

Internet Security



Internet Security is a specialized form of computer security that safeguards ^{connected} Internet systems, including devices, networks, and data, from cyber threats like malware, phishing and unauthorized access. It employs a range of tools, policies, and protocols, such as encryption, firewalls and antivirus software to protect online activities and ensure the confidentiality, integrity and availability of information exchanged over the internet.

What Involves

- Protecting users and systems.
- Securing Data and Transactions
- Preventing Cyber Threats: It aims to prevent various threats.
 - Malware
 - Phishing
 - Hacking
 - Identity Theft.

Key components of Internet Security

- Encryption
- Firewalls
- Antivirus Software
- Identity and Access Management
- Data Loss Prevention

Types of Internet Security focus on protecting networks, data, applications and user identities by categorizing ^{security} security measures like network security (firewalls, VPNs), cloud security

AD

(Protecting cloud infrastructure), application security (securing software), endpoint security (protecting devices like phones and computers), identity and access management (MFA, access controls), information security (data loss prevention, data encryption), operational security (securing daily processes) and IoT security (for connected devices)

Examples

Firewalls: Control incoming and outgoing network traffic.
Virtual Private Networks (VPNs): Create encrypted, secure connections for remote access.
Intrusion Prevention System (IPS): Monitor network activity for suspicious patterns and prevent attacks.

cloud security

What it is: Secures data, applications, and infrastructure hosted in cloud environments.

Includes: Protecting against unauthorized access to cloud computing.

Endpoint Security

Involves protecting individual devices, or "endpoints", such as laptops, smartphones, and servers, from threats.

Application Security

Addresses vulnerabilities in web and other applications, which are often prime targets for cyberattacks.

Data Security

Encompasses a range of strategies and technologies to protect data's confidentiality, integrity and availability.

Identity and Access Management

~~controls~~

Controls who has access to what resources and ensures that users are who they claim to be, often using multifactor

authentication (MFA).

Operational Security

Protects against threats that could compromise operational processes and the critical infrastructure that supports them, accounts, data, value, & people.

Authentication and Authorization

Authentication is the process of verifying that a user is who they claim to be, often by checking their credentials like a username and password. Authorization is the subsequent process of determining what actions and resources that authenticated user is permitted to access within a system or application.

Authentication always comes first as you we must know who a user is before you can decide what they are allowed to do.

Authentication

Purpose - Verification of user identity.

Principle - users provide credentials (like an username and password, a finger print, or a one time PIN) which the system validates to confirm they are who they say they are.

Example - When you login to our e-mail, we will provide our userid and password for successful login.

Authorization

Purpose - To grant or deny access to specific resources or actions after authentication.

How it works - Based on the authenticated user's identity and role, the system checks predefined permissions to determine what they can see or do.

Ex. After authenticating into your email, authorization determines which e-mails you can open, what you can write, and whether you can delete them, often based on your settings.

Firewalls Protect networks, encryption scrambles data for Privacy, and decryption makes it readable again. SSL is a protocol that uses encryption and decryption to secure web traffic by creating a private link between a browser and server preventing attacks. Firewalls can perform SSL decryption to inspect encrypted traffic for threats, then re-encrypt it before it leaves the network.

Firewalls

Function : Firewalls are network security devices that monitor and control incoming and outgoing network traffic based on predefined security rules.

Purpose : They act as a barrier between trusted internal networks and untrusted external networks (like the internet), preventing unauthorized access and malicious threats.

Encryption & Decryption

Encryption - The process of converting plain text data into an unreadable format called ciphertext using an algorithm and a key.

Decryption - The process of converting ciphertext back into its original plain text form, which requires the correct decryption key.

Purpose - They act as a barrier between trusted internal networks and untrusted external networks (internet), preventing unauthorized access and malicious threats.

Encryption and decryption

Purpose - To ensure data confidentiality and prevent unauthorized parties from understanding sensitive information during transmission.

SSL (Secure Sockets Layer)

Function : A security protocol that establishes an encrypted link between a client (like a web browser) and a server.

Mechanism : It uses a combination of asymmetric (public-key) and symmetric encryption to securely ~~an~~ exchange data.

Benefits : Provides security for data transferred between web browsers and servers, ensuring that the data remains private and secure from attackers.

SSL Decryption with Firewalls

Purpose : To enable deep packet inspection of encrypted traffic, which often contains threats that traditional firewalls cannot see.

Process : A firewall intercepts encrypted traffic, decrypts it using the server's private key or its own certificate, inspects the content for malicious activity or policy violations, and then re-encrypts the traffic before sending it to its destination.

Types

SSL Forward Proxy : Used for decrypting outbound traffic from internal users to external servers.

SSL Inbound Inspection : Used for decrypting incoming traffic targeting internal servers, often requiring the server's private key.

SS

Java Script

<html>

<head> -- </head>

<body>

<script>

// statements

</script>

</body>

</html>

body section

Head section

<html>

<head>

<script>

// statements

</script>

</head>

<body>

</body>

</html>

Using variables

<html>

<head>

<script>

var n = 5;

document.write(x);

</script>

</html>

O/p - 5

10/

To add two numbers

```
<html>
<head>
<script>
var x=5;
var y=10;
var z=x+y;
document.write(z);
</script>
</head>
</html>
```

Type conversion

eval() function is used for conversion string value to numeric value.
parseInt() function is used for conversion of string to integer value.

Ex-1

```
<html>
<head>
<script>
var exp = prompt("please enter an expression");
alert("Result of Expression " + exp + " is " + eval(exp));
</script>
<body>
</body>
</html>
```

Ex-2

```
<html>
<head>
<script>
function factorial(n)
{
var f=1;
for (var i=1; i<=n; i++)
{
f=f*i;
}
return f;
}
```

```

var a = prompt ("Enter a Number", "");
var fact = factorial(a);
alert ("Factorial of " + a + " is " + fact);
</script>
</head>
<body></body>
</html>

```

Using Conditional statements :

```

<html>
<head>
<script>
var x = 5;
if (x < 10)
{
    document.write(x); // console.log(x);
}
</script>
</head>
</html>

```

Java Script Built-in Functions

alert (string or variable) : To give a message.
Prompt ("message", "default value") : To give a message and get value in the form of string and pass it some variable.
confirm ("message") : used to confirm something from user by giving a confirmation dialog box with command buttons OK and cancel.

ex

```

<html>
<head>
<title> Built in function </title>
</head>
<body>
<script language = "JavaScript">

```

```

alert ("welcome to Javascript");
var name = Prompt ("please Enter your name", "");
if (confirm ("should I wish you")) {
    alert ("Dear "+name+" Thanks for using Javascript");
}
else {
    alert ("Try next time");
}
</script>
</body>
</html>

```

Array

arrayname = new array (Array length)
 or
 arrayname = new array, (Array length)

```

<html>
<head>
<title> array elements </title>
</head>
<body>
<script language = "java script">
    a = new array (5)
    a[0] = "AB"
    a[1] = "CD"
    a[2] = "EF"
    document.write (a[1]);
</script>
</Body>
</html>

```