RGPVNOTES.IN

Program : **B.E**

Subject Name: **Information Security**

Subject Code: **IT-8001**

Semester: **8th**

Software Vulnerabilities

In simple terms, a vulnerability can be an error in the way that user management occurs in the system, an error in the code or a flaw in how it responds to specific requests.

Outcomes of it

- Allows an attacker to conduct information gathering activities
- Allows an attacker to hide activities
- Includes a capability that behaves as expected, but can be easily compromised
- Is a primary point of entry that an attacker may attempt to use to gain access to the system or data is considered a problem according to some reasonable security policy

Phishing attack

Phishing is a type of social engineering attack often used to steal user data, including login credentials and credit card numbers. It occurs when an attacker, masquerading as a trusted entity, dupes a victim into opening an email, instant message, or text message. The recipient is then tricked into clicking a malicious link, which can lead to the installation of malware, the freezing of the system as part of a ransomware attack or the revealing of sensitive information.

Phishing attack examples

The following illustrates a common phishing scam attempt:

- A spoofed email ostensibly from myuniversity.edu is mass-distributed to as many faculty members as possible.
- The email claims that the user's password is about to expire. Instructions are given to go to myuniversity.edu/renewal to renew their password within 24 hours.

Buffer Overflow attack

A buffer is a sequential section of memory allocated to contain anything from a character string to an array of integers.

- A buffer overflow, or buffer overrun, occurs when more data is put into a fixed-length buffer than the buffer can handle.
- The extra information, which must go somewhere, can overflow into adjacent memory space, corrupting or overwriting the data held in that space.
- This overflow usually results in a system crash, but it also creates the opportunity for an attacker to run arbitrary code or manipulate the coding errors to prompt malicious actions.
- Many programming languages are prone to buffer overflow attacks. However, the extent of such attacks varies depending on the language used to write the vulnerable program.
- For instance, code written in Perl and JavaScript is generally not susceptible to buffer overflows. However, a buffer overflow in a program written in C, C++, Fortran or Assembly could allow the attacker to compromise the targeted system fully.

Buffer Overflow Causes

- Coding errors are typically the cause of buffer overflow. Common application development mistakes that can lead to buffer overflow include failing to allocate large enough buffers and neglecting to check for overflow problems.
- These mistakes are especially problematic with C/C++, which does not have built-in protection against buffer overflows. Consequently, C/C++ applications are often targeting buffer overflow attacks.

Buffer Overflow Solutions

- To prevent buffer overflow, developers of C/C++ applications should avoid standard library functions that are not bounds-checked, such as gets, scanf, and strcpy.
- In addition, secure development practices should include regular testing to detect and fix buffer overflows.

Format string vulnerability

The Format String exploits when the submitted data of an input string is evaluated as a command by the application. In this way, the attacker could execute code, read the stack, or cause a segmentation fault in the running application, causing new behaviors that could compromise the security or the stability of the system.

To understand the attack, it's necessary to understand the components that constitute it.

- The Format Function is an ANSI C conversion function, like printf, fprintf, which converts a primitive variable of the programming language into a human-readable string representation.
- The Format String is the argument of the Format Function and is an ASCII Z string which contains text and format parameters, like printf ("The magic number is: %d\n," 1911);
- The Format String Parameter, like %x %s, defines the type of conversion of the format function.

Cross-site Scripting (XSS) Attack

Cross-site Scripting (XSS) refers to client-side code injection attack wherein an attacker can execute malicious scripts (also commonly referred to as a malicious payload) into a legitimate website or web application.

- XSS is amongst the most rampant of web application vulnerabilities and occurs when a web application makes use of invalidated or unencoded user input within the output it generates.
- By leveraging XSS, an attacker does not target a victim directly. Instead, an attacker would exploit a vulnerability within a website or web application that the victim would visit, essentially using the vulnerable website as a vehicle to deliver a malicious script to the victim's browser.

The following server-side pseudo-code is used to display the most recent comment on a web page.

print "<html>"
print "<h1>Most recent comment</h1>"
print database. latestComment
print "</html>"

- The above script is simply printing out the latest comment from a comments database and printing the contents out to an HTML page, assuming that the comment printed out only consists of text.
- The above page is vulnerable to XSS because an attacker could submit a comment that contains a malicious payload such as <script>doSomethingEvil(); </script>.

Users visiting the web page will get served the following HTML page.

<html>
<h1>Most recent comment</h1>
<script>doSomethingEvil() ;</script>
</html>

When the page loads in the victim's browser, the attacker's malicious script will execute, most often without the user realizing or being able to prevent such an attack.

SQL Injection Attack

SQL injection, also known as SQLI, is a common attack vector that uses malicious SQL code for backend database manipulation to access information that was not intended to be displayed.

- This information may include any number of items, including sensitive company data, user lists or private customer details.
- The impact SQL injection can have on a business is far-reaching. A successful attack may result in the unauthorized viewing of user lists, the deletion of entire tables and, in certain cases, the attacker gaining administrative rights to a database, all of which are highly detrimental to a business.
- When calculating the potential cost of an SQLI, it's important to consider the loss of customer trust should provide personal information such as phone numbers, addresses and credit card details being stolen.
- For example, the input http://www.estore.com/items/items.asp?itemid=999 UNION SELECT username, password FROM USERS produces the following SQL query:

SELECT ItemName, ItemDescription
FROM Items
WHERE ItemID = '999' UNION SELECT Username, Password FROM Users;

Using the UNION SELECT statement, this query combines the request for item 999's name and description with another that pulls names and passwords for every user in the database.

Email Security: -

- E-mail messages are generally sent over untrusted networks-external networks that are outside the organization's security boundary.
- When these messages lack appropriate security safeguards, they are like postcards that can be read, copied, and modified at any point along these paths.
- Securing an e-mail system is the responsibility of an organization's IT department and e-mail administrator.
- However, anyone responsible for the confidentiality, integrity, and availability of the information sent via e-mail should be aware of the threats facing e-mail systems and understand the basic techniques for securing these systems.

Security Issues

- Not Realtime can afford to use public key cryptosystems more.
- Certification of keys is much harder because anyone can send anyone else some mail
- Strictly end-to-end, IPSec/firewalls might get in the way here
- A single message can be sent to many parties
- A single message can be sent to one or more distribution lists
- There can be message forwarding loops due to distribution lists or even someone's .forward file
- Duplicate copies can be sent to the same individual
- The recipient or intermediate node may not be ready to receive mail

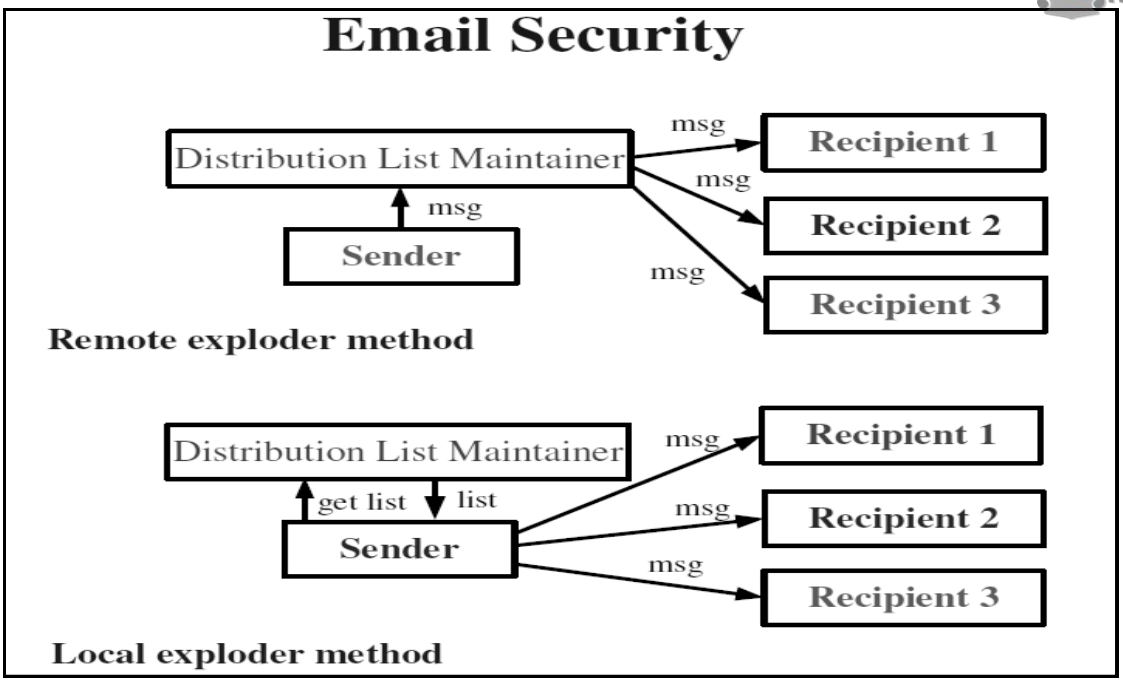

Figure 4.1 Email Security

Email Security Comparison

Local exploder method has some advantages:

- Easier to prevent mail forwarding loops caused by the sender
- The sender may be able to prevent duplicate copies to the same recipient
- The sender knows in advance what traffic will be generated (may be important if billing is based on traffic)

Remote explorer method has some advantages:

- You can send to a list of people you are not allowed to know.
- Lots of traffic may be generated away from the sender's network.
- If the distribution list is huge, it is economical to have mail.
- Distribution lists may explode to other lists – the number of recipients would be too hard for a sender to keep up with.
- Parallelism is exploited!

Security Services Over Email

- **Privacy:** No one should read message except the recipient.
- **Authentication:** Recipient should know exactly who the sender is.
- **Integrity:** Recipient should be able to tell whether a message was altered in transit.
- **Nonrepudiation:** Recipient can prove that the sender really sent it.
- **Proof of submission:** Verification to the sender that the mailer got it.
- **Proof of delivery:** Verification to the sender that the recipient got it.
- **Message flow confidentiality:** Eavesdropper cannot determine the sender's ID.
- **Anonymity:** Ability to send to the recipient does not know the sender.
- **Containment:** Ability to keep secure messages from "leaking" out of a region.
- **Audit:** Logging of events having relevance to security.
- **Accounting:** Maintain usage statistics (might charge for service).
- **Self-destruct:** Message is destroyed on delivery.
- **Message sequence integrity:** Sequence of messages have arrived in order, without loss.

Establishing Keys:

- Most services are best provided using cryptographic means
- But the email infrastructure may require many keys – where are they?

Establishing Public Keys:

- The receiver may have sent it by some other means say NY times

- The receiver may have appended it to an email message (signed)
- The receiver may have certified it through a CA
- The receiver may have posted it on a Public Key Infrastructure

Establishing Secret Keys:

- Both parties meet in private to set a key
- Communicate on the phone
- Sender gets a "ticket" from a KDC and includes it in the message

Privacy – needed because:

- An eavesdropper can easily listen especially at "No Such Agency."
- Mail can be rerouted never to reach the intended recipient
- Conflicts employee wants privacy, company wants assurance employee is not giving away company secrets.

End-to-end Privacy:

- Problem: how to encrypt lots of copies to multiple recipients?
- Secret key encryption is preferable since it is 1000 times faster
- Not desirable to use a long-term key more than needed

Hence: sender may choose a secret S only for encrypting message msg encrypted with S + S encrypted with public key > recipient S encrypted multiple times with recipients' public keys
To: Prakash, Master blaster, Cokane
From: Franco
Key-info: Prakash-7567484385785467
Key-info: Master blaster-734478868274684
Key-info: Cokane-9062346667642424
Msg-info: jkdiuqwydfkjhjdfreuigfkjsdfkjsyfuieihfuigf

Privacy with Distribution List Exploders:

Problem: Sender may not know public keys of recipients,

- The sender must have a key K shared with the distribution list exploder
- The sender encrypts a message with a secret S and sends it with
- S encrypted using K to the list exploder
- Distribution list exploder decrypts S then encrypts it with the keys of recipients (without decrypting the email?) and sends the email forward (possibly to other distribution list exploders).
- But now sender loses some assurance that the message arrives as intended.

Authentication of the Source:

Prevent C from sending mail to B with 'From A.'

Public Keys:

- Sender signs hash of message with its private key
- Works on multiple messages (same signature!)
- The public key might be sent with the message with a chain of certificates

Secret Keys:

- The sender computes a MAC: one of
- CBC residue of the message computed with a shared secret
- Hash of shared secret appended to message
- The encrypted message digest of a message

Multiple emails: use 3rd method compute
MD once, then encrypt for each addressee.

Authentication of the Source Distribution Exploders:

Public keys: Just forward the messages as is, use sender's public key to authenticate

Secret keys:

- The sender cannot be assumed to share secrets with all recipients or know who all the recipients are.
- Distribution list exploder must remove sender's authentication information from emails and replace. It with its own.
- Distribution list exploder must verify the source of the email because recipients cannot do that themselves although they can authenticate the exploder.
- Exploder may need to include the name of the sender in the body of the encrypted email.

Message Integrity: in the world of secured communications, Message Integrity describes the concept of ensuring that data has not been modified in transit. This is typically accomplished with the use of a Hashing algorithm.

- It deals with methods that ensure that the contents of a message have not been tampered with and altered.
- The most common approach is to use a one-way hash function that combines all the bytes in the message with a secret key and produces a message digest that is impossible to reverse.
- Integrity checking is one component of an information security program.

Non-Repudiation

Public Key: -

- Non-Repudiation: sender signs the message with a private key.
- Plausible Deniability: Sender computes a MAC using a random key S and sends [[S]Bob Public] Alice Private.

Secret Key:

- Non-Repudiation: Notary N. N and recipient share a secret.
- N computes a seal = digest of the message and Alice's name using a secret key.
- N shares a secret key with the recipient and sends A MAC of the message, seal, and Alice.
- A judge could ask N to verify if the seal is valid.

Computer Virus: -

"A computer virus is a program that may disturb the normal working of a computer system." Virus attaches itself to files stored on floppy disks, USBs, email attachments and hard disks. A file containing a virus is called an infected file. If this file is copied to a computer, the virus is also copied to the computer.

A computer virus cannot damage computer hardware. IT may cause many damages to a computer system. A virus can:

- A computer virus can damage data or software on the computer.
- It can delete some or all files on the computer system.
- It can destroy all the data by formatting the hard drive.
- It may display a political or false message very few times.

Types

Boot Sector Virus - As the name suggests a boot sector virus affects the boot section on your computer. Evidently, the boot sector is the section which is accessed at the very first when the computer is turned on.

- It is used to boot the information used by the operating system.
- A Boot sector virus gains complete control over the Master Boot Record (MDR) or the DOS by replacing the contents of the OS with that of its own resulting in errors during booting or 'cannot boot' message.

File Infector Virus - This is the most popular and most prevalent variant of a compiled computer virus. It attaches itself to executable programs such as word processors, game files, spreadsheets applications, etc.

The file infector virus fixes itself into the host file and begins its operation whenever the file is executed. Here is a snapshot of one such threat detected by antivirus.

Multipara-site Virus - Unlike other types of viruses, the multipartite finds multiple breeding areas for a target. It may attach itself to the boot sector, the executable files or both depending on machine variants like the type of OS and other variables.

Worms: - A computer worm is a self-replicating malware that duplicates itself to spread to uninfected computers. Worms often use parts of an operating system that is automatic and invisible to the user.

- It is common for worms to be noticed only when their uncontrolled replication consumes system resources, slowing or halting other tasks.
- A computer worm is not to be confused with WORM (write once, read many).
- A computer worm infection spreads without user interaction. All that is necessary is for the computer worm to become active on an infected system.
- Before the widespread use of networks, computer worms were spread through infected storage media, such as floppy diskettes, which, when mounted on a system, would infect other storage devices connected to the victim system.
- USB drives are still a common vector for computer worms.

RGPVNOTES.IN

We hope you find these notes useful.

You can get previous year question papers at https://qp.rgpvnotes.in .

If you have any queries or you want to submit your study notes please write us at rgpvnotes.in@gmail.com

LIKE & FOLLOW US ON FACEBOOK
facebook.com/rgpvnotes.in