-and-nefarious-use-of-cloud-computing-information-technology-essay.php?vref=1.
2196. Fighting the Top 12 Threats to Cloud Cyber Security: Threats 10-12, from https://www.whoa.com/fighting-the-top-12-threats-to-clouds-cyber-security-threats-10-12.
2197. (2020), Top 5 Shared Hosting Security Risks (And How To Prevent Them), from https://www.malcare.com/blog/shared-hosting-security/#How_To_Protect_Your_Website_From_Shared_Hosting_Security_Risks.
2198. Karen Kent and Murugiah Souppaya, Guide to Computer Security Log Management, from https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-92.pdf?msclkid=4318dd3dc52911ec9e00e6aa020addc.
2199. ISO 27001 Annex A.12.1, from https://www.isms.online/iso-27001/annex-a-12-operations-security/?msclkid=5ff75514c52511ecbbc28744fc08bf64.
2200. Iohna Till Johnson, (2020), 5 Steps to Help Prevent Supply Chain Cybersecurity Threats, from https://www.techtarget.com/searchsecurity/tip/5-steps-to-help-prevent-supply-chain-cybersecurity-threats.
2201. C.J. Haughey, (2021), 5 Global Supply Chain Security Threats (and How to Handle Them), from https://securityintelligence.com/articles/global-supply-chain-security-threats-how-to-handle/.
2202. Edward Kost, (2022), 11 Ways to Prevent Supply Chain Attacks in 2022 (Highly Effective), from https://www.upguard.com/blog/how-to-prevent-supply-chain-attacks.

2203. (2021). Top 15 Cloud Security Issues, Threats and Concerns, from https://www.checkpoint.com/cyber-hub/cloud-security/what-is-cloud-security/top-cloud-security-issues-threats-and-concerns/.

2204. Mike Engle, (2021), Cloud Authentication Services | Security for Enterprise IT, from https://www.1kosmos.com/authentication/cloud-authentication-services/.

2205. What is Vendor Lock-in? | Vendor Lock-in and Cloud Computing, from https://www.cloudflare.com/en-in/learning/cloud/what-is-vendor-lock-in/.

2206. Muhammad Raza, (2020), 10 Best Practices to Avoid Cloud Vendor Lock-In, from https://www.bmc.com/blogs/vendor-lock-in/.

2207. Haim Glickman, (2020), Cloud Governance Best Practices: How To Create A Framework For Success, from https://www.forbes.com/sites/forbestechcouncil/2020/01/17/cloud-governance-best-practices-how-to-create-a-framework-for-success/?sh=5894a34148ce.

2208. (2020), 5 Cloud Governance Best Practices, from https://eviden.com/about-us/legacy-brands/cloudreach/.

2209. David Howell, (2021), How to Stop IT Equipment Theft, from https://www.techradar.com/in/news/world-of-tech/roundup/how-to-stop-it-equipment-theft-1089354.

2210. (2021), Best Practices for Data Destruction, from https://bigdataanalyticsnews.com/best-practices-for-data-destruction/.

2211. Jeff Melnick, (2020), Top 6 Security Threats in Cloud Computing and How to Mitigate Them, from https://blog.netwrix.com/2020/09/08/cloud-security-threats/.

2212. (2021), How to Avoid Cloud Account Hijacking Attacks, from https://www.insightsforprofessionals.com/it/cloud/avoid-cloud-account-hijacking-attacks.

2213. What is Cloud Jacking?, from https://fraudwatch.com/4-tips-to-protect-your-business-against-cloud-jacking/.

2214. Injy ElDeeb, (2021), 6 Outsider Cloud Data Security Attacks to Keep Looking out for in 2022, from https://gatilabs.com/blogpost/6-ousider-cloud-data-security-attacks-2021/.

2215. Gurpreet Singh, (2022), Container Security Benefits and its Best Practices | Complete Guide, from https://www.xenonstack.com/insights/container-security.

2216. Melanie Tafelski, (2022), 7 Container Security Best Practices For Better Apps, from https://www.trendmicro.com/en_us/devops/22/b/container-security-best-practices.html.

2217. Bernard Brode, (2020), Container Security Best Practices Taking Shape, from https://cloudnativenow.com/topics/cloudnativesecurity/container-security-best-practices-taking-shape/.

2218. Container Security Best Practices, from https://www.tigera.io/learn/guides/container-security/container-security-best-practices/.

2219. Container Security Tips and Best Practices, from https://www.threatstack.com/blog/container-security-tips-and-best-practices.

2220. Top 7 Container Security Issues, from https://www.checkpoint.com/cyber-hub/cloud-security/what-is-container-security/top-7-container-security-issues/.

2221. Misbah Thevarmannil, (2024), 10 Container Security Risks to look out for in 2024, from https://www.practical-devsecops.com/container-security-risks/.

2222. OWASP Kubernetes Top Ten, from https://owasp.org/www-project-kubernetes-top-ten/.

2223. Etay Maor, (2023), LOL Attacks Can Now Live off the Cloud: Three Strategies to Reduce LOC Risk, from https://www.spiceworks.com/it-security/vulnerability-management/guest-article/lol-attacks-can-now-live-off-the-cloud-strategies-to-reduce-loc-risk/.

2224. Julie Pottison-Gordon, (2022), 'Living Off the Cloud': Hackers Modernize an Old-School Tactic, from https://www.govtech.com/security/living-off-the-cloud-hackers-modernize-an-old-school-tactic.

2225. (2024), Multi Cloud Attack Coverage Essentials - Resource Abuse, from https://azuremarketplace.microsoft.com/en-us/marketplace/apps/azuresentinel.azure-sentinel-solution-multicloudattackcoverage?tab=overview.

2226. Jack Dwyer, (2023), Implementing Security Across Multi-Cloud Environments, from https://ceet.co/blog/multi-cloud-security-solutions.

2227. Wagner Nascimento, (2022), Multi-Cloud Security: Challenges & Best Practices, from https://www.synopsys.com/blogs/chip-design/multi-cloud-security.html.

2228. (2023), Privilege Escalation in Windows, Linux, and K8s and 6 Ways to Prevent It, from https://www.aquasec.com/cloud-native-academy/supply-chain-security/privilege-escalation/#Privilege-Escalation-with-the-CSR-API.

2229. Rory McCune, (2022), Kubernetes RBAC: How to Avoid Privilege Escalation via Certificate Signing, from https://www.aquasec.com/blog/kubernetes-rbac-privilege-escalation/.

2230. Temporary elevated access, from https://docs.aws.amazon.com/singlesignon/latest/userguide/temporary-elevated-access.html.

2231. (2024), Just-in-Time (JIT) Access Explained: Types, Use Cases, FAQs, from https://www.conductorone.com/glossary/what-is-just-in-time-access/.

2232. (2024), Abuse Elevation Control Mechanism: Temporary Elevated Cloud Access, from https://attack.mitre.org/techniques/T1548/005/.

2233. Elizabeth Montalbano, (2024), 'Cuttlefish' Zero-Click Malware Steals Private Cloud Data, from https://www.darkreading.com/cloud-security/cuttlefish-zero-click-malware-steals-private-cloud-data.

2234. (2024), New Cuttlefish Malware Hijacks Router Connections, Sniffs for Cloud Credentials, from https://thehackernews.com/2024/05/new-cuttlefish-malware-hijacks-router.html.

2235. What Is Vulnerability Assessment? Benefits, Tools, and Process, from https://www.hackerone.com/knowledge-center/what-vulnerability-assessment-benefits-tools-and-process.

2236. Esteban Borges, (2024), Information Gathering: Techniques and Tools for Effective Research, from https://www.recordedfuture.com/threat-intelligence-101/intelligence-sources-collection/information-gathering.

2237. What is Cloud Penetration Testing?, from https://www.guidepointsecurity.com/education-center/cloud-penetration-testing/.

2238. (2022), Introduction To Post-Exploitation Phase, from https://www.geeksforgeeks.org/introduction-to-post-exploitation-phase/.

2239. Shodan Cheat Sheet, from https://cheatography.com/sir-slammington/cheat-sheets/shodan/.

2240. (2024), Shodan Dork Cheatsheet, from https://book.martiandefense.llc/notes/security-research/shodan-dork-cheatsheet.

2241. (2024), MASSCAN: Mass IP port scanner, from https://gitpiper.com/resources/pentest/networkreconnaissancetools/robertdavidgraham-masscan.

2242. MASSCAN CHEATSHEET, from https://cheatsheet.haax.fr/network/port-scanning/masscan_cheatsheet/.

2243. (2023), Tutorial: Discover and manage shadow IT, from https://learn.microsoft.com/en-us/defender-cloud-apps/tutorial-shadow-it.

2244. (2021), Detect Security Risks in Your Cloud Account using CloudSploit, from https://alphasec.io/detect-security-risks-in-your-cloud-account-using-cloudsploit/.

2245. (2024), Cloud8rute: Harnessing the Power of Cloud Storage Enumeration, from https://braincraze.net/cloudbrute-harnessing-the-power-of-cloud-storage-enumeration/.

2246. (2022), The Ultimate Guide for Cloud Penetration Testing, from https://www.prplbx.com/resources/blog/cloud-pentesting/.

2247. AWS Enumeration – Part II (Practical Enumeration), from https://securitycafe.ro/2022/12/14/aws-enumeration-part-ii-practical-enumeration/.

2248. AWS Command Line Interface, from https://aws.amazon.com/cli/.

2249. Describe-db-instances, from https://docs.aws.amazon.com/cli/latest/reference/rds/describe-db-instances.html.

2250. (2024), AWS - Relational Database (RDS) Enum, from https://cloud.hacktricks.xyz/pentesting-cloud/aws-security/aws-services/aws-relational-database-rds-enum.

2251. Cloudsplaining, from https://cloudsplaining.readthedocs.io/en/latest/.

2252. (2023), Principals, from https://opensource.salesforce.com/cloudsplaining/#/.

2253. (2024), Cognito Identity Pools, from https://cloud.hacktricks.xyz/pentesting-cloud/aws-security/aws-services/aws-cognito-enum/cognito-identity-pools.

2254. (2024), AWS - DynamoDB Enum, from https://cloud.hacktricks.xyz/pentesting-cloud/aws-security/aws-services/aws-dynamodb-enum.

2255. Cartography documentation, from https://lyft.github.io/cartography/usage/tutorial.html#given-a-node-label-what-other-node-labels-can-be-connected-to-it.

2256. (2024), Security group exposes risky ports to the internet, from https://securitylabs.datadoghq.com/cloud-security-atlas/vulnerabilities/security-group-open-to-internet/.

2257. Stratus Red Team - User Guide, from https://stratus-red-team.cloud/user-guide/getting-started/.

2258. (2024), AWS - EC2 Persistence, from https://cloud.hacktricks.xyz/pentesting-cloud/aws-security/aws-persistence/aws-ec2-persistence.

2259. Environment variables to configure the AWS CLI, from https://docs.aws.amazon.com/cli/v1/userguide/cli-configure-envvars.html.

2260. Nestori Syynimaa, (2020), Quest for guest access: Azure Active Directory reconnaissance as a guest, from https://aadinternals.com/post/quest_for_guest/.

2261. (2023), Configure cryptographic key auto-rotation in Azure Key Vault, from https://learn.microsoft.com/en-us/azure/key-vault/keys/how-to-configure-key-rotation.

2262. Eng Soon Cheah, (2022), Enumerating subscription information with MicroBurst, from https://dev.to/cheahengsoon/enumerating-subscription-information-with-microburst-35a1.

2263. Jevon Davis, (2023), Securing Azure: Hunting with AzureHound, from https://infosecwriteups.com/securing-azure-hunting-with-azurehound-d7ebb58e0fde.

2264. (2023), Goblob - A Fast Enumeration Tool For Publicly Exposed Azure Storage Blobs, from https://www.kitploit.com/2023/11/goblob-fast-enumeration-tool-for.html.

2265. Eitan Shteinberg, (2024), Protect your storage resources against blob-hunting, from https://techcommunity.microsoft.com/t5/microsoft-defender-for-cloud/protect-your-storage-resources-against-blob-hunting/ba-p/3735238.

2266. (2023), Create, change, or delete a network security group, from https://learn.microsoft.com/en-us/azure/virtual-network/manage-network-security-group?tabs=network-security-group-cli.

2267. (2023), How network security groups filter network traffic, from https://learn.microsoft.com/en-us/azure/virtual-network/network-security-group-how-it-works.

2268. (2024), Abusing Managed Identities, from https://hackingthe.cloud/azure/abusing-managed-identities/#:~:text=the%20same%20account.,Exploiting%20Azure%20Managed%20Identity,used%20to%20authenticate%20to%20Azure.

2269. Az role definition, from https://learn.microsoft.com/en-us/cli/azure/role/definition?view=azure-cli-latest.

2270. (2023), List Azure role definitions, from https://learn.microsoft.com/en-us/azure/role-based-access-control/role-definitions-list.

2271. Az network vnet peering, from https://learn.microsoft.com/en-us/cli/azure/network/vnet/peering?view=azure-cli-latest.

2272. (2024), What is Azure Virtual Network?, from https://learn.microsoft.com/en-us/azure/virtual-network/virtual-networks-overview.

2273. Gabe Thompson, (2021), Enumerating the Google Cloud Platform (GCP), from https://gmbeltwarrior.medium.com/enumerating-the-google-cloud-platform-gcp-a580da510a23.

2274. Liat Vaknin, (2022), Google Cloud Storage Explorer: Enumerating Google Cloud's Bucket Access Permissions, from https://orca.security/resources/blog/google-cloud-platform-storage-explorer/.

2275. gcloud auth print-access-token, from https://cloud.google.com/sdk/gcloud/reference/auth/print-access-token.

2276. Privilege Escalation in Google Cloud Platform – Part 2 (Non-IAM), from https://rhinosecuritylabs.com/cloud-security/privilege-escalation-google-cloud-platform-part-2/#gcp-privesc-scanner.

2277. (2024), GCP - IAM Post Exploitation, from https://cloud.hacktricks.xyz/pentesting-cloud/gcp-security/gcp-post-exploitation/gcp-iam-post-exploitation.

2278. kubectl Quick Reference, from https://kubernetes.io/docs/reference/kubectl/quick-reference/.

2279. (2024), Amazon EKS: User Guide, from https://docs.aws.amazon.com/pdfs/eks/latest/userguide/eks-ug.pdf#install-kubectl.

2280. Jack Roper, (2024), Kubectl Cheat Sheet – 15 Kubernetes Commands & Objects, from https://spacelift.io/blog/kubernetes-cheat-sheet.

2281. Docker ps command, from https://tecadmin.net/tutorial/docker/docker-ps-command/.

2282. docker container exec, from https://docs.docker.com/reference/cli/docker/container/exec/#:~:text=The%20docker%20exec%20command%20runs,working%20directory%20of%20the%20container.

2283. (2024), Pentesting Docker, from https://book.hacktricks.xyz/network-services-pentesting/2375-pentesting-docker.

2284. Kavish Tyagi, (2019), Lxd Privilege Escalation, from https://www.hackingarticles.in/lxd-privilege-escalation/.

2285. (2024), lxd/lxc Group - Privilege escalation, from https://book.hacktricks.xyz/linux-hardening/privilege-escalation/interesting-groups-linux-pe/lxd-privilege-escalation.

2286. Alex Kaluski, Cloud Discovery: Find and keep track of cloud privileged accounts, from https://delinea.com/blog/cloud-discovery-find-and-keep-track-of-cloud-privileged-accounts.

2287. Ashish Gavali, (2023), Audit AWS Cloud Security using ScoutSuite, from https://medium.com/globant/audit-aws-cloud-security-using-scoutsuite-4bc9073d2fc4.

2288. (2023), Use Image Integrity to validate signed images before deploying them to your Azure Kubernetes Service (AKS) clusters (Preview), from https://learn.microsoft.com/en-us/azure/aks/image-integrity?tabs=azure-cli.

2289. Fabien Soulis, Hardening Container Images: Best Practices and Examples for Docker, from https://medium.com/@SecurityArchitect/hardening-container-images-best-practices-and-examples-for-docker-e941261cab13#:~:text=Use%20Minimal%20Base%20Images&text=Smaller%20images%20contain%20fewer%20components%2C%20reducing%20the%20potential%20attack%20surface.&text=Tips%20%3A%20For%20maximum%20stability%20of,and%20predictability%20of%20your%20images.

2290. (2024), What is Container Security?, from https://www.wiz.io/academy/what-is-container-security.

2291. Restricting Application Capabilities Using Seccomp, from https://docs.openshift.com/container-platform/3.11/admin_guide/seccomp.html.

Module 20: Cryptography

2292. (1999), Cracking S/MIME encryption using idle CPU time, from https://securiteam.com/tools/3J5PRCODPO/.

2293. (2001), Announcing the ADVANCED ENCRYPTION STANDARD (AES), from https://csrc.nist.gov/csrc/media/publications/fips/197/final/documents/fips-197.pdf.

2294. John Talbot and Dominic Welsh, (2006), Complexity and Cryptography an introduction, from https://www.cambridge.org/gb/academic/subjects/mathematics/discrete-mathematics-information-theory-and-coding/complexity-and-cryptography-introduction?format=PB&isbn=9780521617710.

2295. Bruce Schneier, Applied Cryptography, Second Edition, from https://www.schneier.com/books/applied_cryptography/.

2296. Josh Ryder, Introduction to Encryption, from https://www.developer.com/tech/article.php/630681.

2297. Digital Certificates, from https://www.bitpipe.com/tlcl/Digital-Certificates.html.

2298. Vijay Bollapragada, Mohamed Khalid, and Scott Wainner, IPSec Authentication and Authorization Models, from http://www.ciscopress.com/articles/article.asp?p=421514&seqNum=4%20-%2031k%20-.

2299. SHA (Secure Hash Algorithm), from http://safeexm.safescrypt.com/SafeDoXX_User_Manual.pdf.

2300. PGP Attack FAQ: The asymmetric cipher, from https://www.iusmentis.com/technology/encryption/pgp/pgpattackfaq/asymmetric/.

2301. What is Public-Key Cryptography?, from http://www.x5.net/faqs/crypto/q3.html.

2302. The Heartbleed Bug, from http://heartbleed.com/.

2303. Dr. B. R Gladman, (2000), The Regulation of Investigatory Powers Bill - The Provisions for Government Access to Keys, from https://www.fipr.org/rip/RIPGAKBG.pdf.

2304. YongBin Zhou and DengGuo Feng, Side-Channel Attacks: Ten Years after Its Publication and the Impacts on Cryptographic Module Security Testing, from https://csrc.nist.gov/csrc/media/events/physical-security-testing-workshop/documents/papers/physecpaper19.pdf.

2305. Van Geelkerken F.W.J, (2006), Digital Mixing (MIX nets), from https://www.iusmentis.com/society/privacy/remailers/onionrouting/.

2306. Josh Ryder, (2000), Introduction to Encryption, from https://www.developer.com/tech/article.php/630681/Introduction-to-Encryption.htm.

2307. Yuan Xue, (2009), Digital Signature, from https://tao.truststc.org/Members/yuanxue/network_security/Public resources/lecture12.

2308. Email Security, from https://www.livingentenet.com/e/es.htm.

2309. Hash-based message authentication code, from https://en.wikipedia.org/wiki/Hash-based_message_authentication_code.

2310. Margaret Rouse, (2010), Hash-based Message Authentication Code (HMAC), from http://searchsecurity.techtarget.com/definition/Hash-based-Message-Authentication-Code-HMAC.

2311. How and when do I use HMAC?, from https://security.stackexchange.com/questions/20129/how-and-when-do-i-use-hmac/20301.

2312. Krishna Gehlot, (2015), Message Authentication Code & HMAC, from, https://www.slideshare.net/slideshow/message-authentication-code-hmac/48533320.

2313. Shaanan Cohney, Nadia Heninger, Matthew D. Green, (2017), The DUHK Attack, from https://duhkattack.com/.

2314. Gurubaran S, (2017), DUHK Attack allows Hackers to Recover Encryption Keys and Decrypt Communications Passing Over VPN, from https://gbhackers.com/duhk-attack-decrypt-communications-vpn/.

2315. Mohit Kumar, (2017), DUHK Attack Lets Hackers Recover Encryption Key Used in VPNs & Web Sessions, from https://thehackernews.com/2017/10/crack-prng-encryption-keys.html.

2316. Catalin Cimpanu, (2017), DUHK Crypto Attack Recovers Encryption Keys, Exposes VPN Connections, More, from https://www.bleepingcomputer.com/news/security/duhk-crypto-attack-recovers-encryption-keys-exposes-vpn-connections-more/.

2317. John Leyden, Thomas Claburn and Chris Williams, (2017), 'First ever' SHA-1 hash collision calculated. All it took were five clever brains... and 6,610 years of processor time, from https://www.theregister.com/2017/02/23/google_first_sha1_collision/.

2318. Collision attack, https://en.wikipedia.org/wiki/Collision_attack.

2319. Hash Collision Attack, https://privacycanada.net/hash-functions/hash-collision-attack/.

2320. Threefish from https://www.schneier.com/academic/skein/threefish.html.

2321. (2018), Threefish, from https://en.wikipedia.org/wiki/Threefish.

2322. (2017), Threefish Block Cipher, from https://tinycrypt.wordpress.com/2017/01/07/asmcodes-threefish/.

2323. (2019), Serpent (cipher), from https://en.wikipedia.org/wiki/Serpent_(cipher).

2324. Serpent Algorithm, from http://en.kryptotel.net/serpent.html.

2325. (2019), CAST-128, from https://en.wikipedia.org/wiki/CAST-128.

2326. (2019), Camellia (cipher), from https://en.wikipedia.org/wiki/Camellia_(cipher).

2327. (2020), GOST (block cipher), from https://en.wikipedia.org/wiki/GOST_(block_cipher).

2328. Muhammad Iqbal, Yudi Sahputra, and Andysah Putera Utama Siahaan, (2016), The Understanding of GOST Cryptography Technique, from https://www.researchgate.net/publication/308061220_The_Understanding_of_GOST_Cryptography_Technique.

2329. (2019), YAK(cryptography), from https://en.wikipedia.org/wiki/YAK_(cryptography).

2330. Mohsen Toorani, (2015), Cryptanalysis of a robust key agreement based on public key authentication, from https://onlinelibrary.wiley.com/doi/full/10.1002/sec.1373.

2331. Margaret Rouse, Elliptical Curve Cryptography (ECC), from https://www.techtarget.com/searchsecurity/definition/elliptical-curve-cryptography.

2332. (2018), Elliptic Curve Cryptography, from https://www.keycdn.com/support/elliptic-curve-cryptography.

2333. Maria Korolov and Doug Drinkwater, (2019), What is quantum cryptography? It's no silver bullet, but could improve security, from https://www.csoonline.com/article/3235970/what-is-quantum-cryptography-it-s-no-silver-bullet-but-could-improve-security.html.

2334. (2020), Quantum cryptography, from https://en.wikipedia.org/wiki/Quantum_cryptography.

2335. Shai Halevi, (2011), Homomorphic Encryption, from https://www.iacr.org/conferences/crypto2011/slides/Halevi.pdf.

2336. (2019), What is the purpose of Homomorphic encryption?, from https://crypto.stackexchange.com/questions/62386/what-is-the-purpose-of-homomorphic-encryption.

2337. Casey Crane, (2019), What is Homomorphic Encryption?, from https://www.thesslstore.com/blog/what-is-homomorphic-encryption/.

2338. Hardware-based Encryption Devices, from https://sourcedaddy.com/aplus/hardware-based-encryption-devices.html.

2339. Darril, TPM and HSM Hardware Encryption Devices, from https://blogs.getcertifiedgetahead.com/tpm-hsm-hardware-encryption-devices/.

2340. Hardware Security Modules (HSMs), from https://cpl.thalesgroup.com/encryption/hardware-security-modules.

2341. Crypto USB - What is AES 256-Bit Hardware-Based Encryption?, from https://incegralmemory.com/faq/what-aes-256-bit-hardware-based-encryption.

2342. (2020), GNU Privacy Guard, from https://en.wikipedia.org/wiki/GNU_Privacy_Guard.

2343. GNUPG (GPG), from http://www.alt.es/producto/goanywhere/products/director/encryption/gpg.html.

2344. NanoDano, (2018), GPG Tutorial, from https://www.devdungeon.com/content/gpg-tutorial.

2345. (2019), Web of trust, from https://en.wikipedia.org/wiki/Web_of_trust.

2346. (2019), Related-key attack, from https://en.wikipedia.org/wiki/Related-key_attack.

2347. Alex Biryukov and Dmitry Khovratovich, Related-key Cryptanalysis of the Full AES-192 and AES-256, from https://eprint.iacr.org/2009/317.pdf.

2348. (2019), Padding oracle attack, from https://en.wikipedia.org/wiki/Padding_oracle_attack.

2349. Nimrod Aviram, Sebastian Schinzel, Juraj Somorovsky, Nadia Heninger, Maik Dankel, Jens Steube, Luke Valenta, David Adrian, J. Alex Halderman, Viktor Dukhovni, Emilia Käsper, Shaanan Cohney, Susanne Engels, Christof Paar, and Yuval Shavitt, (2016), The DROWN Attack, from https://drownattack.com/.

2350. (2019), DROWN attack, from https://en.wikipedia.org/wiki/DROWN_attack.

2351. Ambika Choudhury, (2020), Quantum-Proof Cryptography and Its Role in Security, from https://analyticsindiamag.com/quantum-proof-cryptography-its-role-in-security/.

2352. Leading in Quantum-Safe Standards, from https://www.post-quantum.com/.

2353. (2022), Post-Quantum Cryptography, from https://en.wikipedia.org/wiki/Post-quantum_cryptography.

2354. (2022), Lightweight Cryptography, from https://csrc.nist.gov/projects/lightweight-cryptography.

2355. Cihangir Tezcan, (2020), Lightweight Crypto for IoT - 1.1: Network of Things, from https://www.youtube.com/watch?v=KBORs9yrhmQ&ab_channel=CihangirTezcan.

2356. William J. Buchanan, (2017), Lightweight Cryptography Meth, from https://www.tandfonline.com/doi/full/10.1080/23742917.2017.1384917.

2357. Block Cipher Modes of Operation, from https://www.tutorialspoint.com/cryptography/block_cipher_modes_of_operation.htm.

2358. (2022), Block Cipher Mode of Operation, from https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#Cipher_block_chaining_(CBC).

2359. Kattamuri Meghna, (2021), Block Cipher Modes of Operation, from https://www.geeksforgeeks.org/block-cipher-modes-of-operation/.

2360. Shreedarshan K, (2019), Block Cipher Modes of Operation: Explanation of all 4 types | Cryptography and Network Security, from https://www.youtube.com/watch?v=A0CVudRYzuY.

2361. (2022), Authenticated Encryption, from https://en.wikipedia.org/wiki/Authenticated_encryption#Authenticated_encryption_with_associated_data_(AEAD).

2362. Toshendra Kumar Sharma, Public vs. Private Blockchain: A Comprehensive Comparison, from https://www.blockchain-council.org/blockchain/public-vs-private-blockchain-a-comprehensive-comparison/.

2363. Naveen Joshi, (2020), Public vs Private Blockchain: Who Wins?, from https://www.allerin.com/blog/public-vs-private-blockchain-who-wins.

2364. (2019), Public vs. Private Ledger, from https://www.mme.ch/en/magazine/articles/public-vs-private-ledger.

2365. Akash Takyar, Blockchain Technology Explained, from https://www.leewayhertz.com/blockchain-technology-explained/.

2366. (2019), What Different Types of Blockchains are There?, from https://dragonchain.com/blog/differences-between-public-private-blockchains.

2367. (2018), Types of Blockchain, from https://www.youtube.com/watch?v=uiHbiPZlT1k&ab_channel=Teusko.

2368. (2021), Encrypt Email Messages, from https://support.microsoft.com/en-us/office/encrypt-email-messages-373339cb-bf1a-4509-b296-802a39d801dc.

2369. Mithilesh Tata, (2021), How to Encrypt Email in Outlook, from https://www.arysontechnologies.com/how-to/encrypt-outlook-email.

2370. Nicky Mathew, (2021), Encrypt Email Messages in Outlook – Methods to Encrypt, from https://bobcares.com/blog/encrypt-email-messages-in-outlook.

2371. River Hart, (2022), How to Encrypt Outlook Emails?, from https://proprivacy.com/email/guides/encrypt-outlook-email.

2372. Quantum Cryptanalysis, from https://www.pqsecurity.com/quantum-cryptanalysis/.

2373. (2022), Quantum Cryptanalysis: Hype And Reality Of The Threat, from https://www.insidequantumtechnology.com/news-archive/quantum-cryptanalysis-hype-and-reality-of-the-threat/.

2374. (2021), Side-channel attacks explained: everything you need to know, from https://www.rambus.com/blogs/side-channel-attacks/#countermeasures.

2375. Alex Zhang, (2022), What Is a Side-Channel Attack & What Countermeasures Exist?, from https://blog.enconnex.com/what-is-a-side-channel-attack-vulnerabilities-and-countermeasures.

2376. Hacken and Barwikowski Bartosz, (2024), 51% Attack: The Concept, Risks & Prevention, from https://hacken.io/discover/51-percent-attack/.

2377. (2024), 51% Attack: Definition, Who Is At Risk, Example, and Cost, from https://www.investopedia.com/terms/1/51-attack.asp#:~:text=A%2051%25%20attack%20is%20an%20attack%20on%20a%20blockchain%20by,other%20miners%20from%20completing%20blocks.

2378. (2023), Finney Attacks in Cryptocurrency, from https://www.immunebytes.com/blog/finney-attacks-in-cryptocurrency/.

2379. (2024), Finney Attack And Ways It Affects The Cryptoecosystem, from https://steemit.com/hive-150122/@rmeto445/finney-attack-and-ways-it-affects-the-cryptoecosystem

2380. (2023), What is an Eclipse Attack?, from https://www.gate.io/learn/articles/what-is-an-eclipse-attack/952.

2381. (2024), What is Race Attack?, from https://www.bitdegree.org/crypto/learn/crypto-terms/what-is-race-attack.

2382. Haseem Alhalabi, (2024), What is a DeFi Sandwich Attack?, from https://b2binpay.com/en/what-is-a-defi-sandwich-attack/.

2383. Holger Schulze, (2023), Quantum Computing Threats: A How-to Guide for Preparing Your Company's Cybersecurity Defenses, from https://www.linkedin.com/pulse/quantum-computing-threats-how-to-guide-preparing-your-holger-schulze/.

2384. (2024), What are the potential threats and vulnerabilities of quantum computing for cybersecurity?, from https://www.linkedin.com/advice/1/what-potential-threats-vulnerabilities-quantum