Assignment 3

SENG2250

Sam Dolbel – 3130069

Network Security

a)

- When an employee is away from the business, they need to establish a secure connection to the company server. Depending on the key exchange protocol used for the client-server connection, the connection may be vulnerable to a **Man-In-The-Middle** attack.
  The main way to prevent a MITM attack is to prevent malicious users from entering access points in the first place. Strong protection/encryption on router access points can prevent attackers from working, as can Public Key authentication like RSA.

- Since the bank is dealing with sensitive financial information, customers and employees will be required to use secure passwords to protect themselves. Passwords are susceptible to a few attacks, the simplest being the **Brute Force** attack, where the attacker attempts to guess the password.
  Brute force attacks rely on being able to attempt millions of passwords in a short space of time. This is generally curtailed by imposing limits on how many wrong guesses the user can make. In this solution, the account is locked if more than a small number of wrong guesses are made, whether for 24 hours or indefinitely.

- The customer services themselves are vulnerable to a **DDoS (Distributed Denial of Service)** attack. This means that the bank's servers can be overwhelmed with malicious traffic from bots, blocking customers and employees from using vital services.
  The current solution to this attack is to employ anti-DDoS protection services like Cloudflare. These services detect suspicious traffic and block bots, keeping the server connection clearer.

b) To begin, this process should only be performed internally. Ideally, the process should be blocked to all users outside the network/IP. Without the added risk of establishing a secure external client-server connection, this curtails the danger of outside attackers Sniffing or performing a Man-In-The-Middle attack. This also means there should be no Back Door.
Next, the employee should input their credentials; user name and password. A secure, difficult-or-impossible-to-guess password will prevent access via a Dictionary attack, and wrong answer limitations prevent a Brute Force attack. Only with this two-factor authentication will the user be granted access.
Finally, the user will attempt to perform the action. Although the user has input the right credentials, they still need to be authorised – for example, a trainee or intern should have access restricted to some sensitive applications. The user sends their authorisation details to the server for analysis, using a service like OAuth or Kerberos. If the user is authorised, they can successfully update

c) SAML would be the better option for internal system access and authorisation for an employee. OAuth is ideal for granting limited access to a large external base, making it ideal for social networks and the like. Kerberos is extremely restrictive, forbidding authentication forwarding and forcing encryption even offline.
By contrast, SAML provides a single login for multiple applications – once the user is properly authenticated, they have access to all the internal applications. Once the user is authenticated, they can navigate the internal system with efficiency.

d) IPSec has the advantage of flexibility compared to its compatriots. As a Transport Layer protocol, IPSec can carry many different varieties of data. Additionally, the ability to operate in Tunnel Mode allows IPSec to completely obscure its IP address.

SSL Handshake Code

```
== Initial greeting ==
Client sends: Hello.
Server receives: Hello.

Server shares public key: 65537
Client acknowledges public key: 65537
```

```
== Start handshake ==
Server sends server ID: i00xjmaj99w2
Server receives session ID: 3280723325235
Client receives server ID: i00xjmaj99w2
Client receives session ID: 3280723325235

Client sends client ID: fa92naoc9amma
Server receives client ID: fa92naoc9amma

Server generates and sends RSA signature:
33f687f28a4e83058f2f32b3f50b808a56aceaf08e68a891258f3b61737852d024c49070cb73f20e34330a9aa245a1214f33ae03909c1e1fa4d0d557766190c71347f66e5177fd48148fae785b3f114b77a3dd1bc6
be271caed11750d4caaf45b2c5274e6ab5b519fa4b272289a231971ff19aec89586b846dc1b0f76b59b4b3ce2ef7602f0a117dccee8faba1d016cef82e89c3f3e01af2ae949fa79e8d782412664ee6808bd1d4431d
1721e3ecbf82f71be0dd6628d56923a21af44295e4b0f039b80e9f08f871243ca154d9153dcdec063f36d294284965e9631670a039339e0f0c37f2a8f503b11542cf2565a725358bccd5330ce668f3891b2da01fd9
3
Signature verified: true

Server sends Diffie-Hellman server public key:
102044864152708252563181806870918099912023180398453572352344202513204479163107305908092971451876446155986906654533065893023601572050448299892258623185398836839538827091818
816636179770936677039711899934822817952000195145919046994913018321212850285780121813337401472616334233875438089496711942376952970698226375 9
Server sends initiation vector: 3f176cef1204d3c7739acd6ceb49866d
Client receives Diffie-Hellman server public key:
102044864152708252563181806870918099912023180398453572352344202513204479163107305908092971451876446155986906654533065893023601572050448299892258623185398836839538827091818
816636179770936677039711899934822817952000195145919046994913018321212850285780121813337401472616334233875438089496711942376952970698226375 9
Client receives initiation vector: 3f176cef1204d3c7739acd6ceb49866d

Client sends Diffie-Hellman client public key:
116247519607903338687804173345648335571653856712464410650111180079582309287648235779301245985437795482224238110988092784108088728491287092313477522356511777701927739406965
301535126217253127960577476642175699307294575595450756353857723317439138121608920001148254512264480262863468674135387628337636695922186454
Server receives Diffie-Hellman client public key:
116247519607903338687804173345648335571653856712464410650111180079582309287648235779301245985437795482224238110988092784108088728491287092313477522356511777701927739406965
301535126217253127960577476642175699307294575595450756353857723317439138121608920001148254512264480262863468674135387628337636695922186454

Client combines and encrypts Diffie-Hellman public keys: afe5cdd21ddd582304ea0145c581fddb37919d295e12ddb91758499cc8503773
Server combines and encrypts Diffie-Hellman public keys: afe5cdd21ddd582304ea0145c581fddb37919d295e12ddb91758499cc8503773

== End handshake ==
```

```
== Send first message ==

Client encrypts a message: Clients are superior to a server, who should simply stop trying.
Client sends a message to server: 7e56c89c65b8c427826015707078d60ff517cec776963b94683f46817ccf73e4e71dd010f3f032cc8768529c385beee11acad8f960ce22ec13d4e78c57eff4b6
Server receives a message: 7e56c89c65b8c427826015707078d60ff517cec776963b94683f46817ccf73e4e71dd010f3f032cc8768529c385beee11acad8f960ce22ec13d4e78c57eff4b6
Server decrypts a message: Clients are superior to a server, who should simply stop trying.

== Send second message ==

Server encrypts a message: Servers are superior to a client, who should simply stop trying.
Server sends a message to client: 04d4013b074867bfbd0f7fd7f112c5baa049cef331989345c6e20ec1d6f2487b42613647846dc6fc1a05ee77207a200e3ed10c631d21f95189f6c792091209d7
Client receives a message: 04d4013b074867bfbd0f7fd7f112c5baa049cef331989345c6e20ec1d6f2487b42613647846dc6fc1a05ee77207a200e3ed10c631d21f95189f6c792091209d7
Client decrypts a message: Servers are superior to a client, who should simply stop trying.
```