Journal of Engineering Science and Technology
Vol. 11, No. 12 (2016) 1744 - 1757
© School of Engineering, Taylor's University

SESSION – PACKET INSPECTOR MOBILE AGENT TO PREVENT ENCRYPTED COOKIES AND HTTP POST HIJACKING IN MANET

ARUN V.*, SHUNMUGANATHAN K. L.

Research Scholar, Department of Computer Science and Engineering,
Sathyabama University, Chennai, India
*Corresponding Author: arunvragavan@gmail.com

Abstract

A statistics states that public Wi-Fi hotspots are set to grow from 1.3 million in 2011 to 5.8 million by 2015. Due to increase in public free Wi-Fi access, users count increases rapidly letting them to expose in vulnerable intrusion attacks. Accessing the social network and other confidential sites such as banking in public Wi-Fi, exposes the credential to the intruder. Man-in-the-middle attack plays a major role in accessing the user credential. Analysing the packets exposes the HTTP post information in the network. Capturing encrypted cookies by the man-in-the-middle can access the secure site and allows the intruder to be an authorized person. Mobile agent is a piece of code that migrates and performs actions in the host and is used to prevent intrusion. We propose a model to prevent intrusion in the wireless ad-hoc network using mobile agent. Session-Packet inspector is proposed to handle HTTP post hijacking and encrypted cookie usage by the intruder.

Keywords: Intrusion detection, mobile agent, MANET, cookie, HTTP.

1. Introduction

Wi-Fi hotspot offers internet access which is connected to the internet service provider. Public Wi-Fi hotspot becomes popular which provides internet access to the public for free. Many public places such as malls, hotels provide free Wi-Fi access with higher bandwidth. Accessing the confidential sites using the public Wi-Fi hotspot is vulnerable and exposes the credentials to the intruder. Since many people are connected to the Wi-Fi hotspot, man-in-the-middle attack threatens the users. Nowadays cookies are widely used to store the credential and

Abbreviations

AODV	Ad hoc On-Demand Distance Vector
CC	Challenge Collapsar
DOS	Denial of Service
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
RIPEMD	RACE Integrity Primitives Evaluation Message Digest
SSL/TLS	SSL – Secure Socket Layer; TLS – Transport Layer Security
TCM	Tracer Checker Model
Wi – Fi	Wireless Fidelity

webpage information in client site. Those cookies are transferred between server and client during authentication process. These cookies even though encrypted may expose the use of encrypted cookie by the intruder. Since the wireless ad-hoc network does not rely on any pre-existing infrastructure, each node in the network may participate in routing packets. Man-in-the-middle attack is vulnerable for this network and can access the header of the packets to access the credential details. Hotspots are ad-hoc network and may expose the packet header to the intruders.

Weak session identifier generated by web applications is the major cause for the session hijacking attacks [1] cross-site scripting (XSS) enables hijacker to inject client-side script into the web pages and bypass access controls of the victim. Cross-site scripting accounts 84% of the data handled vulnerability documented by Symantec in 2007 which includes session hijacking technique.

Cookies are stored in the browser and are used to maintain the state for web sites. Authentication cookies are sent to the server to authenticate the user [2]. These cookies hold the credential information which is vulnerable. SSL/TLS are widely used to encrypt the cookies and transmitted among server and client. But man-in-the-middle can access the encrypted cookie and gain access as another user. Mobile Agent is a piece of code that can actually run on the client machine to perform actions and gather information to the server. We propose an effective algorithmic model to handle the credential information migration from server to the client. We used mobile agent to handle the information between the server and client.

Figure 1 illustrates the man-in-the-middle attack in public Wi-Fi hotspot. The Hacker may gain access as a normal user to the hotspot and can access the packets transferred among the hosts [3]. Since it is an ad-hoc network, each host involve in routing strategy. Http post may involve in transferring the credential information on the header. Inspecting the http post header packets may result in credential disclosure. When a user allows cookie authentication in browser, it supports automatic login. Even though cookie can be encrypted, the intruder can get the encrypted cookie and sent to the server to gain access. So we proposed a model agent algorithmic approach to protect the header information and cookie.

TCM server

Figure 2 illustrates the each TCM server holds the Tracer Mobile Agent (TMA) [4]. The duty of the mobile agent is to check for the behavior and accessed resources by the mobile agent that arrives to the host. After checking the certificate, if it is authorized, then the mobile agent is allowed to visit the host. If suppose, hacker achieved the key to create the certificate like authorized one, TMA is used to identify it. TMA Checks for the behavior of the arrived mobile agent. If it is performing certain unwanted or malicious action in the host, then the TMA alarm the TCM server. TCM Server uses Tracer Mobile Agent to check any intrusion in host and raise alarm in the system. Checker Mobile Agent prevents intrusion in other host by checking the malicious host. Mobile host is registered with TCM and in addition with unique ids generated in TCM, it carries sandbox UID (Unique Identification) for each host.

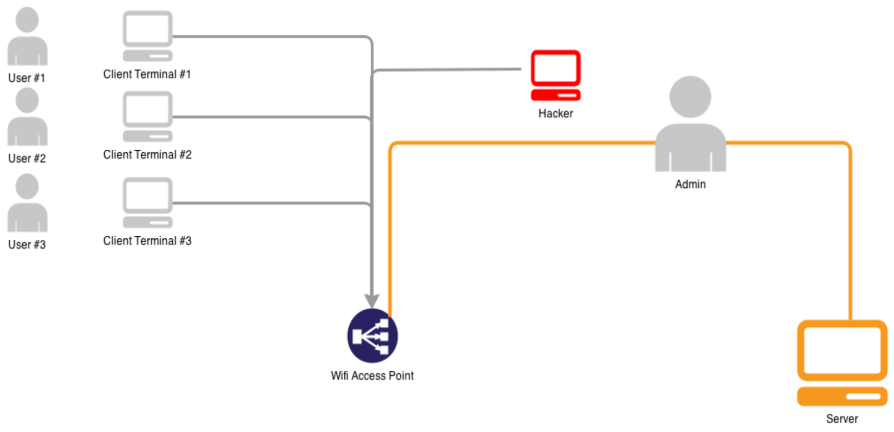


Fig. 1. Attack in mobile ad-hoc wireless network.

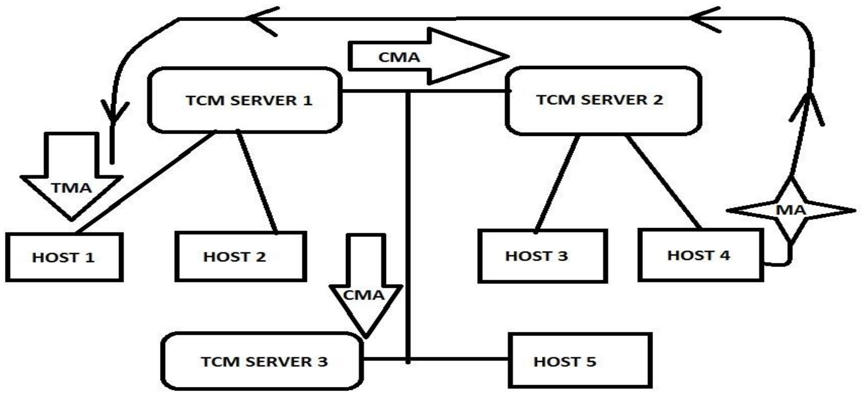


Fig. 2. TCM architecture.

2. Related Work

Putthacharoen. R et al. [1], described dynamic cookies rewriting technique to overcome XSS attacks. Cross-site scripting (XSS) attacks enable attackers to inject the client-side scripting into the web pages. Dynamic Cookies Rewriting technique is developed to render the cookies useless for XSS attacks. This technique is implemented in a web proxy and it will automatically rewrite the cookies that are sent back and forth by the user and application. The browser cookie are not valid for web application hence XSS attacks will not be performed.

Xia Chun-Tao et al [3], analyzed the basic technique used for CC attacks and proposed an algorithm to avoid CC attacks. CC attack destroys the characteristics of web page's random distribution. Proposed algorithm detects the attack and uses HTTP redirection and cookie to block the attacks. A network design is proposed that can handle the attacks and prevents the depletion of resources.

VratonjicN et al [5], declares that user's location privacy is compromised by the hackers especially at the access points which has a single public IP and uses network address translation. Location privacy can be compromised by accessing the hotspot with the geographical location of the user when user is connected to the access point. A novel location-privacy causing the single IP access point is traced down and related user's location are compromised. Vratonjic proposed a new model to prevent the location privacy.

YiXie Tang S et al [6] introduced a new hidden Demi-Markov process parameterized by Gaussian-mixture. Gaussian distribution is proposed that are time-varying traffic behavior of web proxies. A novel server side defense scheme is proposed that resist the web proxy distribution DOS attack. Soft control converts the suspicious traffic into relatively normal by behavioral reshaping rather than discarding the traffic. This helps to provide quality of services to the legitimate users.

Jin Wang et al [7], proposed a novel anomaly-based HTTP- flooding detection scheme to eliminate the influence of the Web-crawling algorithm. Most of the system does not filter the Web-crawling traces of the unknown bots mixed with the normal Web browsing logs. The proposed model is immune to the interferences of unknown search session and detected all HTTP-flooding attacks.

Yue Li et al [8], described a web log processing technique. Due to incomplete sessions, Cookie pair for the session are merged. Few Cookie pairs can be used by websites for identification. Two algorithms are proposed to identify user-Cookies and terminal-Cookies. With these key-Cookies the process of session merging is improved.

3. HTTP Attacks

Wireless ad-hoc is subjected to various attacks. Passive attacks are unidentifiable and Wi-Fi hotspots are vulnerable for it. We proposed this model that avoids major attacks [9]. We exercised on http header hijacking attack and encrypted cookie capture attack using existing tools.

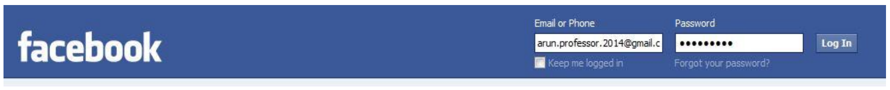
3.1. Hijacking tools

3.1.1. Facebook password hack using packet sniffing

User authentication credentials are posted to the server for processing the request. This authentication header carries the username and password that are required to login into web application. Existing application such as firesheep and wireshark are widely used to listen to the header that transmits through the network [10]. This tool captures the packets in the network and analyzes the content in the packets. Figure 3(a) shows the login page of facebook. Figure 3 (b) shows the hijacked information using firesheep tool which is used by the hacker in same Wi-Fi hotspot.

3.1.2. Firesheep

Firesheep is an open source tool used to identify the headers in HTTP request. It is an extension in browser that intercepts unencrypted cookies from websites. It allows user to login into the credential of the user from the HTTP header.



facebook

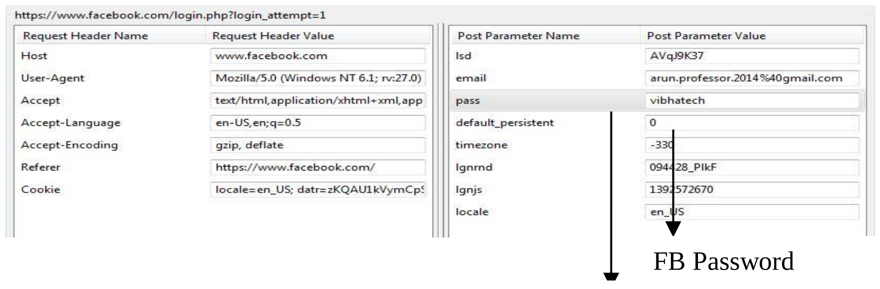
Email or Phone	arun.professor.2014@gmail.c
Password	•••••••••

Log In

Keep me logged in

Forgot your password?

(a) Facebook login page.



https://www.facebook.com/login.php?login_attempt=1

Request Header Name	Request Header Value
Host	www.facebook.com
User-Agent	Mozilla/5.0 (Windows NT 6.1; rv:27.0)
Accept	text/html,application/xhtml+xml,app
Accept-Language	en-US,en;q=0.5
Accept-Encoding	gzip, deflate
Referer	https://www.facebook.com/
Cookie	locale=en_US; datr=zKQAU1kVymCp

Post Parameter Name	Post Parameter Value
lsd	AVqJ9K37
email	arun.professor.2014%40gmail.com
pass	vibhatech
default_persistent	0
timezone	-330
lgnrnd	094428_PIkF
lgnjs	1392572670
locale	en_US

(b) Facebook user credentials.

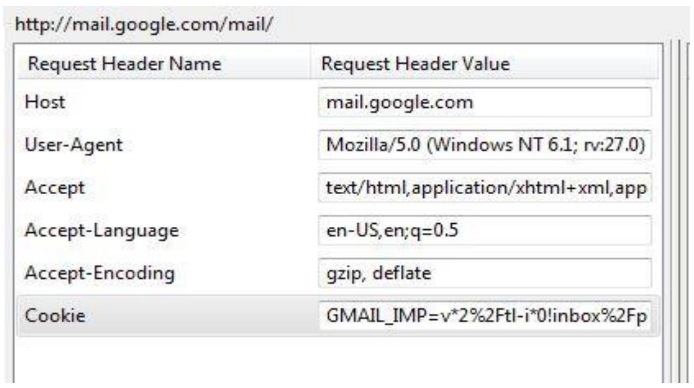
Fig. 3. HTTP hijacked with facebook user credentials using firsheep.

3.2. Attack techniques

3.2.1. Hijacking in google account using cookie access

Cookies are used to save information in client browser to maintain state by the server. Nowadays cookies are encrypted and are transmitted in the network7. Encrypted cookie from client acts as a authentication ticket for accessing the resource by the server. Man-in-the-middle can capture the encrypted cookie from the client and send to the server [11].

The server authenticates the hacker and allows accessing the user resource. Gmail is a widely used email account which uses cookie management in client. Figure 4 shows the captured cookie information of Gmail Account from the client. Tools like Tamper Data are used to analyze the cookie and other packet information.



http://mail.google.com/mail/

Request Header Name	Request Header Value
Host	mail.google.com
User-Agent	Mozilla/5.0 (Windows NT 6.1; rv:27.0)
Accept	text/html,application/xhtml+xml,app
Accept-Language	en-US,en;q=0.5
Accept-Encoding	gzip, deflate
Cookie	GMAIL_IMP=v*2%2Ftl-i*0!inbox%2Fp

Fig. 4. Gmail cookie hijacking using tamper data.

3.2.2. Tamper Data

Tamper Data is an add-on for browser is a fast, efficient tool which can penetrate the network and captures the cookie information. Encrypted cookie are captured and used to hack the user who uses cookie for login credentials [12].

4. Session-Packet Inspector Model Architecture

Figure 5 shows the Architecture involves in three major blocks. Client block, Server block and MA processor block. Each block has typical behaviour and acts on the packets between the client and the server.

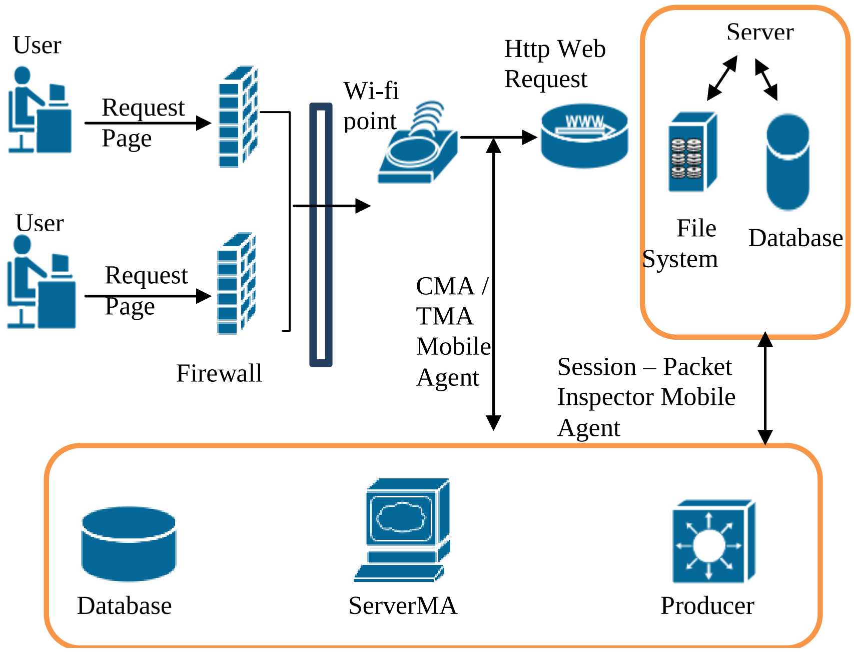


Fig. 5. Session - packet inspector model architecture.

4.1. Client block

It has client access components that interact with the server using mobile agent. This block is the initial phase and requests the server for authentication. Components that are involved in this block are browser, Mobile Agent Generator, Mobile Agent Handler and Firewall.

4.2. Server block

Server block are responsible for the analysing of the client request using mobile agent. The request is analysed and process is sent to the Mobile Agent Handler. Components used to build the server block are Firewall, Mobile Agent Analyser.

4.3. MA processor block

This block process the request from the server and process the web pages for the client. This block decides with user credential and accesses the web pages for the client. Components used are mobile agent processor, database and web store.

5. Session-Packet Inspector Model

5.1. Key generation

Session – Packet Inspector Mobile agent (SPIMA) is responsible for producing the session key to encrypt the cookie. The session key is generated at runtime without transferring it. Since the key is not transferred the rate of attacking nature is minimized. For each session the key is generated uniquely for encryption and decryption.

SPIMA connects with the server to store the number of key produced. It refers the centralized server database for key generation. The record holds the key generated information as shown in Table 1. Each Session – Packet Inspector mobile agent has unique ID and is entered into the record with encryption and decryption key status with timestamp.

Table 1. Key generation information.

Session – Packet Inspector Mobile Agent ID	**Encryption Key Generated Status**	**Timestamp for Encryption Key**	**Decryption Key Generated Status**	**Timestamp for Decryption Key**

Four parameters as described in Fig. 6 are used to create a session key. RequestURL refers to the URL that has been requested including query string. Server public key is used in client side where as Server private key is used on server side. SPIMA unique key is generated by the mobile agent. TCM server generated one time session key for the transaction. During Encryption process, the parameters are used to generate a checksum named as Session Key. We used RIPEMD-320 algorithm to generate the session key.

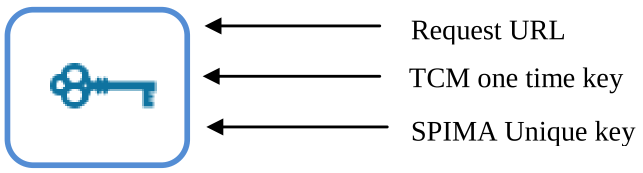


Fig. 6. Session key generation using RIPEMD-320 algorithm.

5.2. Session – Packet inspector model algorithm

```
(1)  GetResponse(string loginUrl)
(2)  {
(3)  Byte[] cookie= getCookie(GetDomain(loginUrl));
(4)  MobileAgentmobileAgent= GetSPMobileAgent();
(5)  mobileAgent.RegisterUser();
(6)  Byte[] Session_KEY=GenerateKey();
(7)  If(cookie==null)
(8)  {
(9)  HTMLDocument document =
     mobileAgent.feedPage(getAuthenticationPage(url));
(10) mobileAgent.decrypt(Session_KEY);
(11) mobileAgent.DisplayInBrowser(document);
(12) mobileAgent.GetAuthenticationParams(UserName,Password);
(13) }
(14) else
(15) {
(16)EncryptedCookieencrytCookie=mobileAgent.Encrypt
(cookie,Session_KEY);
(17) mobileAgent. GetAuthenticationParams (encrytCookie);
(18) }
(19) mobileAgent.encrypt(Session_KEY);
(20) mobileAgent.AddCredentialTicket();
(21) mobileAgent.Request();
(22) if(mobileAgent.Authenticated)
(23) {
(24) while(true)
(25) {
(26) If(mobileAgent.IsRequestNewPage())
(27) {(28) HTMLDocument document= mobileAgent.getPage(url);
(29) If(mobileAgent.CanSetCookie)
(30) {
(31) mobileAgent.SetCookieInBrowser();
(32) }
(33) mobileAgent.decrypt(Session_KEY);
(34) mobileAgent.DisplayInBrowser(document);
(35) mobileAgent. SetParams(params);
(36) }
(37)  }
(38)  }
```

5.3. llustration

Client block initiates the login session to the server. GetResponse method in described algorithm is executed by the client with the login url link. Initially the cookie is retrieved from the browser. Browser has unique key-value pair for the domain and cookie to retrieve the cookie. If user allows storing login cookie for the domain, it will be retrieved. Session key for that transaction is generated. This session key is used throughout the transaction. Mobile Agent Generator component creates a mobile agent by duplicating and initializing SPIMA. We checks for the available login credential cookie in the browser. If not exist, the mobile agent gets the authentication page from the server. The retrieved login page is decrypted with the session key and displays in the browser. When the user enters the credential, SPIMA stores the encrypted credential information. If cookie exists, SPIMA encrypts the cookie with session key and retrieves the authentication ticket from the cookie. SPIMA encrypts the content with the session key and make a request to the server.

Server block get initiated when a request is triggered by the client. SPIMA carries the request from the user and credential ticket. Mobile Agent Analyser analyses the SPIMA generates the session key and checks the credential ticket. Credential ticket is used to authenticate the mobile agent. If it is defined properly, it forwards the SPIMA to the MA processor block.

SPIMA is processed in the component called Mobile Agent Processor. Session key is generated and the decryption process is carried out. The requested login credentials are used to authenticate the user. When the authentication process succeeds, it stores the request into the database. The webpage is processed for the user and transferred to the Mobile Agent Handler. It encrypts the web page with the session key and sends back to the user.

Initially when a user makes a request to the web server, the requested is transmitted to the TCM server. TCM server generates Session-Packet inspector mobile agent and send to the requested user. Mobile agent register the user to the TCM and receives the one-time TCM key. Session key is generated by mobile agent using TCM one-time key, SPIMA unique key and requested URL. MA checks for the cookie in the for requested website login. If the cookie is not found, MA requests the authentication page from the web server.

TCM stores Sessionkey generated for the user requested session.MA decrypts the webpage and display in the browser. When user enters the login credential, MA generates an encrypted cookie for the website. If cookie is identified for the user requested website, MA encrypts the cookie with the Session Key and sends to the web server. MA decrypts cookie in the web server and connects with the TCM. TCM verifies the user, MA encrypt the user page and transmit to the TCM server. TCM server identifies the client that requested initially and send the packets to the client.

6. Experimental Results and Output

The proposed technique is stimulated in Ns2 stimulator and the configured values are tabulated in Table 2.

AODV algorithm is used for transmission of packets in MANET. CBR traffic are increased to identify the performance of the given algorithm in worst case. Minimum size of the cookies are set as 2kb to enable the start-up size of the process in cookie transmission13.

AODV algorithm is used for transmission of packets in MANET. CBR traffic are increased to identify the performance of the given algorithm in worst case. Minimum size of the cookies are set as 2kb to enable the start-up size of the process in cookie transmission. Figure 7 describes the stimulation result of cluster with user, TCM server and Web server. Page transmission is based on cookie credential. A group of 87 nodes are created as a cluster with users, TCM servers and web servers. Figure 8 indicates the header transmitted from node 1 to node 3.

Table 3, 4 and 5 explains comparative report over the number of packets transmitted through the network in given time span, compromised node details and packet transmission. It illustrates the behaviour of the other parameter that affects the quality of services. Packet transmitted gradually decreases due to parameter such as bandwidth and traffic [13].

Figure 8 describes the rate of packets transmitted by the Mobile agent for given time. 24 nodes are stimulated as hacking user and accessed the cookie file transmission. Web server accepts the cookie but the file is transmitted to the requested original user that have been registered. Session-Packet inspector mobile agent migrates from the server and client to handle the requested web pages. Session key generated using RIPEMD 320 algorithm are generated in 223 milliseconds for each transition. This prevents cookie hijacking and HTTP header hijacking. HTTP header are considered as packets that are transmitted during the transmission of actual data packets. SPIMA encodes the packets and uses default compression technique for transmission.

Table 2. Stimulation initialization details.

Parameter	Values
Stimulator Time	1450s
Number of Nodes	87 nodes with one server and one intruder
Routing Protocol	AODV
Traffic Type	CBR
Channel	Wireless Channel
Energy Model	Energy Model
Minimum Initial Energy	5.23 joules
Interface Queue Type	DropTail
Cookie Minimum Size	2Kb

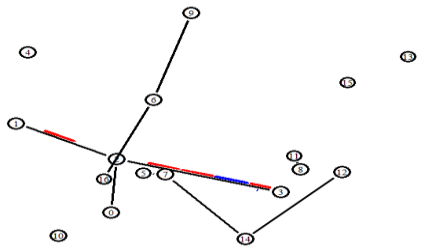


Fig. 7. Simulation result.

Figure 9 visualizes the effectiveness of the proposed model by comparing the stimulation report generated. The intrusion over the network by hijacking the cookies are effectively handled by the proposed model. Intruded nodes in proposed model are hacking through since the nodes are infected with the Trojan virus. Secure nodes are handled to avoid intrusion through cookie access. Session hijacking attempts are made continuously over the designed proposed model and existing model. The experimental result explains that the compromised nodes are minimized about 78% from the existing model.

Table 3. Mobile agent transmitted in given time.

Time Taken (ms)	Mobile agent Transmitted(count)
10	53
20	60
50	98
80	135
100	144
130	199
180	215
200	221
230	232
250	244

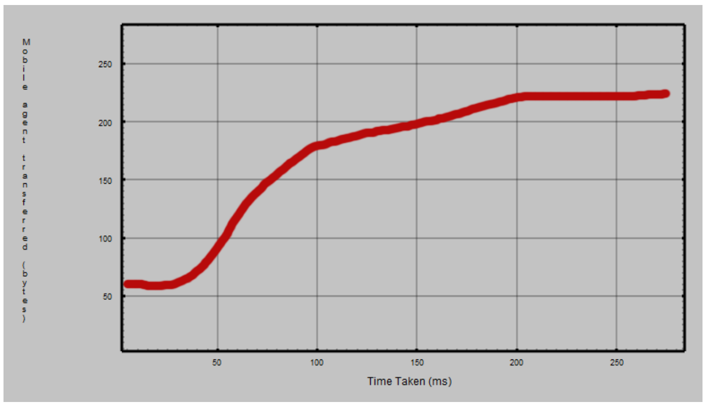


Fig. 8. Mobile agent packet transmitted vs. time taken in X graph.

The detailed captured are graphed from the stimulator in Fig. 10 describes the comparative report for the effect of attack over packet transmission. The following table describes the comparative study of the behaviour existed for stimulated report over existing and proposed model. Table 3 tabulates the mobile agent transmitted in given time. Detailed comparative values are tabulated below in Table 4 and 5 reveals that the proposed model covers best intrusion detection technique over existing model.

Table 4. Nodes affected count in given time

Time Taken (ms)	Existing Model(count)	Proposed Model(count)
10	6	2
20	12	4
50	23	8
80	52	11
100	74	11
110	80	14
120	89	15
130	91	17
135	102	17
140	110	19

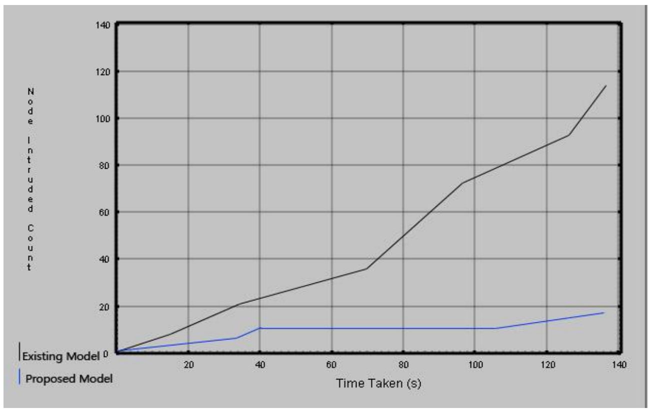


Fig. 9. Intrusion comparison graph.

Table 5. Compromised node count for hijacking attempts

Session Hijack Attempts	Existing Model(count)	Proposed Model(count)
10	3	3
20	4	3
50	14	6
80	34	8
100	38	8
130	53	8
180	63	11
200	83	14
230	89	16
250	93	19

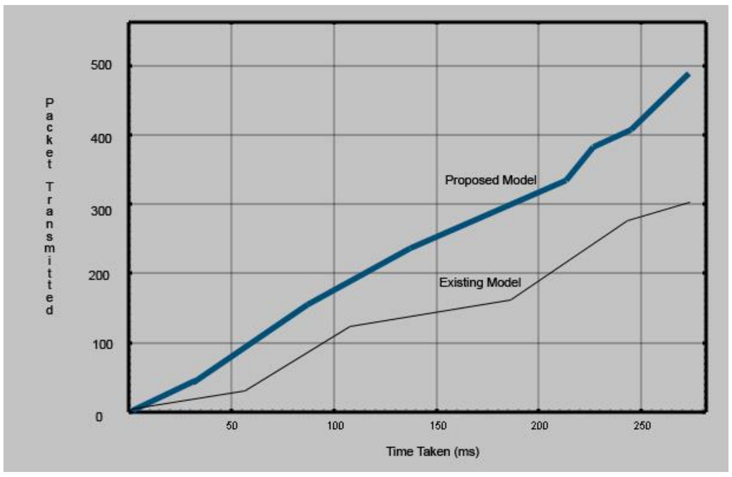


Fig. 10. Comparative report for compromised nodes vs. session hijack Attempts.

7. Conclusion

In this paper, we have proposed Session-Packet inspector Mobile agent that prevents the HTTP post hijacking and encrypted cookie hijacking by the intruder in open wireless network. The TCM is an effective model to prevent the malicious mobile agent to penetrate into the hosts3. TCM server handles the mobile agent generation and web page transmission from web server to client.MA handles the credential transmission from web server to the user that prevents the credential HTTP header hacking. Since TCM handles the transmission between user and web server, cookie hijacking by hacker may result in transmission of pages to the user. Proposed model rectifies the attacks and prevents intrusion in the ad-hoc network. Session-packet inspector mobile agent prevents the intrusion and along with TCM model it triggers alarm to other host to initiate prevention measures.

Key generation for Session-Package Inspector handles the intrusion system effectively with TCM model. Session hijacking using Tamperdata and Firesheep are prevented with the proposed model. Comparison result with network analysis of packet sniffing is explained in the result session which demonstrate the effectiveness of the proposed model using mobile agent.

Our future work may enhance the web server page transmission handling to prevent the double transmission between user, web server and TCM.

References

1. Putthacharoen, R and Bunyatnoparat, P (2011). Protecting cookies from Cross Site Script attacks using Dynamic Cookies Rewriting technique. *International Conference on Advanced Communication Technology (ICACT)*, 1090-1094.

2. Ping, Yi; Xinghao, Jiang; Yue, Wu; and Ning, Liu (2005). Distributed intrusion detection for mobile ad hoc networks. *Symposium on Application and the internet Workshops*.
3. Xia Chun-Tao; u Xue-Hui; Cao Li-Feng; and Chen Hua-Cheng (2012). An Algorithm of Detecting and Defending CC Attack in Real Time. *International Conference on Industrial Control and Electronics Engineering (ICICEE)*, 1804-1806.
4. Arun, V; and Shunmuganathan, K.L. (2011). Tracer and Checker Model against Malicious Mobile Agent. *National Conference on Wireless Information and Networking in Global Systems*.
5. VratonjicN; Kudelski Security; Ceseaux-sur-Lausann -e Switzerland; Huguenin, K; Bindschaedler, V; Hubaux, J-P (2014). A Location-Privacy Threat Stemming from the Use of Shared Public IP Addresses. *IEEE Computer Society on Mobile Computing*, 2445-2447.
6. YiXie Tang, S; Xiang, Y; and Hu J (2013). Resisting Web Proxy-Based HTTP Attacks by Temporal and Spatial Locality Behavior. *IEEE Parallel and Distributed Systems*, Vol. 24, Issue 7, 1401-1410.
7. Jin Wang; Min Zhang; Xiaolong Yang; and Keping Long (2013). HTTP-sCAN: Detecting HTTP-flooding attaCk by modeling multi-features of web browsing behavior from noisy dataset. *Asia-Pacific Conference on Communication*, 677-682.
8. YueLi; Meng Zhou; and Dehua Chen (2013). Automatically identify the website's key-Cookies for merging sessions. *Ninth International Conference on Natural Computation*, 782-786.
9. Alabrah, A; and Bassiouni, M (2013). A hierarchical two-tier one-way hash chain protocol for secure internet transactions. *IEEE Global Communications Conference (GLOBECOM)*, 868-873.
10. Chomsiri, T (2008). Sniffing Packets on LAN without ARP Spoofing. *International Conference on Convergence and Hybrid Information Technology*, Vol. 2, 472-477.
11. Shakshuki, E.M; Jodrey Wolfville, NS; Nan Kang; and Sheltami (2012). EAACK—A Secure Intrusion-Detection System for MANETs. *IEEE Transactions on IEEE Industrial Electronics Society Journal*, Vol. 60, Issue 3, 1089-1098.
12. Pauli, J.J; Engebretson, P.H; Ham, M.J; Zautke, M.J (2011). CookieMonster: Automated Session Hijacking Archival and Analysis. *Eighth International Conference on IEEE transaction Information Technology: New Generations (ITNG)*, 403-407.
13. Mohammed Abdul Qadeer; Arshad Iqbal; Mohammad Zahid; and Misbahur Rahman Siddiqui (2010). Network Traffic Analysis and Intrusion Detection using Packet Sniffer. *Second International Conference on Communication Software and Networks.*313-317.