Efficient Method for Preventing Password Sniffing Using MD5 Algorithm

Vikram Singh[1], Manoj Kumar[2], Lekh Raj[3]

[1]Prof. Dept. Computer Science & Application, Chaudhary Devi Lal University, Haryana, India
[2,3] Student M.Tech. Dept. Computer Science & Application, Chaudhary Devi Lal University, Haryana, India

Abstract— *Internet usage is increasing day by day. Online financial (cash) transactions are increasing day by day like online Banking, online shopping etc With Such increased internet transactions, the need for the security also arises as hackers are being active to have access the private information like user id/Password, credit card no, pin etc. The objective of sniffing is to steal Password (E mail, FT, SQL Telnet, Web Etc), Email text etc. This paper focused on how to detect and prevent Password Sniffing on web applications using MD5 Algorithm encryption technique. The goal of this paper is to provide improved security by developing a method which prevents illegal access to the web applications.*

Keywords— Sniffing, Password attacks, password sniffing, MD5 Algorithm.

I. INTRODUCTION TO SNIFFING

Sniffing is a program/device that captures the vital information network traffic to a particular traffic. Sniffing is a data interception technology. The objective of sniffing is to steal Password (E mail, FT, SQL Telnet, Web etc.), Email text, FTP (FTP Files, Email Files, SMB) etc.

Many attackers can capture passwords simply by using keyboard-sniffing Trojan horse or one of the many physical keyboard-logging hardware devices for sale on the Internet. Most steal passwords. Anyone can buy a keyboard keystroke logger that can log more than 2 million keystrokes. Physical keyboard logging devices less than an inch long can easily be slipped between the keyboard cord and the computer's keyboard port. And let's not forget how easy it is to sniff passwords from wireless keyboards even from a city block away.

Password sniffing - Some password crackers can sniff authentication traffic between a client and server and extract password hashes or enough authentication information to begin the cracking process. Cain & Abel both sniffs authentication traffic and cracks the hashes it retrieves. Other sniffing password crackers are ScoopLM (http://www.securityfriday.com/tools/ScoopLM.html) and KerbCrack (http://ntsecurity.nu/toolbox/kerbcrack), a sniffer and cracker for cracking Kerberos authentication traffic. None of these can crack NTLNv2 authentication traffic.

Sniffing involves capturing, decoding, inspecting and interpreting the information inside a network packet on a TCP/IP network. The purpose is to steal information, usually user IDs, passwords, network details, credit card numbers, etc. Sniffing is generally referred to as a "passive" type of attack, wherein the attackers can be silent/invisible on the network. This makes it difficult to detect, and hence it is a dangerous type of attack.

TCP/IP packet contains vital information required for two network interfaces to communicate with each other. It contains fields such as source and destination IP addresses ports, sequence numbers and the protocol type. Each of these fields is crucial for various network layers to function, and especially for the Layer 7 application that makes use of the received data.

By its very nature, the TCP/IP protocol is only meant for ensuring that a packet is constructed, mounted on an Ethernet packet frame, and reliably delivered from the sender to the receiver across networks. However, it does not by default have mechanisms to ensure data security. Thus, it becomes the responsibility of the upper network layers to ensure that information in the packet is not tampered with.

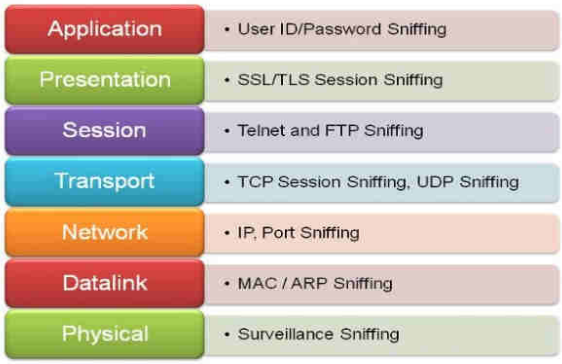


Fig. 1: Mapping of OSI layers

II. TYPES OF PASSWORD ATTACKS

Password Guessing

Password guessing is most common type of attack. Password guessing isn't always as difficult as you'd expect Attackers can guess passwords locally or remotely

using either a manual or automated approach. Most networks aren't configured to require long and complex passwords, and an attacker needs to find only one weak password to gain access to a network. Not all authentication protocols are equally effective against guessing attacks. For example, because LAN Manager Authentication is case-insensitive, a password guessing attack against it doesn't need to consider whether letters in the password are uppercase or lowercase.

Many tools can automate the process of typing password after password. Some common password guessing tools are Hydra (see http://www.thc.org for links to the downloadable tool), for guessing all sorts of passwords, including HTTP, Telnet, and Windows logons; TSGrinder (http://www.hammerofgod.com/download.htm), for brute-force attacks against Terminal Services and RDP connections; and SQLRecon (http://www.sqlsecurity.com/DesktopDefault.aspx?tabid=26), for brute-force attacks against SQL authentication (a brute-force attack, or exhaustive key search, is a cryptanalytic attack that can, in theory, be used against any encrypted data).

Automated password guessing programs and crackers use several different approaches. The most time consuming—and most successful—attack method is the brute-force attack, in which the attacker tries every possible combination of characters for a password, given a character set (e.g., abcd…ABCD…1234…!@#$) and a maximum password length.

Dictionary attacks work on the assumption that most passwords consist of whole words, dates, or numbers taken from a dictionary. Dictionary attack tools require a dictionary input list. You can download varying databases with specific vocabularies (e.g., English dictionary, sports, even Star Wars trivia) free or commercially off the Internet.

Hybrid password guessing attacks assume that network administrators push users to make their passwords at least slightly different from a word that appears in a dictionary. Hybrid guessing rules vary from tool to tool, but most mix uppercase and lowercase characters, add numbers at the end of the password, spell the password backward or slightly misspell it, and include characters such as @!# in the mix.

Password Resetting

Attackers often find it much easier to reset passwords than to guess them. Many password cracking programs are actually password resetters. In most cases, the attacker boots from a floppy disk or CD-ROM to get around the typical Windows protections. Most password resetters contain a bootable version of Linux that can mount NTFS volumes and can help you locate and reset the Administrator's password.

Password Cracking: Although password resetting is a good approach when all you need is access to a locked computer, resetting passwords attracts unwelcome attention. Attackers usually prefer to learn passwords without resetting them. Password cracking is the process of taking a captured password hash (or some other obscured form of the plaintext password or challenge-response packets) and converting it to its plaintext original. To crack a password, an attacker needs tools such as extractors for hash guessing, rainbow tables for looking up plaintext passwords, and password sniffers to extract authentication information.

Hash guessing: Some password cracking tools can both extract and crack password hashes, but most password crackers need to have the LM password hash before they can begin the cracking process. (A few tools can work on NT hashes.) The most popular Windows password hash extractor is the Pwdump family of programs. Pwdump has gone through many versions since its release years ago, but Pwdump4 is the current version. (You can download Pwdump4 at http://pr.openwall.net/dl/pwdump/pwdump4.zip.)

To extract password hashes using Pwdump, you must have administrative access to the local or remote machine you're attacking, and you must be able to use NetBIOS to connect to the admin$ share. There are ways around the latter requirement, but the tool alone requires it. When you run Pwdump4 successfully, it extracts LM and NT password hashes and, if Windows' password history tracking is active, all hashes for older passwords. By default, Pwdump saves password hashes to the screen, but you can also output them to a file, then feed them to a password cracker.

Many password cracking tools accept Pwdump-formatted hashes for cracking. Such tools usually begin the cracking process by generating some guesses for the password, then hashing the guesses and comparing those hashes with the extracted hash.

Password sniffing: Many attackers can capture passwords simply by using keyboard-sniffing Trojan horse or one of the many physical keyboard-logging hardware devices for sale on the Internet. Symantec reports that 82 percent of the most commonly used malware programs steal confidential information. Most steal passwords. Anyone can buy a keyboard keystroke logger that can log more than 2 million keystrokes. Physical keyboard logging devices less than an inch long can easily be slipped between the keyboard cord and the computer's keyboard port. And let's not forget how easy it is to sniff passwords from wireless keyboards even from a city block away.

Some password crackers can sniff authentication traffic between a client and server and extract password hashes

or enough authentication information to begin the cracking process.

Sniffing involves capturing, decoding, inspecting and interpreting the information inside a network packet on a TCP/IP network. The purpose is to steal information, usually user IDs, passwords, network details, credit card numbers, etc. Sniffing is generally referred to as a "passive" type of attack, wherein the attackers can be silent/invisible on the network. This makes it difficult to detect, and hence it is a dangerous type of attack.

Password Capturing: Many attackers capture passwords simply by installing a keyboard-sniffing Trojan horse or one of the many physical keyboard-logging hardware devices for sale on the Internet. Symantec reports that 82 percent of the most commonly used malware programs steal confidential information. Most steal passwords. Anyone can buy a keyboard keystroke logger that can log more than 2 million keystrokes. Physical keyboard logging devices less than an inch long can easily be slipped between the keyboard cord and the computer's keyboard port. And let's not forget how easy it is to sniff passwords from wireless keyboards even from a city block away.

Detecting of Password Sniffing

Dsniff

The dsniff tool is a member of the Dsniff suit toolset , it's an advanced password sniffer that recognizes several different protocols, including TELNET, FTP, SMTP, Post Office Protocol (POP), Internet Message Access Protocol (IMAP), HTTP, CVS, Citrix, Server Message Block (SMB), Oracle, and many others.

Cain & Abel tool

Cain & Abel is a password recovery tool for Microsoft Operating Systems. It allows easy recovery of various kind of passwords by sniffing the network, cracking encrypted passwords using Dictionary, Brute-Force and Cryptanalysis attacks, recording VoIP conversations, decoding scrambled passwords, recovering wireless network keys, revealing password boxes, uncovering cached passwords and analyzing routing protocols. The program does not exploit any software vulnerabilities or bugs that could not be fixed with little effort. It covers some security aspects/weakness present in protocol's standards, authentication methods and caching mechanisms; its main purpose is the simplified recovery of passwords and credentials from various sources, however it also ships some "nonstandard" utilities for Microsoft Windows users.

Wireshark

WireShark is a Fee packet sniffing tool. WireShark uses the winpcap to capture the packets so, it can only capture the packet on the network supported by winpcap. Captured files can be programmatically edited via command line. Wireshark will give us tons of additional information about the connection and the individual packets.

Wireshark has a rich feature set which includes the following:

- Deep inspection of hundreds of protocols, with more being added all the time
- Live capture and offline analysis
- Standard three-pane packet browser
- Multi-platform: Runs on Windows, Linux, OS X, Solaris, FreeBSD, NetBSD, and many others
- Captured network data can be browsed via a GUI, or via the TTY-mode TShark utility
- The most powerful display filters in the industry
- Rich VoIP analysis
- Read/write many different capture file formats: tcpdump (libpcap), Pcap NG, Catapult DCT2000, Cisco Secure IDS iplog, Microsoft Network Monitor, Network General Sniffer® (compressed and uncompressed), Sniffer® Pro, and NetXray®, Network Instruments Observer, NetScreen snoop, Novell LANalyzer, RADCOM WAN/LAN Analyzer, Shomiti/Finisar Surveyor, Tektronix K12xx, Visual Networks Visual UpTime, WildPackets EtherPeek/TokenPeek/AiroPeek, and many others
- Capture files compressed with gzip can be decompressed on the fly
- Live data can be read from Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB, Token Ring, Frame Relay, FDDI, and others (depending on your platform)
- Decryption support for many protocols, including IPsec, ISAKMP, Kerberos, SNMPv3, SSL/TLS, WEP, and WPA/WPA2
- Coloring rules can be applied to the packet list for quick, intuitive analysis
- Output can be exported to XML, PostScript®, CSV, or plain text

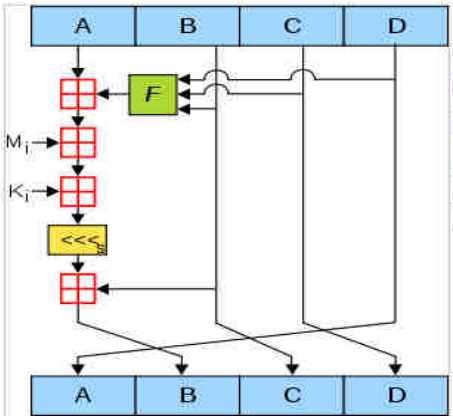
III. PROPOSED MODEL

The proposed system focuses on how to prevent Password sniffing on web applications using MD5 algorithm at application level. The password is encrypted using MD5 (Message Digest 5) algorithm which is fast, and requires little memory. The MD5 message digest algorithm is a widely used cryptographic hash function producing a 128 bit (16 byte) has value and one way algorithm (No Decryption), typically expressed in test format as a 32 digit hexadecimal number. MD5 has been utilized in a wide variety of cryptographic application. In fig 2, explained the MD5 algorithm working & in fig 3. Proposed model is explained.

MD5 Algorithm

MD5 processes a variable-length message into a fixed-

length output of 128 bits. The input message is broken up into chunks of 512-bit blocks (sixteen 32-bit words); the message is padded so that its length is divisible by 512. The padding works as follows: first a single bit, 1, is appended to the end of the message. This is followed by as many zero's as are required to bring the length of the message up to 64 bits less than a multiple of 512. The remaining bits are filled up with 64 bits representing the length of the original message, modulo 2^{64}.



In figure2, One MD5 operation. MD5 consists of 64 of these operations, grouped in four rounds of 16 operations.

F is a nonlinear function; one function is used in each round.

Mi denotes a 32-bit block of the message input

Ki denotes a 32-bit constant, different for each operation.

left shifts denotes a left bit rotation by s places; s varies for each operation. Addition denotes addition modulo 232.

The main MD5 algorithm operates on a 128-bit state, divided into four 32-bit words, denoted A, B, C, and D. These are initialized to certain fixed constants. The main algorithm then uses each 512-bit message block in turn to modify the state. The processing of a message block consists of four similar stages, termed rounds; each round is composed of 16 similar operations based on a non-linear function F, modular addition, and left rotation. Figure 1 illustrates one operation within a round. There are four possible functions F; a different one is used in each round:

$$F(B,C,D) = (B \wedge C) \vee (\neg B \wedge D)$$
$$G(B,C,D) = (B \wedge D) \vee (C \wedge \neg D)$$
$$H(B,C,D) = B \oplus C \oplus D$$
$$I(B,C,D) = C \oplus (B \vee \neg D)$$

$\oplus, \wedge, \vee, \neg$ denote the XOR, AND, OR and NOT operations respectively.

IV. PROPOSED MODEL ALGORITHM

Step 1: Request received from the Client side /User

Step 2: Generate Key using Random number generation algorithm & respond to Client/user. While responding to Client/user, send Key in hidden variable

Step 3: Client/User entered UserID & Password. Submit the request. During Submission at Client side, Password gets encrypted through MD5 using Key

Step 4: Send the encrypted values & UserID to server (or we can say submit the reques)t.

Step 5: At Server, Fetch password from DB for same user.

Step 6: Convert the password to encrypted value using MD5 & same Key

Step 7: Compare the client & Server encrypted values.

Step 8: if match then provide the access else Block the user

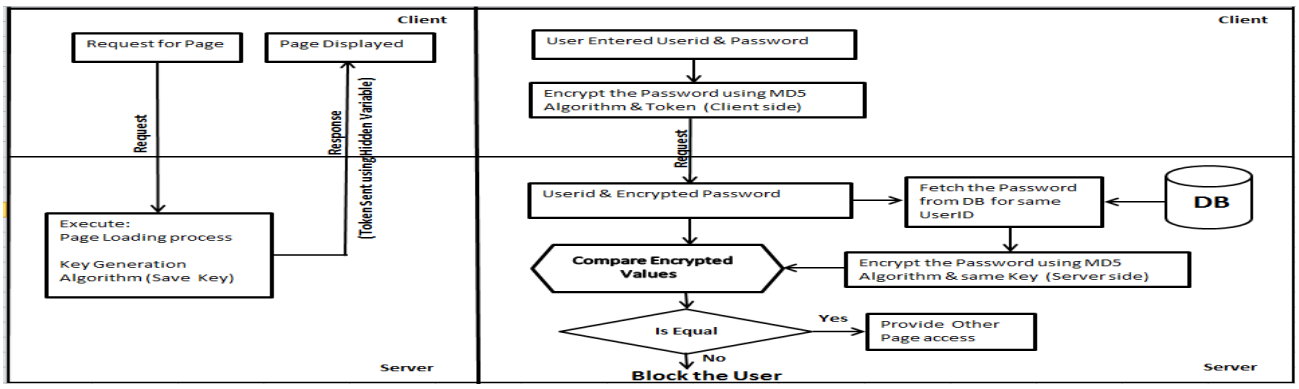


Fig.2: Proposed Architecture

V. CONCLUSION AND FUTURE WORK

This paper has presented a method to prevent Password

sniffing using MD5 (Message Digest 5 algorithm). The password is encrypted using MD5 (Message Digest 5) algorithm which is fast, and requires little memory. The MD5 message digest algorithm is a widely used cryptographic hash function producing a 128 bit (16 byte) has value. The advantage is MD5 is one way algorithm means there is no way to decrypt the MD5 algorithm encryptions value. It is impossible to decrypt. This approach does not require major changes to application code and has negligible effect on performance. It can also be easily applied to any other language without major changes. Further explore on the encryption & secure way to protect the password.

REFERENCES

[1] Vishal Mishra, Nidhi Verma of *Analysis on password sniffing "Security against Password Sniffing using Database Triggers"*, International Journal of Research in Advent Technology, Vol.2, No.3, March 2014 E-ISSN: 2321-9637.

[2] Manu Kumar; Tal Garfinkel; Dan Boneh; Terry Winograd;(2007) Reducing Shoulder-surfing by "*Using Gaze-based Password Entry; 3rd symposium on Usable privacy and security"*; ACM Society; pp. 13-19.

[3] Li Yinxiang; Lizhi Zhong; (2010) Research on the S / KEY "*One-Time Password Authentication System and its Application in Banking And Financial Systems*"; International Conference on Digital Content, Multimedia Technology and its Applications, Seoul, Korea; IEEE ;pp. 172-175.

[4] Pallavi Asrodia, Hemlata Patel, "Network traffic analysis using packet sniffer", International Journal of Engineering Research and Application (IJERA), Vol.2, pp. 854-857, Issue 3, May-June 2012.

[5] Rupam1, Atul Verma2 "*An Approach to Detect Packets Using Packet Sniffing*", International Journal of Computer Science & Engineering Survey (IJCSES) Vol.4, No.3, June 2013

[6] Varsha Khokhar, Shehnaz Khan, Priyanka Muppuri, Prachi Ahlawat, "*Sniflyzer: A Network Sniffer*" OPEN JOURNAL OF INFORMATION SECURITY AND APPLICATIONS, ISSN(Print): 2374-6262 ISSN(Online): 2374-6289, DOI: 10.15764/ISA.2014.02001, Volume 1, Number 2, September 2014

[7] Pallavi Asrodia , Mr. Vishal Sharma "*Network Monitoring and Analysis by Packet Sniffing Method"* International Journal of Engineering Trends and Technology (IJETT) - Volume4 Issue5- May 2013 ISSN: 2231-5381

[8] Qadeer M.A., Zahid M., Iqbal A., Siddiqui M.R "*Network Analysis and Intrusion Detection Using Packet Sniffer ICCSN* ' Second International Conference, 2010, Page(s): 313 – 317

[9] S. Ansari, Rajeev S.G. and Chandrasekhar H.S, "*Packet Sniffing: A Brief Introduction*", IEEE Potentials, Dec 2002- Jan 2003, Volume: 21 Issue: 5, pp: 17 – 19

[10] A. Dabir, A. Matrawy, "Bottleneck Analysis of Traffic Monitoring Using Wireshark", 4th International Conference on Innovations in Information Technology, 2007, IEEE Innovations '07, 18-20 Nov 2007, Page(s):158 – 162

[11] Inderjit Kaur , Harkarandeep Kaur , Er. Gurjot Singh "*Analysing Various Packet Sniffing Tools*" International Journal of Electrical Electronics & Computer Science Engineering Volume 1, Issue 5 (October 2014), ISSN : 2348 2273

[12] Wolf-Bastian P¨ottner, and Lars Wolf, "*IEEE 802.15.4 packet analysis with Wireshark and off-the-shelf hardware*", Institute of Operating Systems and Computer Networks, Technische Universit¨at Braunschweig, Germany.

[13] https://en.wikipedia.org/wiki/MD5

[14] https://www.techopedia.com

[15] www.packet-sniffer.net/

[16] http://opensourceforu.efytimes.com

[17] https://www.wireshark.org/