Journal of Theoretical and Applied Information Technology
20th April 2014. Vol. 62 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

PROTECTION OF FREE ROAMING MOBILE AGENTS USING CUSTOMIZED ROOT CANAL ALGORITHM AGAINST MALICIOUS HOST ATTACKS

[1]GEETHA G [2] JAYAKUMAR C
[1] Anna University, Chennai- India, Department of CSE, Jerusalem College of Engineering
[2], Anna University, Chennai- India, Department of CSE, RMK Engineering
E-mail: [1]togeethamohan@gmail.com, [2]cjayakumar@gmail.com

ABSTRACT

Mobile agent plays an important role in developing applications of open, distributed and mixed environments, such as the internet. As an agent travels do execution in different environment in different host or servers, the agent are in need of protection themselves and their data from various types of attacks. Providing security to the mobile agent (static code) and it data (dynamic code) is emergence need in Mobile agent Technology. The main issue of free roaming mobile agent in data collection is colluded truncation attack. In this paper, the Customized Root Canal (CRC) Algorithm is used to protect the code and data of the agent through code and data integrity. To overcome colluded truncation attack CRC algorithm immediately return collected data to the originator
Keywords: *Mobile Agents, Free Roaming, Protection, Attacks, Data Security, Code Integrity, Data Integrity*

1. INTRODUCTION

This An agent is an entity which will do something on behalf of another user to achieve the specified task. Mobile Agents are software programs that are capable of migrating from one host to other host. Mobile agents can block the execution in one host and resume its execution in some other host. Mobile agents decide autonomously when to migrate and where to migrate. When a mobile agent decides to migrate, it saves its own state and transports this saved state to next host and resume execution from the saved state on the remote host.

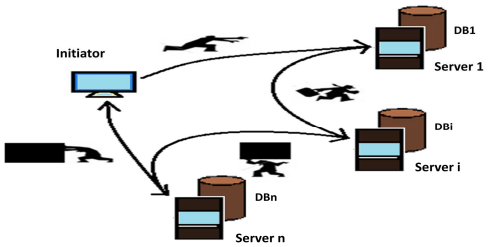

Fig. 1 Mobile Agents Communication

A mobile agent consists of three components: Code (program that defines the agent's behaviour), State (the agent's internal variables which allow it to resume its actions after moving to another host), Attributes (information about its origin, owner, its movement history, resource requirements, and authentication keys). Agent can access the attributes but it cannot modify them.

A_{code}	A_{state}	I_a	d_0

Fig. 2 Components Of Mobile Agents

Each mobile agent has static and dynamic parts. Static consist of the agent code. Dynamic part will be changing during execution of the agent. The Dynamic part consists of data which can be either the one collected or the intermediate results of the computation. This data is cryptographically protected and transferred with the agent from host to host.

There are many types of mobile agents and they are: Single hop mobile agent (This type of agent will be visiting only one remote host and return back to origin). Multi hop mobile agent with Static Itinerary and Static Order (This type of agent will be visiting multiple remote host and go back to the owner with the necessary result. It will visit the remote host based on the itinerary and order given by the owner). Multi-hop mobile agent with Static Itinerary and Dynamic Order (This type of agent will visit the many number of remote hosts based on the itinerary given by the owner but the order is based on the run time decision of the remote host

Journal of Theoretical and Applied Information Technology
20th April 2014. Vol. 62 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

where the agent resides at present. Multi-hop mobile agent with Dynamic Itinerary and Dynamic Order (This type of agent will visit the remote host based on the run time decision without the owner information. Here the owner does not know the particulars about the remote hosts apart from the first remote host Dynamic Itinerary will always follow the Dynamic Order.

There are several benefits in mobile agents when compared to other traditional computational methods. In Traditional methods such as the client server the data is moved to the location where the code is available. But in case of mobile agents the code will be moved to the location where the data is offered. Thus it reduces network load which in turn reduces the network latency by avoiding unnecessary delays. Mobile agents have understanding with their execution environment and react freely to changes. So they are tolerant to network faults. Mobile agents provide flexibility in maintaining an agent.

Normally security issues in mobile agents are classified as agent security and host security. Agent security is further divided into code security and data security. In data security, the method used to protect data in mobile agents depends on type of movement. The migration path of agent can be pre defined itinerary or free roaming In case of predefined itinerary, the path of migration will be specified earlier and it is static. The owner of the agent will be choosing the path. Free Roaming Mobile Agents are those which will decide the next host dynamically. Security in free roaming agents is especially hard to achieve when the mobile code is executed in hosts that may behave maliciously.

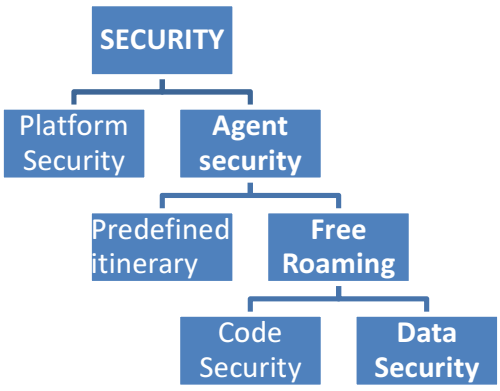

Fig. 3 Types Of Mobile Agents Security

In another aspect agent security is classified into two types and they are: Protecting agents against malicious hosts and protecting agents from other malicious agents. A malicious host can attack the agent, by modifying its data, corrupting or altering its code or state, deny requested services, return false system call values, reinitialize the agent or terminate it completely. It can also masquerade the agent by delaying the agent until the task is no more relevant. A malicious agent may call public methods of another agent to interfere with its work

There are two types of host security and they are protecting a host against malicious agents and protecting a host against other malicious hosts. In case of host security, malicious agents can steal or modify the data on the host. Lack of adequate authentication and access control mechanisms lead to this type of attack. A malicious host will attack the other host available in the mobile agent environment.

An attack can be called as an abuse of expectations of the agent owner caused by one or more than one planned attacker. The attack can be divided into two and they are passive attack and active attack. In passive attacks, an opponent attempts to extract some information from messages exchanged between two communicating parties without modifying the contents of the messages. This attack is known as eavesdropping. In active attacks the opponent will modify the data or the code of a mobile agent. An adversary may impersonate a legitimate user in the system and intercept messages intended for that user.

2. RELATED WORKS

In Reference [4], the self-protected mobile agent scheme proposed by Pedro et al., 2006 will allow the sender to sign the agent code using the disposable key. This sign is provided in order to ensure the integrity of the code. The agent code and its signature are inserted into a digital envelope. The envelope is then encrypted with a symmetric key. This key is the only part of the agent that will be decrypted with the next user's public key at every host. To make the mechanism stronger, the system generates a disposable key pair to sign the agent code and to verify its integrity. On receipt of the agent at the remote platform, it uses its secret key to extract the encrypted disposable key pair and the signature of the agent code. The operation will be successful if the code of the agent has not been modified. The drawback of this model is, multiple

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

numbers of keys and their signature are used with multiple computations.

In Reference [8] Simple Malicious Identification Police was proposed to detect the presence of the malicious code. An agent migrating from one host to another host will become malicious agent if there is any malicious host available in its path or it can be from the malicious originator. The main intention of the malicious agent is to put the remote host in trouble. This Simple MIP will protect against this type of attack. The Simple MIP uses an attack identification scanner (AIS) to scan the agent code in order to detect the presence of any malicious code. The MIP can be compared to police men and the AIS can be compared to a scanning device. The Police men will have the scanner or detector in his hand to scan the people before allowing them into a sensitive place. In this simple MIP model, the agent's byte code is scanned by the AIS to detect the presence of any malicious code in the given agent byte code. If the AIS find any malicious codes within the agent code, then the agent is either discarded or killed by the agent platform. If the agent is legitimate then it will be allowed to carry out its computation. On completion of the necessary computation, the agent will be dispatched to the next remote host or to its home as per its itinerary.

Usually, the agent from the owner or from the intermediate hosts generates the byte code. The agent byte code with an additional malicious code is considered as the malicious agent. The AIS will be scanning the byte code and compares it with the available malicious codes stored in the server. The Malicious code are prohibited codes such as shutting down the platform, accessing a restricted database, killing an agent, cloning many number agents. A malicious agent migrated from one machine to another will carry these types of malicious codes. The MIP with the help of the AIS scanner will scan the agent byte code to detect those malicious lines in the agent program.

In Reference [7], Policy based MIP model was proposed in order to give privileges to the remote host in the network to avoid equal consideration. This Policy based MIP first decrypt the encrypted identity of the agent owner with public key of the agent owner. Then it verifies the resemblance of the identity to avoid masquerading and also it checks for the privileges. After verification MIP initiates the AIS to scan the agent byte code based upon the privileges, AIS then reports the presence of the malicious code, if any, to the MIP. At last MIP decides whether to discard the agent in case of the availability of the malicious code or to allow the agent to execute in case of a legitimate host.

In Reference [9], the Root Canal (RC) algorithm was proposed in which the identification method is done using the hash function. The Agent can be attacked by modifying or deleting the code. Remote host receiving the mobile agent will have no rights to alter the agent code during execution. If there is any modification then the objective of the agent will be changed completely. This can also be considered as an attack. To protect against this type attack the malicious host is recognized and removed from the actual environment. The hash function is applied to the byte code of the agent to get the hash code. The hash code is encrypted and bundled with the agent. On Migration to the another remote host the agent will be having static part of code, dynamic part of the state, the itinerary part, the registry, resource to the agent, and protocol to handle the agent. Remote host that receives the agent, should first take the encrypted hash code and decrypt it and produce the plaintext. Then it should generate the hash code for the agent it receives and compares both the generated hash code and the received hash code. If there is a match between these two codes the agent has been migrated successfully without any attacks.

The RC algorithm does not protect against the false malicious claim. The remote host receiving the mobile agent can have the rights to argue for malicious alteration of the agent code. As the following host identifies the malicious changes in the agent code received from the earlier host, it has the right to claim for the malicious change against the preceding host. In a few cases, the malicious host can claim for the maliciousness against the legal preceding host.

To overcome this Extended Root Canal (XRC) algorithm in Reference [7] uses the concept of digital signature. This Digital Signature concept is used mainly for the purpose of non-repudiation. In case of non repudiation the sender of the data is cannot claim that the data are not mine. The malicious host is unable to claim for a change in the agent code without having the entire proof. The signature on the encrypted hash code by the previous host is considered as the proof. This XRC algorithm was implemented with the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

assumption that the originator is a legitimate host and the agent created at this host

is also a genuine host. So there is no chance of maliciousness the host as well as the agent. This in turn avoids the need for providing digital signature at the originator and also avoids the claim for maliciousness from the first remote host.

In reference[10] propose a new mechanism "Address Forward And Data Backward (AFDB)" to protect the agent's data. In which data was encrypted into a divisible whole for protection. When Task Agent reaching a host, the data and the host identity information is collected will be sent back to the source host (Secondary Agent (SA)) immediately along with encryption and Signature. Then the TA concatenates the Address of the host with the Address collection carried by agent then encrypts the total message and generates hash code for the entire message. Both the message and the hash are attached with the Agent. When agent comes back to source, the SA will compare the two paths (SA, TA) through decryption and signature verification to find out if there attack is exist or not and then sends corresponding data.

Silei [11] proposed method use integrity measurement feature and the integrity reporting feature. In this Integrity measurement is the process of obtaining metrics of platform characteristics where as integrity reporting is the process of attesting to integrity. But this mechanism has two agents, task agent, and secondary agent platform configuration register.

Linna [1] proposed the method signature trust chain mechanism. In STCM data was encrypted in to a whole for protection and sending identity information to trusted third party to resist attack. This mechanism use TTP for Verification.

Yee [12] proposed the method of Partial Results Authentication code (PRAC) in that the results (data) of an agent is encapsulated using MAC in each host in which an agent perform its computations. The Agent data Combined with Message authentication Code (MAC) is called PRAC. This Method is in need of secret Key for each Host.

Roth's [13] proposed method, the agents transfer commitments to other Co-operating Agents, Those agents performs task like storing gathering and verifying But idea behind this approach is TTP.

3. PROPOSED METHOD

Customized Root Canal Algorithm we are using the concept of hash generation, public key encryption and digital signature. To generate the hash value Secure Hash Algorithm - 1 (SHA-1) is used. For the purpose of encryption and decryption in public key cryptography we are using the Rivest Shamir Adelman (RSA) algorithm. Use of RSA algorithm provides confidentiality and authentication. The encapsulated data cannot be truncated in between due to immediate return of data.

The CRC algorithm provides protection to the code and the data collected in each host. The data collected by the agent in each host is sent immediately to the originator. There is a chance for the data to be hacked by the intruders while it is carried by the agent. In this paper, we are proposing the Customized root canal algorithm to provide protection to the code and data that collected by the agent.

3.1 Security Requirements

The Mobile agents have many Security requirements. Few of them are the following requirements:

1) Data Confidentiality: The Code and data carried by the agent must be migrated confidentially. The Location of Mobile agent must also be kept confidential.

2) Integrity: The Mobile Agents must be protected against unauthorized alteration of the data, code and other attributes of the agent. The Agent platform must also be protected against unauthorized access.

3) Authentication: The Identity of the agent must be authenticated to gain the required access rights. Audit log maintains information about the users and their corresponding access rights.

4) Availability: The availability of the services to the agents and the availability of data must be guaranteed by the agent platform.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5) Truncation Resilience : The encapsulated offer can not be broken in between due to immediate return of data .

Out of these requirements, our proposed algorithm concentrates mainly on the integrity part especially code integrity and data integrity. It also provides authentication and confidentiality by using the cryptographic algorithms.

1) Agent at creator S0

i. Hcode= H(Agent Byte Code)
ii. EHcode = EPR0 (Hcode)
iii. S0 →Si : EHcode

2) Agent at Remote host Si

i. RHcode= DPUi-1(EHcode)
ii. Hcode= H(Agent Byte Code)
iii. If (RHcode== Hcode) then
a) Collect Di
b) rdi= EPU0 (SigPRi(Di)||Di||Si|| Si+1)
c) RDi=rdi|| H(rdi)
d) EHcode = EPRi (Hcode)
e) Si →Si+1 : EHcode
f) Si → S0 : RD
Else
Si → S0 : Msg (Error code)

3) Agent at Remote host Sn

i. RHcode= DPUn-1(EHcode)
ii. Hcode= H(Agent Byte Code)
iii. If (RHcode== Hcode) then
a) Collect Dn
b) rdi= EPU0 (SigPRn(Dn)||Dn||Sn|| S0)
c) RDn=rdn|| H(rdn)
d) EHcode = EPRn (Hcode)
e) Sn →S0 : EHcode
f) Sn → S0 : RD
Else
Sn → S0 : Msg (Error code)

4) Agent at S0
For each RDi
i. rdi|| RH(rdi) =RDi
ii. if H(rdi) = RH(rdi)) then
a) SigPRn(Dn)||Dn||Sn|| S0=DPR0(rdi)
b) if (DPUi(SigPRn(Dn))==Dn)
Process Di
Else
S0 →Si : msg(Error Data)

Fig. 3 Mobile Agents data collection using CRC

3.2 Description Of The Algorithm

The agent is created at the originator and then the byte code is generated for the agent's code part. The Hash code (Hcode) is generated with the offer of the agent byte code. The hash code is encrypted using the private key of the originator to get EHcode. Then the agent is dispatched to the next remote host by providing the digital signature using the private key of the originator.

At the first (i.e. at the Si,i=1) remote host, on reception of the agent, the current host will verify the signature and then generate hash value for the received byte code (Hcode). Then the Received EHcode is decrypted using the public key of the previous host .

Decrypted hash code is compared with the hash code generated at this host. If both the codes match exactly, then it indicates that the code has been migrated without any attacks. If there is any mismatch between these codes, then it indicates the presence of malicious code. If so, this host can claim for its maliciousness. After performing this checking process the agent collect the data available at this host. Then it is encrypted using public key of the originator. signed data, data, current host id, next cost id and the hash value of the these are concatenated. Concatenated Data is immediately sent to Originator. Now the agent code is also provided with the digital signature using the private key for migration of the next host.

Originator also verifies the signature of the last host, it also checks for the integrity of the code. Then it decrypts the data collected at all the host and perform any computation. Only the originator can decrypt the data collected at each host because all the data has been encrypted using the public key of the originator. The encapsulated data cannot be truncated in between due to immediate return of data.

3.3 Attacks And Security Anaylsis

This algorithm mainly concentrates on the modification attack. It checks whether the data and code have been modified by any attackers during the migration. Modification ensures the presence of the malicious code. In that case, the agent is discarded otherwise it is performed to do the desired work such as collecting the data at the specified host. Since the code and data has been

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

transmitted immediately in the encrypted format and as the hash value, attackers cannot easily retrieve the original Each and every host dispatches the agent after providing the digital signature using its private key. This signature provides authentication and in turn it avoids the need for the false malicious claim concept. Thus our proposed algorithm satisfies the authentication confidentiality and truncation reliance requirements.

4. CONCLUSION

In this paper, we presented the Customised Root canal algorithm to protect the code and data of free roaming mobile agents. The algorithm also provides protection against active attack such as modification attack. This algorithm also ensures the integrity of the code and data of the mobile agent. At each and every host, the code is checked for the availability of the malicious code. If the malicious code is available then the agent is discarded. Colluded truncation attack of free roaming mobile agent in data collection is avoided through immediate data return to the originator.

REFRENCES:

[1] Fan Linna, Liu Jun. A Free-Roaming Mobile Agent Security Protocol against Colluded Truncation Attack. In the proceedings of 2nd International Conference on Education Technology and Computer (ICETC) 2010.

[2] Huanmei Guan, Huanguo Zhang, Ping Chen and Yajie Zhou. A Protecting protocol of Mobile Agent Integrity. International Conference on Computational Intelligence and Security Workshops, 2007.

[3] Lu Ma and Jeffrey J. P. Tsai. Formal Modeling and Analysis of a Secure Mobile-Agent System. IEEE Transactions on systems, man and cybernetics 2008.

[4] Pedro MV, Cruz-Correia Ricardo Joao, Robles Sergi, Cucurull Jordi, Navarro Guillermo, Martı´ Ramon. Secure integration of distributed medical data using mobile agents. Proceedings of IEEE Intelligent Systems 2006 p:47-54

[5] Qi Zhang, Yi Mu, Minjie Zhang, and Robert H. Deng. Secure Mobile Agents with Designated Hosts. IEEE Third International Conference on Network and System Security 2009.

[6] Raji F. and B. Tork Ladani. Anonymity and security for autonomous mobile agents. IET Information Security, 2010

[7] Venkatesan S, Chellapan C,Vengattaraman T, Dhavachelvan P and Anurika Vaish. Advanced Mobile Agent Security Model For Code Integrity and Malicious Availability Check. International journal of Network and Computer Application. Elsevier 2010.

[8] Venkatesan S. and Chellappan C. Protection of mobile agent platform through attack identification scanner (AIS) by malicious identification police (MIP). In: Proceedings of the international conference in emerging trends in engineering and technology, (ICETET'2008), 2008, p. 1228–1231.

[9] Venkatesan S., Chellappan C. Identifying the split personality of the malicious host in the mobile agent environment. In: Proceedings of 2008 IEEE international conference on intelligent systems, 2008a, p. 14–40 to 14–44.