CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

INSTITUTE : UIE

DEPARTMENT : CSE

Bachelor of Engineering (Computer Science & Engineering)

WEB AND MOBILE SECURITY (Professional Elective-I)

(20CST/IT-333)

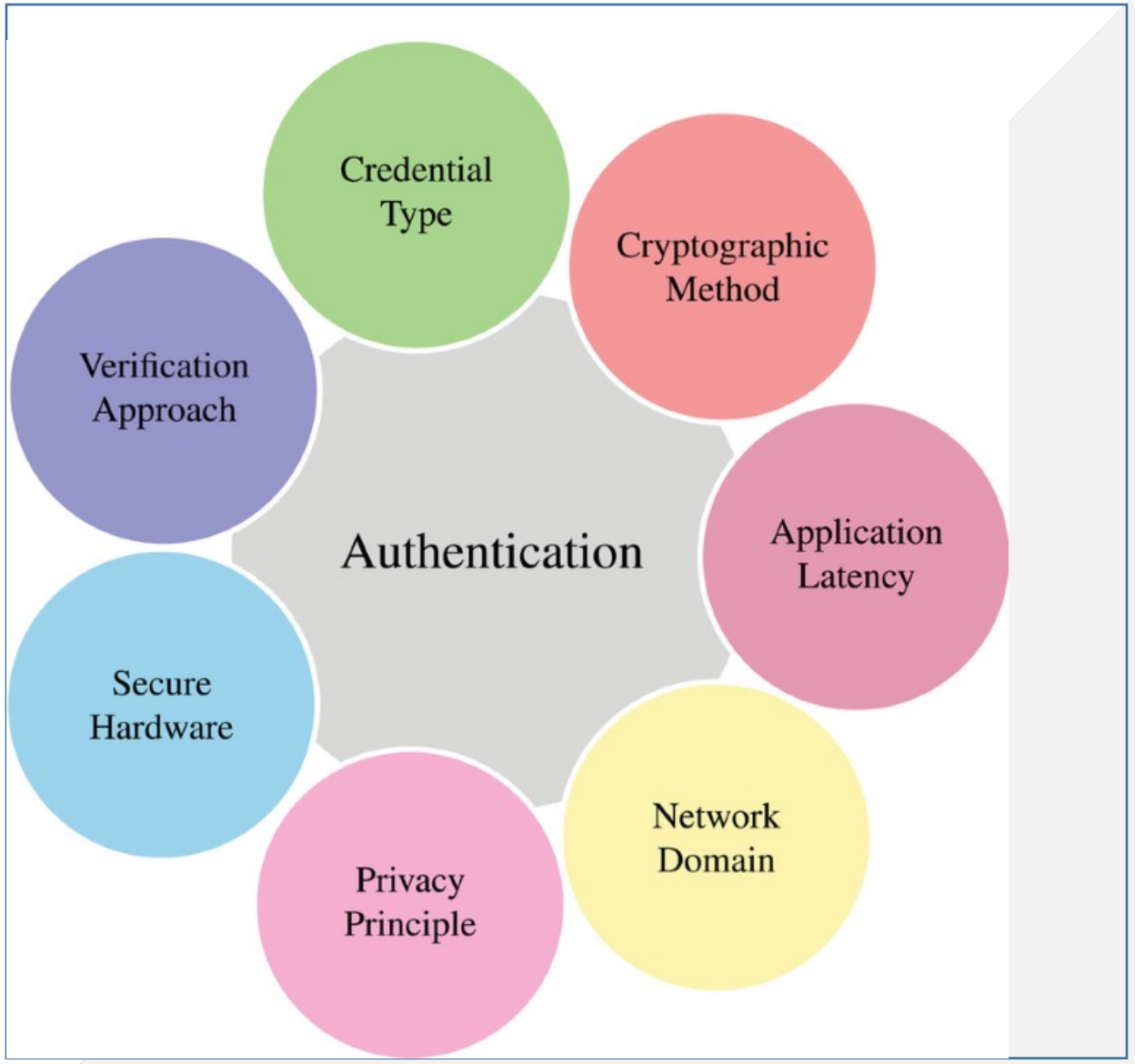
TOPIC OF PRESENTATION:

Attacking web authentication

Lecture Objectives

In this lecture, we will discuss:

attacks and countermeasures for common web authentication mechanisms, including password-based, multifactor (e.g., CAPTCHA), and online authentication services like Windows Live ID.



Web authentication threats
1. Username/Password based Threats

- Although there are numerous ways to implement basic username/password authentication, web implementations generally fall prey to the same types of attacks:
- Username enumeration
- Password guessing
- Eavesdropping

a) Username Enumeration

Username enumeration is primarily used to provide greater efficiency to a password guessing attack. This approach avoids wasting time on failed attempts using passwords for a user who doesn't exist. For example, if you can determine there is no user named Alice, there's no point in wasting time trying to guess Alice's password. The following are some examples of functionality often used in web applications that may allow you to determine the username.

- **Profiling Results**

Smart attackers always review their profiling data because it's often a rich source of such information (textual searches across the profiled information for strings like userid, username, user, usr, name, id, and uid often turn it up).

- **Error Messages in Login**

A simple technique to determine if a username exists is to try to authenticate to a web application using invalid credentials and then examine the resulting error message. For example, try authenticating to the target web application using the username **Alice** and the password **abc123**. You are likely to encounter one of three error messages similar to the ones listed here, unless you actually successfully guessed the password:

- You have entered a bad username.
- You have entered a bad password.
- You have entered a bad username/password combination.

b) Password Guessing

Not surprisingly, password guessing is the bane of username/password authentication schemes. Unfortunately, such schemes are common on the Web today and thus fall prey to this most basic attack techniques. Password-guessing attacks can usually be executed regardless of the actual authentication protocol in place. Manual guessing is always possible, of course, and automated client software exists to perform password guessing against the most commonly used protocols.

Username Guesses	Password Guesses
[NULL]	[NULL]
root, administrator, admin	[NULL], root, administrator, admin, password, [company_name]
operator, webmaster, backup	[NULL], operator, webmaster, backup
guest, demo, test, trial	[NULL], guest, demo, test, trial
member, private	[NULL], member, private
[company_name]	[NULL], [company_name], password
[known_username]	[NULL], [known_username]

Table 4-1 Common Usernames and Passwords Used in Guessing Attacks (Not Case-sensitive)

c) Eavesdropping and Replay Attacks

- Any authentication protocol that exposes credentials while in transit over the network is potentially vulnerable to eavesdropping attacks, which are also called *sniffing attacks* after the colloquial term for network protocol analyzers. A replay attack usually is built upon eavesdropping and involves the use of captured credentials by an attacker to spoof the identity of a valid user.
- Countermeasures: The use of 128-bit SSL encryption can thwart these attacks and is strongly recommended for all web sites that use Basic and Digest authentication. To protect against replay attacks, the Digest nonce could be built from information that is difficult to spoof, such as a digest of the client IP address and a timestamp.

2. Forms-based Authentication Attacks

- Forms-based authentication does not rely on features supported by the basic web protocols like HTTP (such as Basic or Digest authentication). It is a highly customizable authentication mechanism that uses a form, usually composed of HTML with FORM and INPUT tags delineating input fields, for users to enter their username and password.
- After the user credentials are sent via HTTP or HTTPS, they are then evaluated by some server-side logic and, if valid, some sort of unique token of sufficient length, complexity, and randomness is returned to the client for use in subsequent requests.

3. User Registration Attacks

- Sometimes, the easiest way to access a web application is to simply create a valid account using the registration system. This method essentially bypasses attacks against the authentication interface by focusing on the registration process. Of course, filtering account registrations for malicious intent is a challenging proposition, but web applications have developed a number of mechanisms to mitigate against such activity, including *Completely Automated Public Turing Tests to Tell Computers and Humans Apart (CAPTCHA).* CAPTCHAs are often used in web-based applications when the application owner wants to prevent a program, bot, or script from performing a certain action.
- Some examples of CAPTCHA include these:

• **Free e-mail services** Many free e-mail services use CAPTCHA to prevent programs from creating fake accounts, generally to minimize spam.

- **Password-guessing attack prevention** CAPTCHA has been used in login pages to prevent tools and programs from executing automated password guessing attacks.
- **Search engine bot prevention** CAPTCHAs are sometimes used to prevent search engine bots from indexing pages.
- **Online polls** CAPTCHA can be an effective way to prevent people from skewing results of online polls by ensuring that a program is not responding to the polls.
- CAPTCHA is a type of Human Interactive Proof (HIP) technology that is used to determine if the entity on the other side is a human or a computer. This is formally referred to as a *Reverse Turing Test (RTT).* The difference with CAPTCHA is that it is “completely automated,” which makes it suitable for use in web applications.

4. Credential Management Attacks

- Another way to bypass authentication is to attack credential management subsystems.
- For example, most web sites implement common mechanisms for password recovery, such as self-help applications that e-mail new passwords to a fixed e-mail address, or if a “secret question” can be answered (for example, “What is your favorite pet’s name?” or “What high school did you attend?”).

References:

Books:

1. Hacking Exposed Mobile: Security Secrets & Solutions 1st Edition, Kindle Edition, by Neil Bergman, Mike Stanfield, Jason Rouse, and Joel Scambray
2. Hacking Exposed Web Applications, 3rd edition, Joel Scambray, Vincent Liu, Caleb Sima, Released October 2010, Publisher(s): McGraw-Hill

Reference Links:

http://www.ijcse.com/docs/INDJCSE14-05-02-061.pdf
https://www.it.iitb.ac.in/~madhumita/web%20services/Web%20Security%20Threats%20and%20Countermeasu_files/rightframe.htm

Relevant Videos:

https://www.techtarget.com/searchsecurity/tip/Use-these-6-user-authentication-types-to-secure-networks
https://www.techtarget.com/searchsoftwarequality/definition/application-security

THANK YOU