HEXAWARE

Security

Need for Security

Business Challenges Around Security

- Security as an add-on
- Architectural inefficiencies
- Failure to update the Security patches
- Lack of proper auditing practices
- No strong identification and verification practices
- Security policies and control
- Lack of expertise
- Awareness
- Management priority

Security

- To build an application with end-to-end security encompasses all IT system's infrastructure components such as host, applications, users, network devices, client applications, communications etc.

Infrastructure component classification

- The Network Services
- The host operating system
- The target Application

Networking Services

- A Network is a group of computers or information devices and associated peripherals connected by a communication channel capable of sharing information and resources between computers, applications, users and other networks.
- Typical Network components are routers, switches and firewalls.
- Security contributors are
  - ➢ Intrusion Deduction System (IDS)
  - ➢ Router Access Control Lists (ACL)
  - ➢ Virtual Private Networks (VPN)
  - ➢ SSL / Cryptographic accelerator appliances.

Host Operating System

Vital roles of OS:

- Executing
- Managing
- Controlling hardware software applications
- Interacting with other hosts
- Interacting with network-related applications.

Functionalities and services of OS:

- Tools
- administration support utilities
- Application deployment
- End user

Security Threats in OS

Problems with Default configuration of OS:

- Exploitation and attack by hackers
- Information theft
- Spreading Viruses
- Trojan Horse
- Software Buffer Over Flow
- Password cracking

Implementing OS security policy

Updating the security patches and updates

Disabling unused ports and services

Eliminating non essential tools and utilities

The Applications or services

Security Flaws and Exploits:

Input validation failure

Output Sanitation

Buffer Overflow

Data Injection Flaws

Cross Side Scripting

Improper Error Handling

Insecure Data Transit or storage

Weak Session Identifier

Weak Security Tokens

Security Flaws and Exploits:

Weak Password Exploits

Weak Encryption

Session Theft

Insecure Configuration Data

Broken Authentication

Broken Access Control

Policy Failure

Audit and Logging Failures

Multiple Sign – On Issues

Deployment problem

Coding Problem

Four W's

To provide end-to-end

- Which application are we protecting?
- Who are we protecting?
- Where should we protect them?
- Why are we protecting them?

Solution for Application Security Threats

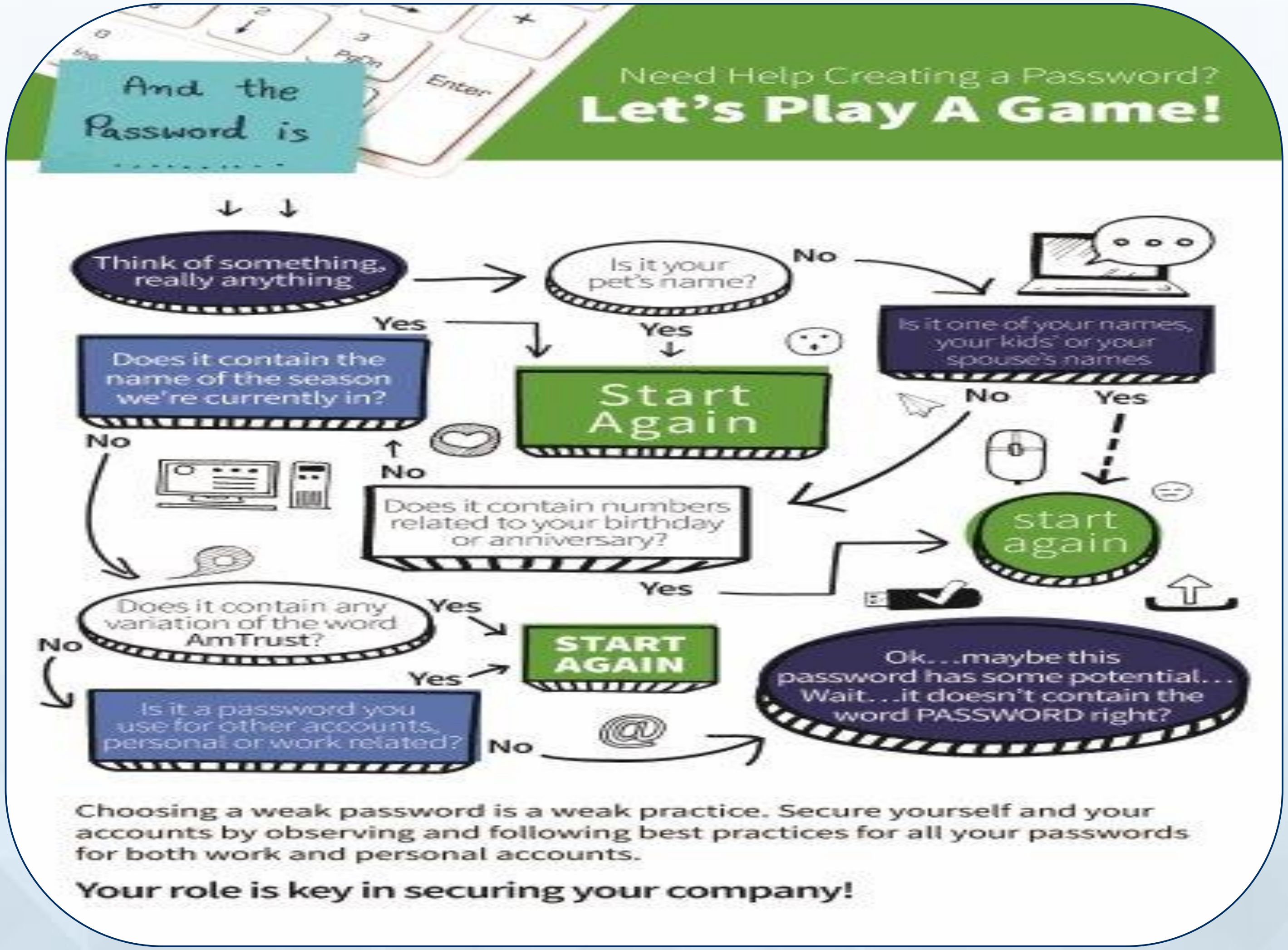
Based on the Four w’s developer should develop the application by following Best practice for coding, Design pattern, Reality check, Proactiv Assessment, Profiling, Defensive Strategies, Recovery and continuity strategies etc.

Secured Application/ Web Application

A secured Application/Web Application should serves the below five goal.

- Confidentiality
- Integrity
- Authentication
- Authorization

Web Application Security - Authentication



And the Password is ....

Need Help Creating a Password?

Let's Play A Game!

Think of something, really anything

Is it your pet's name?

No

Yes

Start Again

Is it one of your names, your kids' or your spouse's names

No

Yes

start again

Yes

Does it contain the name of the season we're currently in?

No

No

Does it contain numbers related to your birthday or anniversary?

Yes

Does it contain any variation of the word AmTrust?

Yes

No

START AGAIN

Yes

Is it a password you use for other accounts, personal or work related?

No

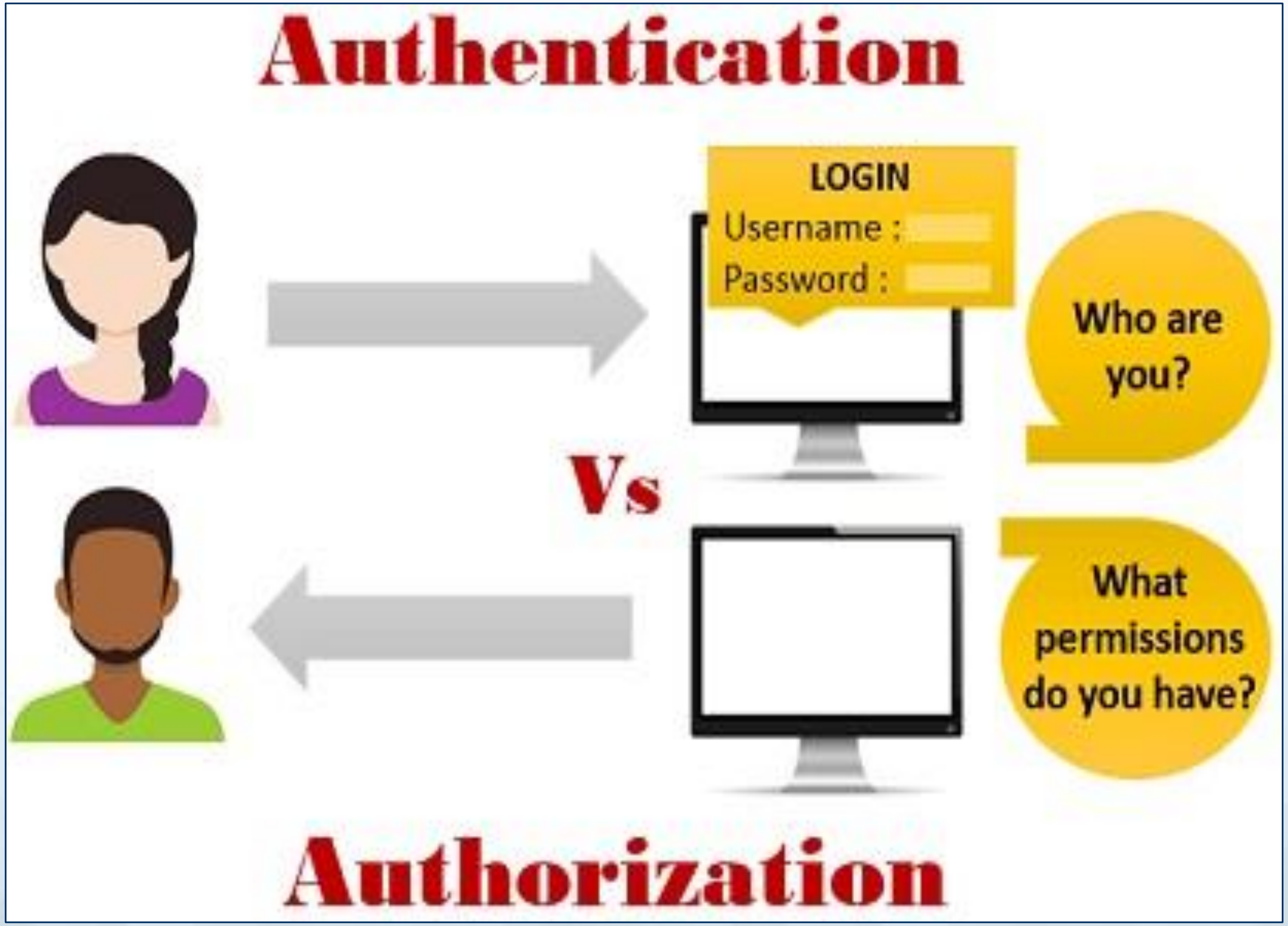
Ok...maybe this password has some potential... Wait...it doesn't contain the word PASSWORD right?

Choosing a weak password is a weak practice. Secure yourself and your accounts by observing and following best practices for all your passwords for both work and personal accounts.

Your role is key in securing your company!

In Computing the process to verify the identity of a user to ensures the user is recognized by the Organization

Web Application Security - Authorization



Authorization is the processes of specifying access rights/privileges to a person on a resources

Web Application Security - Confidentiality

The state of maintaining secret.

Web Application Security - Integrity

Data integrity is the maintenance, and assurance of the accuracy and consistency of data over its entire life-cycle,

Http Authentication

Http protocol provides build in authentication support :

➔basic authentication

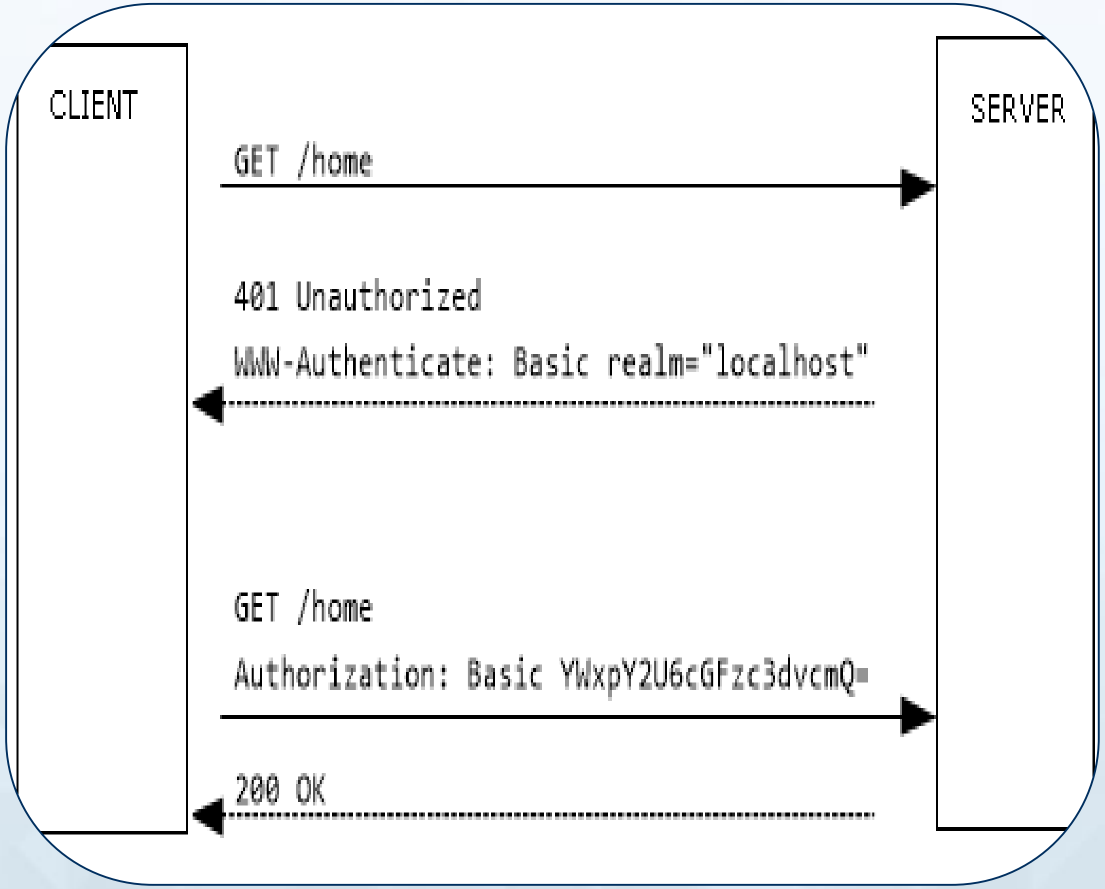
➔Digest authentication

Web server will maintain the username and password for authentication

Basic Authentication

Basic

The username and password are sent as an unencrypted base64 encoded text. Since the password is not encrypted, HTTPS should be used to send the URL.



CLIENT

SERVER

GET /home

401 Unauthorized
WWW-Authenticate: Basic realm="localhost"

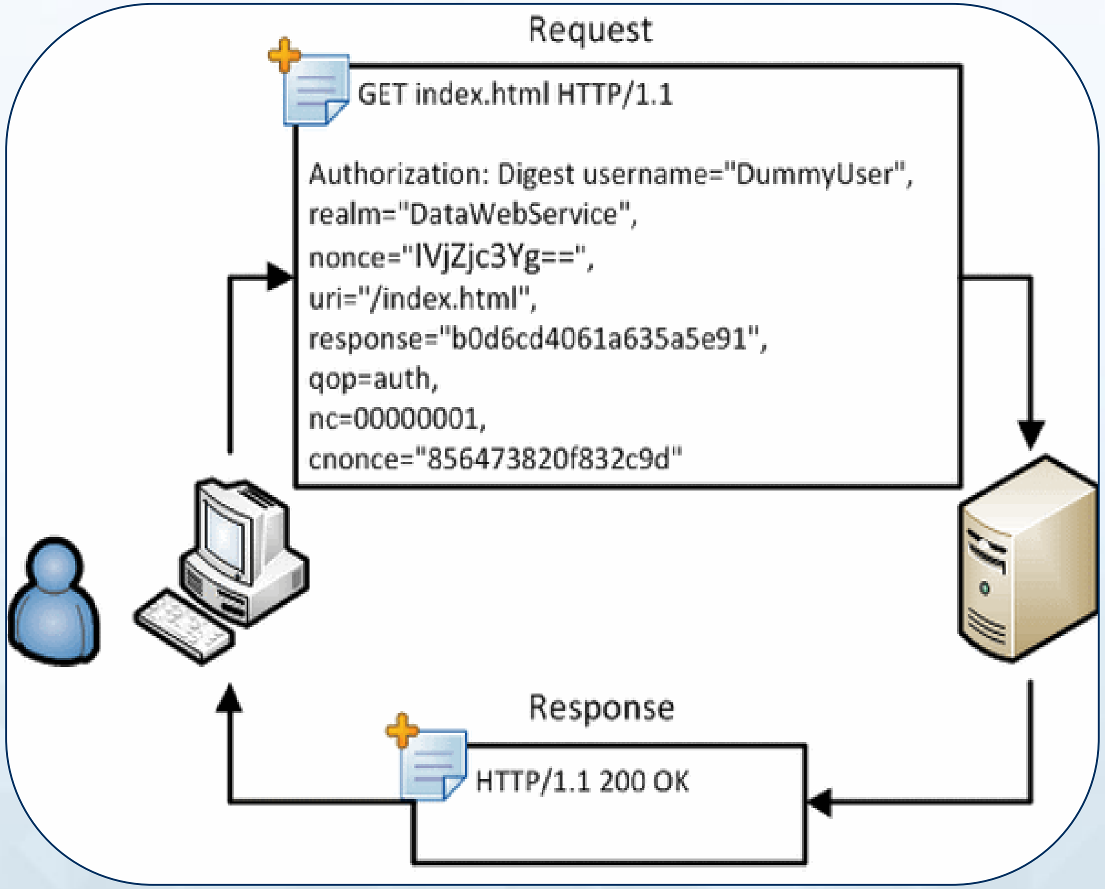
GET /home
Authorization: Basic YWxpY2U6cGFzc3dvcmQ=

200 OK

Digest Authentication

Digest

The credentials are passed to the server in hashed form. Although the credentials cannot be captured over HTTP, the request can be replayed using the hashed credentials.



```
GET index.html HTTP/1.1

Authorization: Digest username="DummyUser",
realm="DataWebService",
nonce="IVjZjc3Yg==",
uri="/index.html",
response="b0d6cd4061a635a5e91",
qop=auth,
nc=00000001,
cnonce="856473820f832c9d"
```

```
HTTP/1.1 200 OK
```

OpenID and OAuth

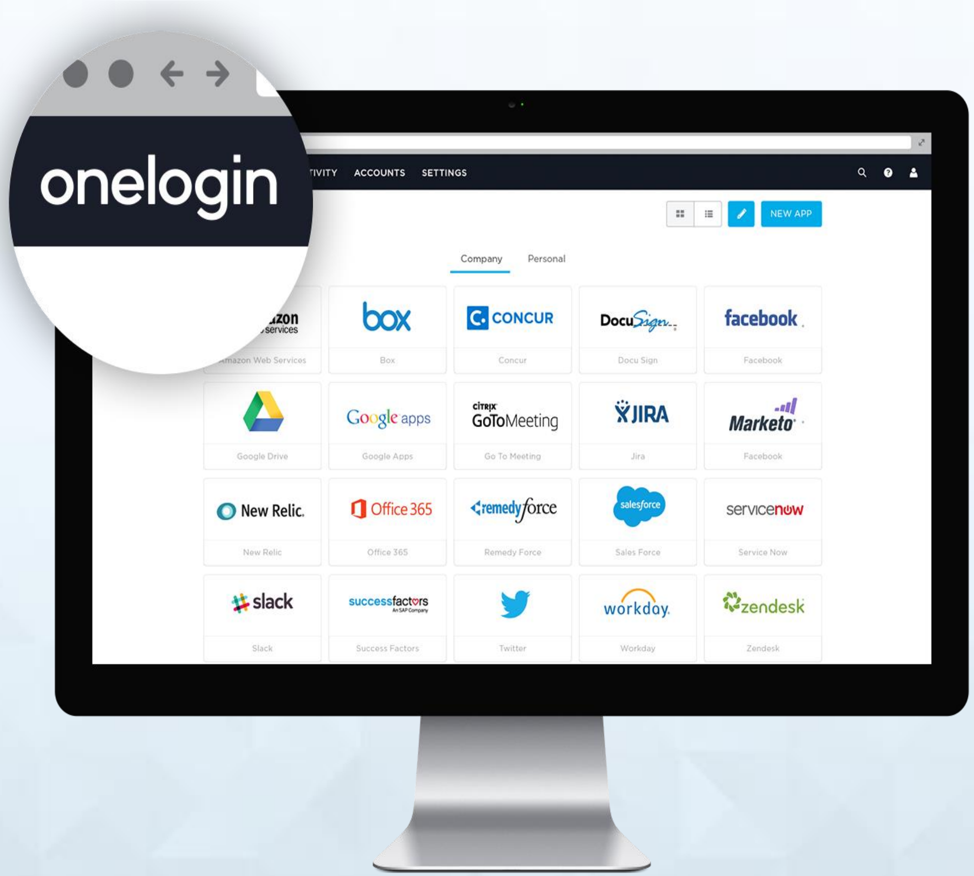
- OpenID Connect (OIDC) is an authentication protocol, based on the OAuth 2.0 family of specifications. It uses simple JSON Web Tokens (JWT), which you can obtain using flows conforming to the OAuth 2.0 specifications.

- While OAuth 2.0 is about resource access and sharing, OIDC is all about user authentication. Its purpose is to give you one login for multiple sites. Each time you need to log in to a website using OIDC, you are redirected to your OpenID site where you login, and then taken back to the website.

- For example, if you chose to sign in to Auth0 using your Google account then you used OIDC. Once you successfully authenticate with Google and authorize Auth0 to access your information, Google will send back to Auth0 information about the user and the authentication performed. This information is returned in a JWT. You'll receive an Access Token and, if requested, an ID Token.

SSO

Single sign-on users only have to enter one set of credentials to access their web apps in the cloud and behind the firewall – via desktops, smartphones and tablets.



HEXAWARE

Innovative Services

Passionate Employees

Delighted Customers

Thank you

www.hexaware.com