Pharming

1. Customer
2. DNS Server
3. 4
4. www.mybank.com

Phishing

Member Log In

Registered users log in here. Be sure to protect your password.

Email Address: [field]
Password: [field]

New users sign up here! It only takes a minute.
Phishing

From: American Express Online 10659

Subject: RE: Fraud Detection: Please view your recent activity and we’ll help you take corrective action. ID0225840

Date: October 20, 2018 at 4:50:35 PM EDT

To: undisclosed-recipients;
TCP Three-Way Handshake

![TCP Three-Way Handshake Diagram]

**TCP Flow & Congestion Control**
- 
  - Sender must determine maximum amount of data in transit that will not overrun either receiver or network
  - Solutions for flow control:
    - Maintain a sliding window to track data in transit
    - Size of window determined by minimum of flow window and congestion window
    - Receiver ACKs slide left side of window forward (right)
    - Opens up another slot at right side of window for transmission

**TCP Header Format**
- 
  - Without options, TCP header 20 bytes
  - IP header is also 20 bytes
  - Thus, typical Internet packet min of 40 bytes (+link header)

**TCP Connection Establishment**
- 
  - Exchange necessary information to begin communication
  - Three-way handshake
    - E.g., server listening on socket
    - Client
      - SYN, sequence # = x
    - Server
      - SYN+ACK, sequence # = y
        - Acknowledgment = x + 1
      - ACK, Acknowledgement = y + 1

**Distributed Denial of Service (DDoS)**
- 
  - Attacker
  - Controller
    - Zombies
    - Victim
  - Without specific details in the image, the concept of Distributed Denial of Service (DDoS) is illustrated through a network diagram showing how an attacker controls a large number of infected systems (zombies) to launch a cyber attack on a targeted server (victim).
Botnet Resource Consumption

<table>
<thead>
<tr>
<th>Attack</th>
<th>Resource Threshold</th>
<th>Requests/Bot</th>
<th>Bots needed to exhaust</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static HTTP GET</td>
<td>60,000/sec</td>
<td>93 requests/sec</td>
<td>645</td>
</tr>
<tr>
<td>Dynamic HTTP GET</td>
<td>3,000/sec</td>
<td>93 requests/sec</td>
<td>40</td>
</tr>
<tr>
<td>SSL Handshake</td>
<td>600/sec</td>
<td>10 requests/sec</td>
<td>60</td>
</tr>
</tbody>
</table>

Heartbleed (2014)

Heartbeat – Normal usage
- Server, send me this 4 letter word if you are there: "bird"
- Server

Heartbeat – Malicious usage
- Server, send me this 500 letter word if you are there: "bird"
- Server

Server
- User Bob has connected
- User Alice wants 4 letters: bird
- Server master key is 3143198531054
- User Carol wants to change password to "password 123"

Server
- User Bob has connected
- User Mallory wants 500 letters: bird
- Server master key is 3143198531054
- User Carol wants to change password "password 123"
What do we need for a secure communication channel?

- Authentication (Who am I talking to?)
- Confidentiality (Is my data hidden?)
- Integrity (Has my data been modified?)
- Availability (Can I reach the destination?)

Example: Eavesdropping - Message Interception (Attack on Confidentiality)

Unauthorized access to information
Packet sniffers and wiretappers
Illicit copying of files and programs

Eavesdropping Attack: Example

- tcpdump with promiscuous network interface
- On a switched network, what can you see?
- What might the following traffic types reveal about communications?
  - Full IP packets with unencrypted data
  - Full IP packets with encrypted payloads
  - Just DNS lookups (and replies)
Fabrication

Masquerader: from A

Tampering

Perpetrator
**Encryption**

Well...

What tools do we have at hand?

- **Hashing**
  - e.g., SHA-1

- **Secret-key cryptography**, aka symmetric key.
  - e.g., AES

- **Public-key cryptography**
  - e.g., RSA

**Secret Key Cryptography**

- Given a key $k$ and a message $m$
  - Two functions: Encryption ($E$), decryption ($D$)
  - Ciphertext $c = E(k, m)$
  - Plaintext $m = D(k, c)$
  - Both use the same key $k$.

**Example: Stream Ciphers**

Alice: $K_{A-B}$ → PRNG → Pseudo-Random stream of $L$ bits

XOR

Message of Length $L$ bits

= Encrypted Ciphertext

Bob uses $K_{A-B}$ as PRNG seed, and XORs encrypted text to get the message back (just like OTP).
**Public Key Cryptography**

Alice and Bob need a shared symmetric key to communicate. The Key Distribution Center (KDC) is used to distribute this key.

1. KDC generates a session key, $R_1$, and shares it with each registered user (e.g., $K_{B-KDC}$ and $K_{A-KDC}$).
2. Alice and Bob know their own symmetric keys, $K_{A-KDC}$ and $K_{B-KDC}$, for communicating with the KDC.
3. Alice and Bob use $R_1$ as a session key for shared symmetric encryption.

**Certification Authorities**

A certification authority (CA) binds a public key to a real-world identity. When Alice wants Bob's public key:

- She gets Bob's certificate (Bob or elsewhere).
- She uses the CA's public key to verify the signature within Bob's certificate, then accepts the public key.

Verification equation: $\text{Verify}(S, K_{B-CA})$. If the signature is valid, Alice uses $K_{B-CA}$ to extract the certificate and establish a secure communication channel.
Certificates

The site's security certificate is not trusted!

You attempted to reach 172.20.0.1, but the server presented a certificate issued by an entity that is not trusted by your computer's operating system. This may mean that the server has generated its own security credentials, which Chrome cannot rely on for identity information, or an attacker may be trying to intercept your communications.

You should not proceed, especially if you have never seen this warning before for this site.

[Links: Proceed anyway, Back to safety, Help me understand]