UNIVERSITY of INFORMATION TECHNOLOGY

Software-Based Attacks

Software-Based Attacks

- Software plays an important role in building reliable security as crypto, access control, and protocols
- Several security issues related to software:

1. 5.1 System Vulnerabilities Attacks
2. 5.2 Malicious Software
3. 5.3 Other Software-based Attacks

Objectives

1. To learn several security issues related to software.
2. To distinguish and classify particular examples of attacks.
3. To understand and analyze software vulnerabilities and security solutions to reduce the risk of exploitation.

System Vulnerabilities Attacks

- In order to protect from the potential threats, the typical vulnerabilities should be learned:

1. Buffer Overflow
2. SQL Injection
3. Cross-Site Scripting (XSS)
4. Cross-Site Request Forgery (CSRF)
5. Session Hijacking

System Vulnerabilities Attacks

1 Buffer Overflow Attack

- Users enter data into a Web form
- Web form is sent to server
- Server writes data to array called buffer, without checking length of input data
- Data “overflows” buffer
    - Such overflow might enable an attack
    - Attack could be carried out by anyone with Internet access

System Vulnerabilities Attacks

1 Buffer Overflow Attack

- What happens when code is executed?

```
int main(){
   int buffer[10];
   buffer[20] = 37;}
```

- Depending on what resides in memory at location “buffer[20]”
  - Might overwrite **user** data or code
  - Might overwrite **system** data or code
  - Or program could work just fine

System Vulnerabilities Attacks

1 Buffer Overflow Attack

Countermeasure

- Rewriting the program’s code by developer
  - check the valid size of input from outside.
  - validation mechanism to every input portion of the program.

System Vulnerabilities Attacks

2 SQL Injection Attack

- Exploits vulnerabilities in input validation to run arbitrary commands in the database.
- Occur when an application (typically a Web application) uses input to construct dynamic SQL statements to access the database.
- Using the SQL injection attack, the attacker can execute arbitrary commands in the database.

System Vulnerabilities Attacks

2 SQL Injection Attack

Example of SQL injection

- What if a user entered the search word as:

‘; DELETE FROM my_table; ‘?

- The **$sql** variable will have the content like this,

SELECT * FROM my_table WHERE name LIKE '%'; DELETE FROM my_table;%‘

- ***It will execute 1 select query and 1 delete query, which deletes all data from the table.***

System Vulnerabilities Attacks

1 SQL Injection Attack

Countermeasure

- The only countermeasure is
  - rewrite the application’s code by developer to neutralize the input from the user so that the input will not be translated to raw SQL commands.

System Vulnerabilities Attacks

3 Cross-Site Scripting (XSS) Attack

- Inject client-side script (typically JavaScript) code into a dynamic Web site so that normal users visiting that Web site will be forced to execute that malicious script.
- The attack targets your application’s users and not the application itself, but it uses your application as the vehicle for the attack.
- Because the script code is downloaded by the browser from a trusted site, the browser has no way of knowing that the code is not legitimate.

System Vulnerabilities Attacks

3 Cross-Site Scripting (XSS) Attack

- The attacker can perform various kinds of malicious activities by using XSS.
  1. Steal a user’s authentication cookies so that the attacker can do session hijacking
  2. Redirect the page to the attacker’s malicious page.
  3. Completely or partly rewrite the genuine Web page into the attacker’s malicious page such as Phishing

UNIVERSITY of INFORMATION TECHNOLOGY

System Vulnerabilities Attacks

3 Cross-Site Scripting (XSS) Attack

Countermeasure

- Rewriting the Web application’s code by developer to sanitize input from the user so that the input will not be translated into raw JavaScript commands.

System Vulnerabilities Attacks

4 Cross-Site Request Forgery (CSRF)

- CSRF attack tricks the authenticated user into unintentionally sending a malicious request to a Web site.
- Similar to XSS but is actually completely different kind of attack.
- CSRF is sometimes also called as XSRF, Session Riding, and One-click attack.

System Vulnerabilities Attacks

4 Cross-Site Request Forgery (CSRF)

- Attacker normally embed a form submission code or HTTP request code of target Web application into a malicious Web page or e-mail.
- When a user visit those malicious pages, the embedded request is automatically executed.
- And if that user has already logged into that target Web application, then the request will be (unintentionally) accepted.
- Transmits unauthorized commands from a user who has logged in to a website to the malicious website.

System Vulnerabilities Attacks

4 Cross-Site Request Forgery (CSRF)

Countermeasure

- From the developer’s point of view, CSRF is rather more difficult to protect than XSS or SQL injection (which are technically easy).
- In order to protect CSRF, the developer must implement a mechanism to distinguish the true, genuine submission of request from the user with false, unintentional request from the genuine user.

System Vulnerabilities Attacks

5 Session Hijacking

- Session hijacking is an attack to steal the user’s session so that the attacker can utilize the target application as if he/she is a genuine user.
- The important difference of session hijacking from password cracking is that, in this case, attacker does not need to obtain the user’s password, but just use the user’s login session.
- Stealing the user’s session is possible by various methods, including XSS and sniffing because the user’s session is typically stored in the Web browser’s cookie.

System Vulnerabilities Attacks

5 Session Hijacking

Countermeasure

- Encrypted session handling is the best solution (by using HTTPS, for example).
- In case the session cannot be encrypted, then some combination of one-time submission mechanisms, source IP checking, referrer checking would help.

Quizzes

1. In cross-site scripting where does the malicious script execute?
   A. On the web server
   B. In the user’s browser
   C. On the attacker’s system
   D. In the web app model code

2. In a _______________ attack, the extra data that holds some specific instructions in the memory for actions is projected by a cyber-criminal or penetration tester to crack the system.
   A. Phishing
   B. SQL Injection
   C. Buffer-overflow
   D. Clickjacking

Quizzes

3. Which of the following terms best describes the weakness in a system that may possibly be exploited?
   A. Threat
   B. Vulnerability
   C. Weakest link
   D. Risk