**UNIT-IV**

**Web-based Password Cracking Techniques**

**Introduction:**

**Definition of Password Cracking:**

Password Cracking is the process of obtaining unknown passwords. It refers to various measures used to discover computer passwords. This is usually accomplished by recovering passwords from data stored in, or transported from, a computer system.

**Why password Cracking?**

Password cracking can be done for several reasons.

1. The most **malicious** reason is in order to gain unauthorized access to a computer without the computer owner’s awareness. This results in cybercrime such as stealing passwords for the purpose of accessing banking information.
2. The **nonmalicious** reasons for password cracking occur when someone has misplaced or forgotten a password.

**Eg:** If a system administrator is conducting tests on password strength as a form of security so that hackers cannot easily access protected systems.

Passwords can be served as a type of **authentication** method used to allow secure access to confidential materials on the Internet.

**Definition of Authentication:**

Authentication is the process of determining whether someone or something is, in fact, who or what the individual or entity claims to be.

**Importance of Authentication:**

It plays a critical role in the security of an application, since all subsequent security decisions are typically made based on the identity established by the supplied credentials.

Generally, in a computer network, authentication of users is performed with a user account that consists of a **user ID and password**. Every user that logs in with a user ID and password is considered to be a valid user if the combination is present in the database. Prior to obtaining an account, the user is required to **register** for the account. Authentication via usernames and passwords is most common today, but other Web-based authentication methods exist that provide stronger security.

Many different authentication methods can be used, depending upon the requirements of the Web site; however, basic security design principles will prevent most attacks.

**Authentication Techniques**:

The authentication process includes the following techniques:

1. HTTP authentication:
2. Basic

b) Digest

2. Integrated Windows (NTLM) authentication

3. Negotiate authentication

4. Certificate-based authentication

5. Forms-based authentication

6. RSA secure tokens

7. Biometrics

1. **HTTP Authentication**
2. **Basic:** Basic authentication is the most basic form of authentication available to Web applications.

* This mechanism requires a simple sign-in with a user ID and password for each realm (Figure 4-1).
* The realm is case-sensitive and its value is a string. The string is a combination of characters and other semantics specific to the authentication scheme being implemented.
* This realm value is considered an opaque (non-transparent) string when compared with the realms in the server for a match.
* Basic authentication begins with a client making a request to a Web server for a protected resource without any authentication credentials. A client’s request to access a secure space on the Web or a URL is validated based on the user ID and password.
* The client sends the user ID and password separated by a colon (“:”), using a Base64-encoded string.
* Most Web browsers deal with such requests automatically by asking the user for a username and password.
* Every network resource that is secured using a basic authentication mechanism requires a response to its “401 authentication required” header from the client.
* If the client responds to the header with the correct credentials, access is granted to the resource (if the client supports the basic authentication mechanism). Since HTTP is a stateless protocol, each time the client requests a resource, credentials must be supplied.
* The client browser can store authentication details and supply a request for authentication each time it is required from the server.
* In the basic authentication mechanism, the login details are stored on the client’s browser corresponding to the server’s name.

**Drawbacks of Basic Authentication:**

* Basic authentication is wide open to eavesdropping attacks, despite the Base64-encoded form of the value it sends in the authentication header.
* This is the most severe limitation of the protocol. Most browsers, including Internet Explorer and Netscape, will cache basic authentication credentials and send them automatically to all pages in the realm, whether or not they use SSL. Due to its simple nature, it is easily passed through proxy servers.

**Solutions:**

* The use of 128-bit SSL encryption can thwart eavesdropping attacks.



**b.) Digest:**

Digest authentication was designed to provide a higher level of security than basic authentication. It is based on the challenge-response authentication model. The main difference is that it doesn't require sending the username and password across the wire in plaintext. It is also immune to replay-attacks, as it uses a one-time number from the server.

Digest authentication offers the same features as Basic authentication, but provides a more secure way of transmitting the authentication credentials.

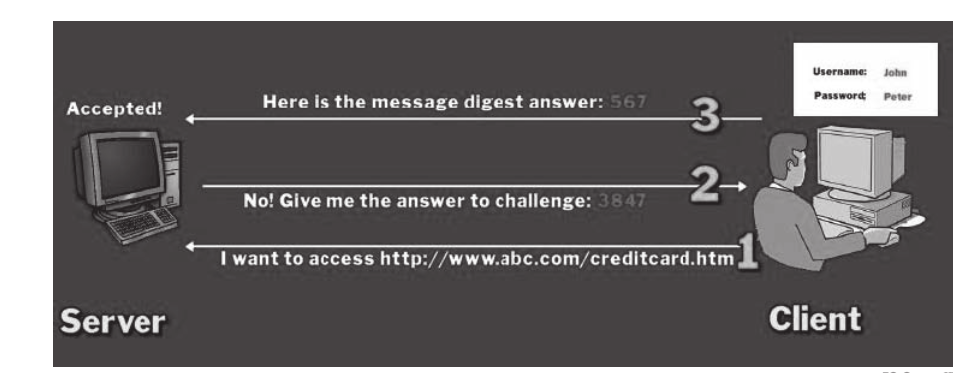
Digest authentication relies on the HTTP 1.1 protocol .This is not supported by all browsers. If a non-HTTP 1.1 compliant browser requests a file when Digest authentication is enabled, the request is rejected. Digest authentication can be used only in Windows domains.

The server gives the client a one-time use number (a nonce) that it combines with the username, realm, password and the URI request. The client runs all of those fields through an MD5 hashing method to produce a hash key. It sends this hash key to the server along with the username and the realm to attempt to authenticate.

Server-side the same method is used to generate a hash key, only instead of using the password typed in to the browser the server looks up the expected password for the user from its user DB. It looks up the stored password for this username, runs in through the same algorithm and compares it to what the client sent. If they match then access is granted, otherwise it can send back a 401 Unauthorized (no login or failed login) or a 403 Forbidden (access denied).

The following steps explains the challenge-response model.

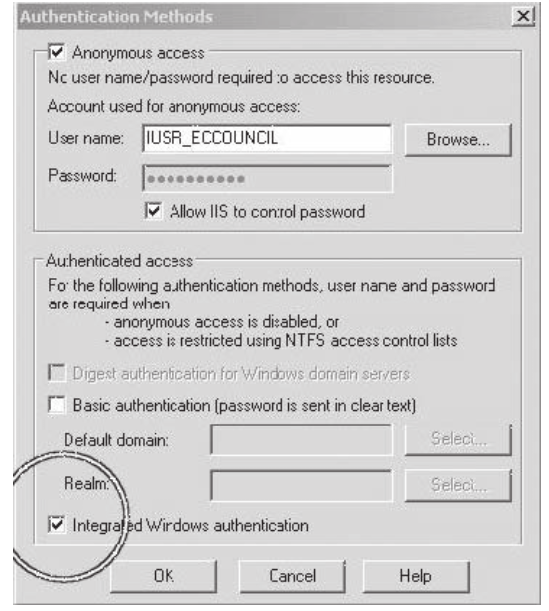
1. Client makes request
2. Client gets back a nonce from the server and a 401 authentication request
3. Client sends back the following response array (username, realm, generate\_md5\_key(nonce, username, realm, URI, password\_given\_by\_user\_to\_browser)) (yea, that's very simplified)
4. The server takes username and realm (plus it knows the URI the client is requesting) and it looks up the password for that username. Then it goes and does its own version of generate\_md5\_key(nonce,username, realm, URI, password\_I\_have\_for\_this\_user\_in\_my\_db)
5. It compares the output of generate\_md5() that it got with the one the client sent, if they match the client sent the correct password. If they don't match the password sent was wrong.



1. **NTLM**

NTLM is a Microsoft-proprietary protocol that authenticates users and computers based on an authentication challenge and response. Integrated Windows authentication, shown in Figure 4-3, n HTTP. It only works between **Microsoft’s Internet Explorer and IIS Web servers**. It works in the same way as digest authentication, using a challenge-response mechanism.

When a client requests a resource protected by Integrated Windows authentication, the server responds with an HTTP401 Access Denied response and a WWW-Authenticate: NTLM [challenge] header. The [challenge] value contains a digest of the NTLM nonce and other information related to the request. Internet Explorer gathers the NTLM credentials for the currently logged-on Windows user, uses the NTLM algorithm to hash the challenge value, and then provides the hashed value in an HTTP response with an Authorization: NTLM [response] header. If these credentials fail three times in a row, Internet Explorer asks the user for a username, password, and domain name in a separate window. The user may now enter details such as the correct username, password, and domain, and the process repeats itself.



1. **Negotiate Authentication:**

Negotiate authentication is an extension of NTLM authentication. It provides **Kerberos- based authentication over HTTP** and is considered secure. It was introduced in Windows 2000 and, as its name suggests, uses a negotiation process to decide on the level of security used. If the hosts are in the same domain, Negotiate will use Kerberos-based authentication, and if they are not in the same domain, then it will use NTLM based authentication. This authentication technique can provide a strong level of security if the hosts are in the same domain and are all running Windows 2000 or a later version. However, this configuration is fairly restrictive and uncommon, except on corporate intranets.

1. **Certificate-based authentication:**

Certificate-based authentication is the use of a [Digital Certificate](https://www.globalsign.com/en/ssl-information-center/what-is-public-key-cryptography/) to identify a user, machine, or device before granting access to a resource, network, application, etc.

It uses **public-key cryptography** and a **digital certificate** to authenticate users. Public-key cryptography is the most common method on the Internet for authenticating a message sender or encrypting a message. A digital certificate is an electronic “credit card” that establishes user credentials while doing business or other transactions on the Web, and a certification authority (CA) issues it.

**What is CA?**

A CA is a trusted entity that signs certificates and can vouch for the identity of the user and the user’s public key. The digital certificate is composed of the CA’s name, serial number, key expiry date, copy of the public key, and digital signature of the certificate-issuing authority so that a recipient can verify that the certificate is authentic.\

**Steps involved in Certificate-based authentication:**

1. When the user connects to a server to authenticate, the user presents a digital certificate containing the public key and signature of the CA.
2. The server first verifies that the signature on the certificate is valid and was generated by a trusted CA. The server then authenticates the user by using public-key cryptography to prove that the user truly holds the private key associated with the certificate.
3. When the recipient obtains the encrypted message, the public key that the CA issues is used to decode the digital certificate. The recipient verifies that the digital certificate has been issued by the CA and then decodes the message using the sender’s public key and identification information held within the certificate. The recipient can use the details obtained from the digital certificate to send further encrypted messages.

**Digital certificates are often used to authenticate users in the following ways:**

**• E-mail:** Digital certificates that are used to digitally sign e-mail messages enhance confidentiality and security through built-in encryption mechanisms.

**• Network security:** Smart cards and other technologies that use digital certificates are deployed by enterprises as a security mechanism to protect their corporate network.

1. **Form-based authentication:**

Itallows the developer to control the look and feel of the login authentication screens by customizing the login screen and error pages that an HTTP browser presents to the end user. When form-based authentication is declared, the following actions occur.

1. A client requests access to a protected resource.
2. If the client is unauthenticated, the server redirects the client to a login page.
3. The client submits the login form to the server.
4. The server attempts to authenticate the user.

a) If authentication succeeds, the authenticated user’s principal is checked to ensure that it is in a role that is authorized to access the resource. If the user is authorized, the server redirects the client to the resource by using the stored URL path.

1. If authentication fails, the client is forwarded or redirected to an error page.

Figure shows what happens when you specify form-based authentication.

##### Form-Based Authentication

Diagram of four steps in form-based authentication between
client and server

1. **RSA Secure tokens:**

The RSA SecureID token is an authentication mechanism assigned to a specific user. The token generates an authentication code every 60 seconds using a built-in clock and the RSA SecureID card’s factory-encoded random key. Each token has a different code, and is loaded into the corresponding SecureID server as the tokens are purchased. The SecureID server uses an algorithm to anticipate the current code displayed on any token. When users attempt to access a network resource, they must enter a PIN and the currently displayed code to access the resource. The server checks these numbers and, if they match, provides authentication.

1. **Biometric Authentication:**

Biometric authentication is a technique that uses physical characteristics to verify a user’s identity. This is a trusted authentication mechanism because biometric characteristics are particular to an individual. Another advantage of biometric authentication, as opposed to a more traditional method, is that biometric measurements cannot be lost or forgotten.

The following biometric technologies are frequently employed as authentication mechanisms:

• Fingerprinting

• Hand-geometry scanning

• Retina scanning

• Facial recognition systems

**How to perform Password Cracking?**

Password crackers use two primary methods to identify correct passwords:

1. Brute-force

2. Dictionary searches.

**Brute-force:** When a password cracker uses brute-force, it runs through combinations of characters within a predetermined length until it finds the combination accepted by the computer system.

**Dictionary Search:** When conducting a dictionary search, a password cracker searches each word in the dictionary for the correct password. Password dictionaries exist for a variety of topics and combinations of topics, including politics, movies, and music groups.