BISM3205 - Assignment 2

Daniel Ciccotosto-Camp S4585727

Q1 (237 words)

Part A:

[E; 8]

Part B:

The Vigenere Cipher.

Part C:

The key is **ALEXBISM**.

Part D:

Using Crackstation.net, there is a known encryption of **universityofqueensland**.

Ordinarily, we would not be able to decrypt a hashed result besides using a brute force technique or looking up hashes in a repository, as we have done.

Part E:

The database administrator should implement salting and peppering for password storage. Salting adds a random string to a password before hashing, making it unlikely for attackers to find salted hashes in hash-lookup tables. Peppering involves storing the salt separately from the database, so if it's compromised, the salt remains secure. Additionally, the security leader should enforce password length and complexity requirements and check passwords against dictionaries and deny lists.

Part F:

The numbers were transformed to base 10 and then looked up on an ASCII table.

What is it called when 2 strings have the same hash digest?

This phenomenon is called a collision. As stated, this is when two *different* strings (or any serializable objects) generate the same hash value.

Part G:

The string of characters in the gif F5CA4F935D44B85C431A8BF788C0EACA is the same as its MD-5 hash f5ca4f935d44b85c431a8bf788c0eaca in hexadecimal form (Appendix 2).

Part H:

MD5 hash of the image plane: 253dd04e87492e4fc3471de5e776bc3d

MD5 hash of the image ship: 253dd04e87492e4fc3471de5e776bc3d

Both images have the same digest (Appendix 2).

Part I:

The fact that both images produce the same MD5 hash indicates that MD5 is vulnerable to collisions. This undermines the integrity and reliability of MD5 for ensuring data authenticity.

Part J:

This is a trick question. If I have encrypted the message with an asymmetric public key, then I need the private key to decrypt it.

Q2 (317 words)

Part A:

54

Part B:

1. I used a substitution cipher to substitute characters in the initial message as it looked like a website link. I could decode most of it except for the two capital letters UV. I assumed it would be QR (for QRcode) and wrote a script to be sure (Appendix 1).
2. I downloaded the zip file, extracted it, and found the QR code within it (Figure 1).

Figure 1: QR code found in QRcode.zip

3. I used a steganographic decoder and uploaded the QR code and navigated to the site alexpudmenzky.com/BISM3205/Message.au.
4. I uploaded the Message.au file to the steganographic decoder and retrieved the bits 10111010010010110.
5. The code in the audio file (174A0), when converted to binary is 10111010010100000.
6. I performed a bitwise XOR operation on the two binary numbers (same length and seemed appropriate) shown in Appendix 3. The result was 00000000000110110. Converting to decimal is 54, which is a single two-digit number, greater than 0, in decimal notation.

Part C:

An IP address is segmented into host and network portions via the use of a subnet mask, which is a series of bits which differentiate which section of the IP address represent the network and which part represent the host.

Bits in the subnet mask that are 1 represent the network portion, so the subnet mask 255.255.128, represented in binary form, i.e.

255.255.128.0 = 11111111.11111111.10000000.00000000 (binary)

Indicating that the first 17 bits represent the network portion of the IP address. A bitwise AND operation can be used to yield the relevant network bits.

11000000.10101000.00000101.00000010 (IP Address) AND

11111111.11111111.10000000.00000000 (Subnet mask)

11000000.10101000.00000000.00000000 (Network portion) = 192.168.0.0

Similarly, a bitwise NAND operation can be used to determine the computer (host) portion of the IP address:

11000000.10101000.00000101.00000010 (IP Address) NAND

11111111.11111111.10000000.00000000 (Subnet mask)

00000000.00000000.00000101.00000010 (Network portion) = 0.0.5.2

Part D:

This is more likely an encrypted message.

1. 127*6 bits – odd length of a hash. Typically an index of 2 – e.g 64, 128, 256, 512 (for SHA-256, e.g.)
2. Password hashes typically do not need to be this long – even a SHA-256 hash in base 64 would be 64 characters long.

Q3 (328 words excl. code)

Part A:

Similarities:

- Both use encryption to verify client requests and include the client’s username.
- Both interact with the KDC.

Differences:

- The rationale. A TGT sends only the username to the Authentication Server (AS) and verifies the client’s identity during login with client Kc. A Service Ticket request requires an encrypted TGT (by Ktgs) to request service-specific access and does not use the client’s password (uses session key provided with TGT).

Part B:

The client cannot decrypt the TGT or Service Ticket because they are encrypted with secret keys known only to the KDC.

The client can still use these tickets because they contain session keys - allowing the client to send encrypted authenticators with their username and timestamps to prove their identity.

Part C:

In a Kerberos environment, an attacker who retrieves a user's Ticket-Granting Ticket (TGT) may face challenges when authenticating to services.

1. If the attacker has the TGT but lacks the hashed password (Kc), they cannot decrypt the session key (Kctgs).
2. If they can access the Kctgs and send a request within the TGT's valid time frame, they could request a service ticket. However,

“If a ticket is compromised, it cannot be used outside of a specified time range -usually short enough to make the risk of a replay attack minimal.” (Microsoft 2009)

Part D:

Do (3). Plugging unknown devices into your computer could introduce malware or ransomware that compromises your company's network. IT have the expertise to investigate it safely.

Part E:

A false positive is an IDS alert for harmless activity, while a false negative misses a real attack. This is far more serious as actual threats go unnoticed, leaving the system vulnerable.

Part F:

An attacker could input the following:

```
SELECT *
FROM USERS
WHERE 1=1
AND Username = '<enteredUsername>'
Password = '<enteredPassword>'
```

Username: *admin*

Password: ‘ OR 1=1 –

- *admin* for login
- ‘’’ to close off the first quotation in the SQL statement
- ‘OR 1=1’ is a tautology returning true
- ‘–‘ is a inline comment (comment out the original closing bracket).

```
SELECT *
FROM USERS
WHERE 1=1
AND Username = 'admin'
Password = '' OR 1=1 -- '
```

SQL injection attacks the Application layer (7) in the OSI model.

Part G:

You can access parts of the deep web using normal browsers like Chrome or Firefox, as it includes content not indexed by search engines, such as private databases and subscription sites. However, to access the dark web, which requires anonymity, you need the Tor browser to connect to ".onion" sites that regular browsers cannot access.

Question 4 (328 words)

Part A:

Sender:

Keys:

- A symmetric session key - encrypt email/attachments.
- Recipient's (asymmetric) public key. Encrypt session key and ensures confidentiality - only the recipient can decrypt.
- Asymmetric private key. Used as a digital signature to provide integrity/authentication. The hashed digest of the encrypted email is encrypted with this.

Certificates:

- The digital certificate to verify the receiver's public key.

Recipient

Keys:

- The sender's public key to decrypt the digital signature.
- (Own) Private key to decrypt the symmetric session key.
- The sender's symmetric key, to decrypt the email and attachments.

Certificates:

- The digital certificate to verify the sender's public key.

Part B:

Filter traffic through the WAP with a firewall and NIDS or remove the WAP and allow access to the CDE only through the internet via the DMZ. Also, consider adding a honeypot in the DMZ.

Part C:

Yes, Kerberos can technically be used in web-based solutions over the internet if a trusted KDC is accessible to both client and server. However, limitations exist:

1. Web applications sending passwords compromise Kerberos's password-free security model.
2. Managing Kerberos tickets globally adds complexity compared to standards (HTTPS/TLS).
3. Kerberos provides authentication only, while TLS also offers confidentiality.

Part D:

S/MIME operates at the Application Layer (OSI 7).

TLS functions at the Transport Layer (OSI 4).

Part E:

Companies are adopting quantum-resistant encryption now to protect against "harvest now, decrypt later" threats. This Addresses the risk that encrypted data harvested today could be decrypted by future quantum computers. By using quantum-resistant algorithms early, companies extend the longevity of data security. Still valuable information remains protected as long as possible, regardless of when quantum computing advances make current encryption vulnerable.

Part F:

The primary objective of a security audit is to assess risk management and validate security controls. Audits ensure controls are 1) properly implemented, 2) fit for purpose, and 3) effective in mitigating risks. By reviewing documentation, performing penetration tests, and testing performance, audits identify vulnerabilities and provide insights into security posture. Post-audit reports offer actionable recommendations, security performance metrics, timelines, and risk levels to address gaps.

Question 5 (321 words)

Part A:

Number 1-4	**Area Name**
2	DMZ
1	General Staff
4	Cardholder Environment
3	Sales Hot desks
Letter (A-M	**Device Name**
A	File Server
I	Stateful inspection Firewall
H	Wireless Access Point
E	Email Server
F	Web Proxy
K	NIDS
L	Database Server
J	VPN Server
G	Border Router
B	Print Server
C	Dynamic Filtering Firewall
M	Web server
D	Dynamic Filtering Firewall (2)

Part B:

Creating a vanity onion address and finding a golden nonce in blockchain mining both involve repeatedly generating values to achieve a specific pattern in a hash output. Both processes rely on brute-force searching and are computationally intensive, repeatedly hashing random values until a match is found. Vanity onion addresses are produced by generating cryptographic key pairs and hashing them to create a Tor address with a recognizable prefix, while blockchain mining aims to find a nonce that when combined with block data produces a hash meeting a specific difficulty target (starting with a certain number of zeros). The key differences are their purposes: vanity onion addresses enhance branding on the Tor network, whereas mining secures the blockchain by validating transactions and rewarding miners.

Part C:

In Bitcoin's Proof of Work (PoW) mining process, the "Golden Nonce" is a specific value that miners search for to create a new block and earn a reward.

The golden nonce is a 32-bit unsigned integer ‘used once’ and is included in the block header. Miners combine it with other block data and hash it to produce a block hash, aiming to find a nonce that results in a hash less than or equal to the current target difficulty.

The target difficulty measures how hard it is to find a new block. It changes periodically in Bitcoin, to maintain a consistent block mining rate of about every 10 minutes. When a miner finds a Golden Nonce that meets this requirement, they successfully create a valid block for the network.

Question 6 (257 words)

a) Vulnerability: The vulnerability chosen is CVE-2020-13956, which affects Apache HttpClient versions prior to 4.5.13 and 5.0.3. This vulnerability allows malformed URLs to bypass the expected target host, redirecting requests to an unintended host. If exploited, this could allow a bad actor to receive traffic intended for a legitimate user.

b) To exploit this vulnerability, a bad actor could familiarise themselves with the source code to understand how to fabricate a malformed URL that would send traffic intended for a legitimate target host (for example `https://legitimate.com`), the traffic would instead be sent to the bad actor's server. An attacker can perform a man in the middle attack where the attacker presents a seemingly legitimate website to the users under a different domain and exploits the vulnerability to forward requests to the server under the guise of a legitimate domain. The initial attack vector here would be domain spoofing. This tactic is often used in phishing attacks or malware delivery attempts. Unsuspecting users are tricked into clicking malicious links or sharing credentials, believing they are interacting with a trusted source.

c) Customers which login into applications backed by the Apache HttpClient which have fallen victim of the man in the middle attacks would have security information (such as passwords) and PII compromised.

d) Countermeasure: A simple but effective countermeasure is to update Apache HttpClient to version 4.5.13 or 5.0.3 (or later), which addresses this vulnerability by validating and correctly interpreting URI components. A bastion host may also be used to configure rules to drop requests with malformed URIs.

Appendix 1

Script used to find if there were any websites under a different domain.

```python
import requests


base_url = "https://alexpudmenzky.com/BISM3205/"

letters_left = ['F', 'G', 'J', 'Q', 'R', 'V', 'W', 'Z']


for letter1 in letters_left:
    for letter2 in letters_left:
        url = base_url + letter1 + letter2 + 'code.zip'
        try:
            response = requests.get(url)
            # Check if the response status code is 200 (OK)
            if response.status_code == 200:
                print(f"Valid route found: {url}")
            else:
                print(f"Route not found: {url} (Status code:
{response.status_code})")
        except requests.exceptions.RequestException as e:
            print(f"Error accessing {url}: {e}")
```

```
(base) PS C:\git\DanielCiccC.github.io\BISM3205\Assignments\A2> python routes.py
Route not found: https://alexpudmenzky.com/BISM3205/FFcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/FGcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/FJcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/FQcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/FRcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/FVcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/FWcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/FZcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/GFcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/GGcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/GJcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/GQcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/GRcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/GVcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/GWcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/GZcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/JFcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/JGcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/JJcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/JQcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/JRcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/JVcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/JWcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/JZcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/QFcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/QGcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/QJcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/QQcode.zip (Status code: 404)
Valid route found: https://alexpudmenzky.com/BISM3205/QRcode.zip
Route not found: https://alexpudmenzky.com/BISM3205/QVcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/QWcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/QZcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/RFcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/RGcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/RJcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/RQcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/RRcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/RVcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/RWcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/RZcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/VFcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/VGcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/VJcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/VQcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/VRcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/VVcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/VWcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/VZcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/WFcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/WGcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/WJcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/WQcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/WRcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/WVcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/WWcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/WZcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/ZFcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/ZGcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/ZJcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/ZQcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/ZRcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/ZVcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/ZWcode.zip (Status code: 404)
Route not found: https://alexpudmenzky.com/BISM3205/ZZcode.zip (Status code: 404)
```

Appendix 2

```python
import hashlib

def calculate_md5(file_path):
    """Calculate the MD5 hash of a file."""
    md5_hash = hashlib.md5()
    with open(file_path, 'rb') as file:
        for byte_block in iter(lambda: file.read(4096), b""):
            md5_hash.update(byte_block)
    return md5_hash.hexdigest()


md5_result = calculate_md5('plane.jpg')
print(f"MD5 hash of the image plane: {md5_result}")
md5_result = calculate_md5('ship.jpg')
print(f"MD5 hash of the image ship: {md5_result}")
md5_result = calculate_md5('number.gif')
print(f"MD5 hash of the number gif: {md5_result}")
```

```
(base) PS C:\git\DanielCiccC.github.io\BISM3205\Assignments\A2> python
.\md_5_hash.py
MD5 hash of the image plane: 253dd04e87492e4fc3471de5e776bc3d
MD5 hash of the image ship: 253dd04e87492e4fc3471de5e776bc3d
MD5 hash of the number gif: f5ca4f935d44b85c431a8bf788c0eaca
```

Appendix 3

References

Microsoft. (2009, 08 10). *Authentication Errors are Caused by Unsynchronized Clocks*. Retrieved from Microsoft Learn: https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc780011(v=ws.10)?redirectedfrom=MSDN