Sumanjit Das*

Secure your Data by Breaking into IT!

Abstract

In this paper we will take an unusual approach to system security. Instead of merely saying that something is a problem, and show _why_ it is one. We will illustrate that even seemingly harmless network services can become valuable tools in the search for weak points of a system, even when these services are operating exactly as they are intended to. In an effort to shed some light on how more advanced intrusions occur, this paper outlines various mechanisms that crackers have actually used to obtain access to systems and, in addition, some techniques we either suspect intruders of using, or that we have used ourselves in tests or in friendly/authorized environments. My motivation for writing this paper is that system administrators are often unaware of the dangers presented by anything beyond the most trivial attacks.

Introduction

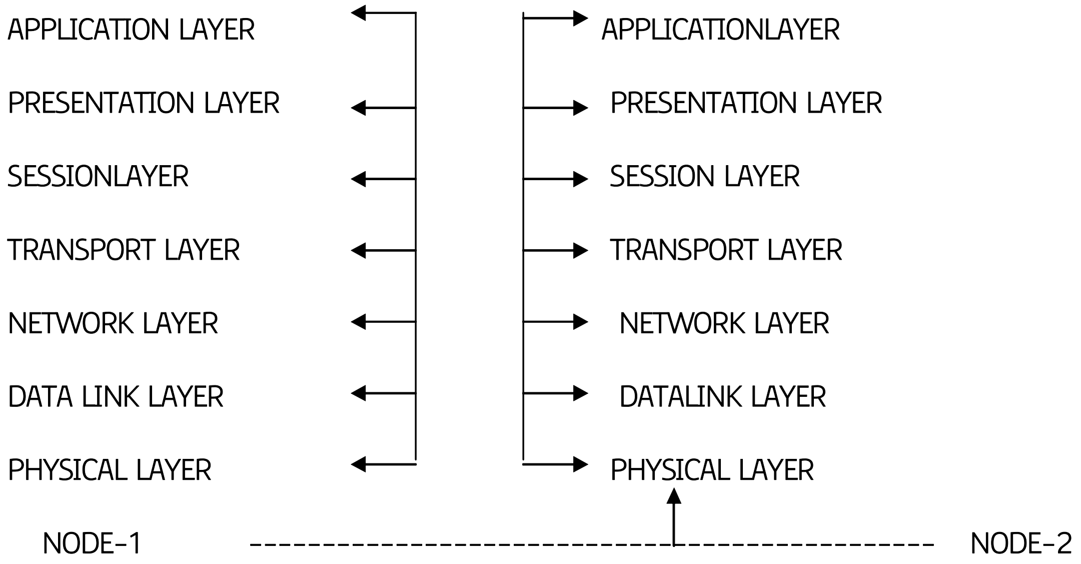
Every day, all over the world, computer networks and hosts are being broken into. The level of sophistication of these attacks varies widely; while it is generally believed that most break-ins succeed due to weak passwords, there are still a large number of intrusions that use more advanced techniques to break in. Less is known about the latter types of break-ins, because by their very nature they are much harder to detect.

To understand the similarity, let's examine the possible entry points for hackers and demonstrate some techniques attackers use to gain access to confidential data. We'll then consider some techniques, including database-level security built into Oracle, for mitigating these risks.

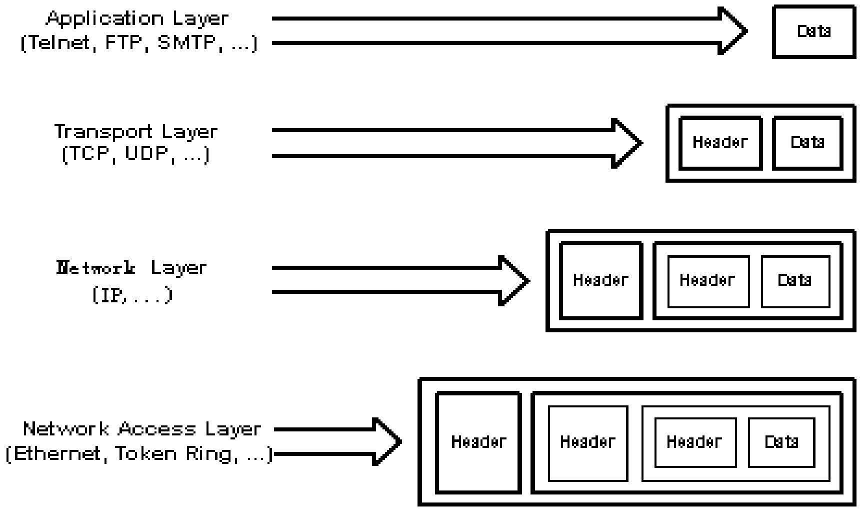
*Faculty MCA, Srusti Academy of Management, Plot No.-38/1, Chandaka Industrial Estate, Near Infocity, P.O.-Patia, Bhubaneswar-31,Odisha, E-mail: dassumanjit@rediffmail.com.

Analyzing the threat

All Web-based applications have numerous possible entry points, and you must check every one. Hackers look at the active IP address to telnet it and try to break into a Web application. When the hacker gets IP of server they try to break the password of the database.Form the figure we will see how the hackers try to access the server and try to hack the data.



OSI MODEL



TCP Security: -

Transmission Control Protocol (TCP) runs on top of IP, and provides a connection oriented service between the sender and the receiver. TCP provides guaranteed delivery, and ensures that the packets are delivered in sequence. The underlying network IP is highly unreliable and does not provide any guarantee for TCP. In order to provide reliability between the sender and the receiver, TCP uses various mechanisms, such as sequence numbers, acknowledgments, 3-way handshakes and timers.

A TCP connection is identified by the 4-tuple **((destination-IP-address, destination-port)**, **(source-IP-address, source-port))**. Ports are the actual end-points of the TCP connection. The working of TCP could be described using a TCP state machine. Transitions to different states are based on events received in the form of TCP segments. The TCP states are very closely associated with **timers**. There are various timers associated with connection establishment (or termination), flow control, and retransmission.

In order to understand the security problems associated with TCP, it is necessary that we look at the state-machine in detail. It is also important to get an overview of TCP implementations, and how they implement the TCP state-machine, the state-transitions and the associated timers.

The TCP layer on either end maintains table entries corresponding to the 4-tuple (remote-IP-address, remote-port, source-IP-address, source-port). This 4-tuple uniquely identifies a connection. For every connection, the end-systems implementing TCP need to keep the TCP state information for the duration of the connection.

Network layer: -

The Internet Protocol (or IP as it generally known), is the network layer of the Internet. IP provides a connection-less service. The job of IP is to route and send a packet to the packet's destination. IP provides no guarantee whatsoever, for the packets it tries to deliver. The IP packets are usually termed datagram's. The datagram's go through a series of routers before they reach the destination. At each node that the datagram passes through, the node determines the next hop for the datagram and routes it to the next hop. Since the network is dynamic, it is possible that two datagram's from the same source take different paths to make it to the destination. Since the network has variable delays, it is not guaranteed that the datagram's will be received in sequence. IP only tries for a best-effort delivery. It does not take care of lost packets; this is left to the higher layer protocols. There is no state maintained between two datagram's; in other words, IP is connection-less.

How easy hacking IP address! :

Internet hackers are searching for servers to attack. For this, they (hackers) write simple scripts that randomly generate and ping IP addresses, looking for servers that respond, "I am live." The response is called a "ping acknowledgement" and is a standard feature of the ping utility, let's see how simple is it? It's not the actual syntax, just a format.

```
C:\ ping 172.168.1.120
```

The out put is.......

```
Pinging 172.168.1.120 with 32 bytes of
data:

Reply from 172. 168.1.120: bytes=32
time=164ms TTL=254

Reply from 172. 168.1.120: bytes=32
time=162ms TTL=254

Reply from 172. 168.1.120: bytes=32
time=170ms TTL=254

Ping statistics for 172. 168.1.120:

Packets: Sent = 5, Received = 5, Lost = 0
(0% loss),

Approximate round trip times in milli-seconds:

Minimum = 167ms, Maximum = 175ms,
Average = 171ms
```

The acknowledgement packet tells the hacker that there's an active server at this IP address. Then the hacker telnets to the server and begins to hack the root or the Oracle user password. The best way to foil this type of attack is to disable all server accounts after three password attempts.

Let's have an example for generating an IP address. The code in a UNIX shell script to internet for vulnerable servers. Hackers run such scripts as daemon processes and scan hundreds of thousands of IP addresses every hour. Please note that I have deliberately introduced syntax errors into the pseudo code routine to prevent it being used by any wannabe hackers. So don't try it in your system!!

```
/*$/bin/bsh

while true

do

$========================================

$ Generate a random IP address

$========================================

$IP_ADD=rnd(1-254).rnd(1-254).rnd(1-
254).rnd(1-254)

$========================================

$ Submit the IP address to the ping
command

$========================================

nohup ping $IP_ADD > /tmp/r.lst 2>&1 &

$========================================

$ If ping is responding – start the attack

$========================================

if 'cat /tmp/r.lst|wc –l' > 0 then invoke
attack _routine fi done
```

Even a computer user can write an attack program and locate server attack opportunities. Although the main method of attack is directly from the IP address, some creative hackers gain entry with I/O-enabled Java applets or programs that compromise cookie-writing. To prevent these types of external attacks, savvy companies employ some of the following techniques:

Add-On Authentication: Kerberos

Another method of limiting an attacker's spoofing abilities is to add authentication onto the application layer. Of course, just adding authentication is not enough without adding encryption; otherwise, after some initial application-level authentication, a hijacking attack may still be successful. An authentication between two parties which exchanges a session key however is secure; even though the IP packets transmitted back and forth are not individually authenticated, they are all encrypted with the secure session key. This scheme is the goal of the Kerberos Authentication System, developed at MIT. The Kerberos system uses cryptographic authentication algorithms to ensure that a user is really who s/he claims to be, and once this is established, an exchanged session key is used to encrypt all transmissions of whatever service the user has requested. Without knowledge of this session key, it is impossible for an attacker to spoof meaningful transmissions between sources. Since this key is generated based on secret keys known only to the actual user and the trusted server, it is very hard for an attacker to acquire. The Kerberos system is resilient to replay attacks as well. Kerberos is generally considered to significantly increase the security of a network, although it is not a panacea. There are problems using Kerberos to authenticate between two machines (instead of a user and a machine), and there are difficulties involving where the keys are cached on a multi-user machine.

Encrypting Individual IP Packets (SKIP)

Instead of exchanging a session key, as we might do via Kerberos for a telnet session, we could choose to encrypt all IP packets, all the time, at the IP level. Naturally, we must encrypt them in a way that the destination can successfully decrypt them. There is a special key-distribution scheme designed for packet-level encryption called the Simple Key-

Management for Internet Protocols (SKIP). SKIP assumes that each site in the network has a public key, which can be used to create many keys between two sites. The basic idea of the algorithm is to encapsulate the packet-key (the key to decrypt that packet) *inside* the packet, and encrypt that with shared secret between the two sites. The SKIP technique also provides an easy method of changing the shared secret between two sites. Since this method is not session-based, it can cover all aspects of TCP/IP communication, not merely applications.

Restricting server access:

If possible, servers should not be accessible over the Internet unless network and systems administrators have followed the general guidelines for authenticated external access. Some companies use domain servers to restrict server access to specified users. However, hackers still might intercept user IDs and passwords. To prevent this, many companies employ tools that utilize secure shell technologies to encrypt external Internet communications. The most popular of these tools is Secure CRT, which gives authorized users Internet access to servers without the fear of someone capturing the user ID and password.

Secure shell tools use sophisticated Huffman cryptography techniques for Internet transmissions; these products are more secure even than the Enigma code that was used during World War II. However, such superb encryption sometimes lulls IT staffs into believing that they are protected from external attack. Remember, the bulk of the security is at the server firewall, not on the Internet.

Trusted IP:

A key concept that appears in both the authentication and confidentiality mechanisms for IP is the *Security Association* (SA). An association is a one-way relationship between a sender and a receiver that affords security services to the traffic carried on it. If a peer relationship is needed, for two-way secure exchange, then two security associations are required. Security services are afforded to an SA for the use of AH or ESP, but not both. A security association is uniquely identified by three parameters:

- Security Parameters Index (SPI): The SPI assigns a bit string to this SA that has local significance only. The SPI is carried in AH and ESP headers to enable the receiving system to select the SA under which a received packet will be processed.
- IP destination address: Currently, only unicast addresses are allowed; this is the address of the destination endpoint of the SA, which may be an end-user system or a network system such as a firewall or router.
- Security protocol identifier: This indicates whether the association is an AH or ESP security association.

Hence, in any IP packet, the security association is uniquely identified by the destination address in the IPv4 or IPv6 header and the SPI in the enclosed extension header (AH or ESP).

An IPSec implementation includes a security association database that defines the parameters associated with each SA. A security association is defined by the following parameters:

- Sequence number counter : A 32-bit value used to generate the sequence number field in AH or ESP headers
- Sequence counter overflow : A flag indicating whether overflow of the sequence number counter should generate an auditable event and prevent further transmission of packets on this SA
- Anti-replay window : Used to determine whether an inbound AH or ESP packet is a replay, by defining a sliding window within which the sequence number must fall
- AH information : Authentication algorithm, keys, key lifetimes, and related parameters being used with AH

- ESP information : Encryption and authentication algorithm, keys, initialization values, key lifetimes, and related parameters being used with ESP
- Lifetime of this security association : A time interval or byte count after which an SA must be replaced with a new SA (and new SPI) or terminated, plus an indication of which of these actions should occur
- IPSec protocol mode : Tunnel, transport, or wildcard (required for all implementations); these modes are discussed later
- Path MTU : Any observed path maximum transmission unit (maxi-mum size of a packet that can be transmitted without fragmentation) and aging variables (required for all implementations)

Restricting database access:

Now that we have reviewed server access, let's explore port (Application access) access. All Web-enabled applications have a listener process that checks a specific port for incoming database requests.

Inside the database, companies run the risk of allowing Web users unauthorized access to information. In an internal environment, each user is clearly identified. On the Web, anybody can try to access the application. It's up to the database administrator to ensure that everyone who accesses the application has the proper authenticity. Only authenticated user can access the data.

Conclusion:

Making your data secure for broadcast over the Internet or Intranet is no easy task. The best way to evaluate your security needs is to weigh the disadvantages of unauthorized users seeing your data. The more privacy your data requires, the more security you should have in place. Security is often far easier to implement than to maintain. Make sure that the necessary processes in your organization keep your system current. Security is often neglected or bypassed by day-to-day users for convenience. By the time loss of data or security breaks have been discovered, much damage may have already been done.

Database and Web server vendors are constantly upgrading their systems, so staying current on the particular architectures is a must. For general security information available on the Web, the best place to begin your search is with the National Computer Security Association (NCSA).

Data is the lifeblood of an organization's IT infrastructure, so it must be protected at all costs. You have some highly available options to jump-start security for your organization's database servers.

This is not "THE END" it's the trailer. My research will go on for giving you more information about security wait for some time I will be back!!!!

References:

1 Das Sumitra, UNIX, New delhi, Tata McGraw- Hill, 2005

2 Forouzon Behrouz A, fegon S.C., data communication & networking, Tata Mc Graw- Hill, 2005

3 Fleeger P.P. , Pfleeger, security in computing, 20006

4 Corner Douglas E., internetworking with TCP/IP principle, protocol and architecture, PHI, 2004

5 Troudet T.P., Walters S.M., Newral Network Architecture, 1991

6 Barrel A., Nelson B.J., "implementing remote procedure call", transaction on computer system, vol-2, pp. 39-59.

7 Chalasomi sandeep, trippireddy suresh, period of the d- sequence based random number generator

www.arxiv.org

8 Hilker Michael, Schommer christoph, service oriented architecture in N/W security – a novel organization in security systems.

http://arxiv.org/find/grp_csl/1/

and t ti: :+security/01/0/2008/0/1