Liu, Xing-peng ................................... TT-SCT3-2
Liu, Yabin ..................................... SS-IPSC2-1
Liu, Yanpeng .................................. TT-SCT6-3
Liu, Yao ........................................ SS-LIE-6
Liu, Yubo ....................................... SS-LIE-5
Liu, Yi-Hua ..................................... TT-SCT2-5
Lo, Jung-Hua .................................... TT-ISST2-4
Lobov, Andrei .................................. TT-SOA2-4
Lobov, Andrei .................................. TT-DENC4-2
Low, Joyce M.W. ............................... TT-BNA1-1
Low, Joyce M.W. ............................... TT-SCML1-6
Low, Malcolm Yoke Hean .................... TT-CCI1-3
Low, Malcolm Yoke Hean .................... TT-SCML3-1
Lu, Xu ........................................... TT-IIA4-4
Luis, Zarate ................................... TT-EPT1-5
Luo, Zhimeng .................................. TT-DENC1-2
Lv, Tao ......................................... SS-TCL-5
Lye, Kong-wei .................................. TT-SCML1-4

M
Ma, Bin ........................................ TT-SCML1-1
Ma, Bin ........................................ TT-SCML3-4
Ma, Tianyun ................................... SS-LIE-7
Ma, Zi .......................................... TT-IIA8-3
Machado, Vinicius ............................. TT-FA1-2
Madani, Sajjad ................................. TT-BNA2-2
Mahlknecht, Stefan ........................... TT-BNA2-2
Malik, Najmus Saqib ......................... TT-ISST3-1
Maria, Ana ..................................... TT-IIA2-1
Martel, Allan .................................. SS-IEC-2
Matsumoto, Shinichi .......................... SS-PAC-3
Matsushiba, Yoshinao ....................... SS-ITE-2
McFarlane, Duncan ......................... TT-DENC4-3
Medeiros, Juliana ............................. TT-FA1-2
Mendes, J. Marco ............................ TT-SCML1-6
Mendes, J. Marco ............................ TT-FA1-2
Meng, Fanyi .................................. SS-RF1-5
Merdan, Munir ................................. TT-CCI3-5
Min, Huasong ................................ TT-DENC2-6
Min, Seung Hwan ............................. TT-BNA2-5
Min, Zhang ................................... TT-IIA6-1
Mitsch, Stefan ................................. TT-EPT1-4
Miyazawa, Kazunori ......................... TT-ISST2-1
Mohamed, Shady Mohamed Korany ....... TT-SCT3-5
Monfared, Radmehr ......................... TT-FA2-4
Moon, Tae Yoon ............................... TT-DENC2-2
Moon, Tae-Yoon .............................. TT-DENC3-3
Moon, Tae-Youn .............................. TT-DENC2-1
Moon, Tae-Youn .............................. TT-IIA3-5
Moon, Tae-Youn .............................. TT-BNA1-5
Moon, Yongseon ............................. SS-IR1-4
Morais, Antônio .............................. TT-SCML1-3
Morais, Antônio ............................. TT-IIA2-1
Morimoto, Sigeaki ......................... TT-SCT6-2
Mous, K. ..................................... TT-SCML1-3
Mrazova, Iveta ............................... SS-PAC1-4
Mvungi, Nerey ............................... TT-MR2-1

N
Nakamura, M. ................................ TT-IIA8-4
Nagashima, Akira ............................ TT-DENC1-4
Nahavandi, Saeid ............................ SS-IPSC1-1
Nahavandi, Saeid ............................ TT-SCT3-5
Neto, Lauro V.B.M. ......................... TT-IIA5-5
Ng, Vincent ................................ TT-IIA6-4
Ng, Wee Keong .............................. TT-ENIE2-2
Kim, Jeong S. ................................ TT-DENC1-1
Kim, Jin Ho .................................. TT-DENC2-2
Kim, Jin-Ho .................................. TT-BNA1-5
Kim, Jin-Ho .................................. TT-DENC2-1
Kim, Jin-Ho .................................. TT-DENC3-3
Kim, Jin-Ho .................................. TT-IIA3-5
Kim, Ki Hoon ................................ TT-SCT3-3
Kim, Seung-Han ................................ TT-BNA2-5
Kim, Seung-Han ................................ TT-DENC3-3
Kim, Seung-Han ................................ TT-IIA3-5
Kim, Sung Hyun ................................ SS-LIE-1
Kim, Sung-Jo ................................ TT-MR1-3
Kim, Taeho .................................... TT-HCM3-3
Kim, Woo-Je .................................. TT-SCML1-2
Kirkham, Tom ................................ TT-FA2-4
Kitoko, LeBel .................................. SS-DES-3
Klein, Lothar .................................. SS-DES-3
Ko, Nak Yong .................................. SS-IR1-4
Koo, Benjamin H.-Y. .......................... SS-LIE-4
Koppensteiner, Gottfried ..................... TT-CCI4-5
Kuikka, Seppo ................................ TT-SOA1-3
Kukacka, Marek ................................ SS-PCCT1-4
Kupzog, Friederich ............................ SS-ITE-4
Kupzog, Friederich ............................ TT-MR1-2
Kupzog, Friederich ............................ TT-ISST3-1
Kurschl, Werner ............................... TT-EPT1-4
Kvas, Gernot .................................. TT-HCM1-3
Kwon, Key Ho ................................ TT-DENC2-1
Kwon, Key Ho ................................ TT-DENC2-2
Kwon, Key Ho ................................ TT-DENC3-3
Kwon, Key-Ho ................................ TT-BNA1-5
Kwon, Key-Ho ................................ TT-IIA3-5
Kwon, Keyho .................................. TT-HCM2-2
Kwon, Ki-Jeong ................................ TT-DENC4-1
Kwon, Soon-Kwang ............................. SS-IRA-4
La, Mai Thanh ................................. TT-FA1-4
Lai, H.F. ....................................... TT-SME-2
Lang, Roland ................................ TT-CCI4-2
Lastra, Jose L. Martinez ...................... TT-SOA1-5
Lastra, Jose L. Martinez ...................... TT-SOA2-2
Lastra, Jose L. Martinez ...................... TT-SOA2-3
Lastra, Jose Luis Martinez .................. TT-DENC3-2
Le, Hieu Tue .................................. TT-FA1-3
Le, Quy Ngoc ................................ TT-FA1-4
Lee, Chang Hoon ................................ TT-SCT3-6
Lee, Cheongjae ................................ TT-HCM2-3
Lee, Choon-Young ............................. TT-EPT1-2
Lee, Choon-Young ............................. TT-HCM1-2
Lee, Dong-Hyun ................................ SS-IR2-1
Lee, E.W. ...................................... TT-SOA2-1
Lee, E.W. ..................................... TT-ENIE2-5
Lee, Eng Wah .................................. TT-SOA2-1
Lee, Fu-Shin .................................. TT-DENC1-6
Lee, Fu-Shin .................................. TT-IIA3-2
Lee, Gary Geunbae ............................ TT-HCM2-3
Lee, Heung-ho ................................. TT-SCT4-4
Lee, Heung-ho ................................. TT-SCT5-1
Lee, Hoonjae .................................. TT-ISST3-2
Lee, Hou-Tsan ................................. TT-SCT1-2
Lee, Hyo Jik .................................. SS-LIE-1
Lee, Hyo-sung ................................ TT-SCT4-2
Lee, JangMyung ............................... TT-MR2-3
Lee, Jiaan-Der ................................ TT-SCT4-2
Lee, Jongchul ................................. SS-RF1-3
Lee, Joon Whoan .............................. TT-IIA6-3
Lee, Joon-Woo ................................. SS-IR2-3
Lee, Joon-Yong ............................... SS-ISA-3
Lee, JuJang .................................... TT-CCI3-3
Domain based Security for Mobile Agents

Najmus Saqib Malik, Friedrich Kupzog
Institute of Computer Technology
Vienna University of Technology
Vienna, Austria
malik,kupzog@ict.tuwien.ac.at

Michael Sonntag
Information processing and Microprocessor Technology
Johannes Kepler University
Linz, Austria
sonntag@fim.uni-linz.ac.at

Abstract—Mobile agent technology has many benefits but it suffers from the possibility of security breaches by agent platforms. In this paper, an infrastructure is proposed to secure mobile agents from the agent platform they reside on, which is especially suited for industrial automation devices having low computational resources. In this infrastructure, a Security Guider Bank (SGB) serves a group of agent platforms (AP), which is called a domain. The SGB maintains information about the domain, which is used by mobile agents to decide whether it is safe to visit the domain or not. This information is represented as vulnerability levels and reputation values. With this domain-based approach the turnaround time of agents is considerably reduced. Instead of collecting reputation information from each platform, the agent can use the cumulative history at the SGB. The SGB also maintains a copy of mobile agents during their visit of a domain, so that they can be renewed if altered by any AP during their journey. It recalculates vulnerability levels and reputation values after a specified amount of time or after the execution of a mobile agent at each agent platform. This scheme is able to detect as well as prevent - to some extent - malicious changes of mobile agents.

I. INTRODUCTION

Mobile agents are software agents [1] that can move around the network to complete a given task. It is a promising paradigm for e-commerce, intrusion detection, distributed and parallel processing or factory automation [2], [3], [4]. Mobile agents interact with other agents via an agent communication language [5]. Agent platform is an infrastructure providing the environment for mobile agents to be executed there. A mobile agent arrives and executes its task according to the privileges assigned by the creator of an agent and agent platform [6], and possibly the designer of its code as well, if different from the creator/owner [7]. A malicious platform can be a security breach for visiting mobile agents and vice versa [8].

This paper focuses on a mechanism through which mobile agents would be able to determine whether they would visit a particular domain or not. Through this decision, the probability of an attack from malicious agent platform on mobile agent is reduced [9]. A domain consists of agent platforms and is controlled by an entity called a manager [10], also termed as Security Guider Bank (SGB) shown in Figure 1.

In the proposed mechanism, an SGB is a trusted entity that maintains the history of a domain as a whole. The history is based the honest execution of mobile agents at a platform, i.e. whether the agent was executed without any modifications of the code or data or any undue influence like delays. To maintain this history, SGB uses two parameters:

1) Vulnerability Level $VL$ - represents a value for a domain that specifies a level based upon the fairness of execution of agents on platforms within a specific domain. A higher value means that a domain is more dangerous for mobile agents and a lower value suggests otherwise.

2) Reputation Value $RV$ - is a percentage value representing the degree of honesty of an agent platform. A larger value of $RV$ means that platform is safer to visit than one with a lower value of $RV$.

With this domain-based approach the turnaround time of agents is considerably reduced. Instead of collecting reputation information from each platform, the agent can use the cumulative history at the SGB. Now, mobile agents decide whether to visit a domain or not based upon the $VL$ and $RV$ values at the SGB. Since $VL$ reflects the cumulative reputation of agent platforms within a domain, it saves the turnaround time of mobile agents (the time that it would spend on checking the reputation of individual platform).

Penalties are assigned to malicious platforms by decreasing their reputation value. When the reputation value of a platform is decreased, fewer agents will visit the platform in future, since low reputation value is not acceptable for most of the agents to visit the platform. On the other hand, obtaining a high reputation value is advantageous for a platform, as more agents will visit it in future. Mobile agents are assigned with the acceptable $VL$ and $RV$ values by their owner before dispatching. These values reflect the protection level desired by the owner. A prerequisite of this scheme is that any malicious changes in an agent caused by any platform are finally detectable. This detection is carried out by the SGB or the agent’s owner. Malicious behavior can modify the mobile agent(s) or not fulfill commitments made to the agents.

Fig. 1. Security Guider Bank (SGB) operating a Domain

The proposed mechanism addresses direct malicious behavior. If the mobile agent is not modified, but is not served
honestly (e.g. delayed), the reputation value could additionally be based on input from the agent, its owner or both. This would improve the model, but is beyond the scope of this paper.

The term reputation used above refers to a property of a platform that is related to its trustworthiness. Reputation is determined by the SGB based on observations of mobile agents visiting the domain.

The remainder of the paper is organized as follows: section II discusses related work. In section III, a proposed infrastructure for the domain based security of mobile agents is described. Section IV provides a discussion on various aspects of the proposed mechanism. A scenario from the production area and its simulation is described in section V. Section VI explains the simulation results and section VII concludes the paper.

II. RELATED WORK

Many mechanisms are proposed in literature that employ cryptographic algorithms to secure mobile agents from malicious intentions of APs, but these schemes usually can only detect the alteration (if any) in mobile agents after its return from the platform. Black box techniques [11] and execution with encrypted function have limitations; e.g. the black box technique is still a theoretical approach and encrypted calculations are limited to polynomial functions [12]. A buddy model approach can also be used for mobile agent security [13] [14]. In this approach, two or more mobile agents called buddies take care of each other. But the buddy model suffers from the same weakness as other reactive approaches: it is no prevention technique. It only performs the rescue task after the buddy suffered from a malicious attack.

Many online business portals such as eBay [15] are based upon a reputation mechanism similar to the approach presented in this paper. Trust is stimulated between members by encouraging buyers and sellers to rate each other at the end of a transaction. This rating can be “positive”, “negative” or “neutral”, together with a short text comment. eBay summarizes the ratings submitted for each member, as well as the comments, and make them publicly available to all its users. Therefore, the reputation is submitted by the users (buyer and seller) not by a trusted third party. eBay is also positively biased; this might be an exchange courtesy [16]. In our approach, the trusted party (SGB) is similar to eBay as it summarizes the individual rating; but this rating is not calculated by the mobile agent itself, rather it is calculated by the SGB for each mobile agent. Additionally, the SGB analyzes and then changes the reputation value of platform or the vulnerability level of the domain based on the problems detected. However, with this approach the reputation is not positively biased and will be calculated impartially.

Basmasak and Zhang [17] present a mechanism in which the owner/creator of a mobile agent calculates and maintains a record of trust and reputation values of agent platforms. In this mechanism the agent platforms are called trusted third party hosts (TTP hosts). Before transferring the mobile agent, the agent owner selects a TTP host that has an acceptable level of reputation. The agent owner then updates the trust and reliability values of each TTP host upon the completion of each transaction. With this mechanism an agent owner has to be alive and keep in contact with the mobile agent for its whole journey. But by definition, mobile agents should be autonomous, which means they can move in a network without continuously interacting with their owner. Also, if there is large number of TTP hosts, like agent platforms spread over the Internet, then it is not a practical approach that the agent owner calculates a reputation and trusted value of all these TTP hosts. With the approach proposed in this work, mobile agents make the decision whether to visit a set of agent platforms by just examining the vulnerability level of the domain. This reduces the turnaround time of the mobile agent as the agent doesn’t need to check the reputation value of each individual platform.

III. DOMAIN BASED SECURITY FOR MOBILE AGENTS

The practical application focused in this paper is production automation, where production units are installed and connected to each other through a field network. Mobile agents from production site visit the sites of clients and receive orders (see section V for details). Each production/client site is called a domain. These domains may be connected to each other through the Internet or an intranet. Each domain is assigned a VL value (High, Medium and Low) controlled by the SGB. Basic services for the mobile agent like migration on the desired platform are provided by each agent platform.

Domain-based security is used to protect mobile agents from the suspicious changes that could be made by the adversary agent platform. It assigns penalties if any platform is found malicious. For example, if VL value is acceptable for mobile agent, then the agent will decide which particular platform(s) it will visit by observing the RV value of each platform. Both VL and RV values are maintained by the SGB. The delays introduced by adversary APs might not be detectable. Also, this approach also cannot prevent eavesdropping.

The SGB is a trusted entity for both the mobile agents and APs. The SGB

1) stores a copy of mobile agents and its hash code before sending them to platform.
2) maintains the VL of a domain and the RV of APs and communicates with mobile agents. Each mobile agent is assigned a safe VL value by its owner. When a mobile agent arrives at SGB, it compares VL value of that domain with the one assigned by its owner. The mobile agent will visit the domain if the VL value of domain is smaller than or equal to the value it has, otherwise not. Similarly, the owner also assigns a safe RV value to mobile agent so that it will only visit those APs, whose RV is greater or equal to RV value it contains.
3) transfers mobile agent, its hash code to AP with Counter of Platform CP = 0. This counter indicates the number of APs visited by a mobile agent within this domain. This counter is incremented every time the mobile agent is transferred to another AP.
When a mobile agent arrives at SGB from any platform within the domain then

- if mobile agent has successfully completed its journey, then it is sent on to the next domain.
- else (in case an AP performed malicious changes in mobile agent) recalculate the VL value of the domain and RV value of the sending platform and send the mobile agent to the next platform.

A. Algorithm for Mobile Agent (MA) Security

In the algorithm described below, the following parameters are used:

- n: Total number of platforms in a domain
- CP: No. of platform visited by a mobile agent
- CCP: No. of honest APs visit by mobile agents in domain
- m: Threshold value assign by implementer of algorithm.

When a mobile agent arrives at SGB from any platform:

1) Query the platform about its VL
2) If both VL (first: acquired from SGB, second: at step 1) are different, stop any further execution and move back to SGB, else continue
3) Start the actual task (query etc) and collect the result(s), if any.
4) Calculate the hash code of it-self (mobile agent) without new results
5) Compare both hash codes (first at step 4, second at step 5) if any.
6) Store the results permanently and calculate the (new) hash code with these results and send it to the SGB.
7) If VL == H then send result to SGB.
8) If VL == M and CP % (n / m) = 0 then send the results of the previous platform to the SGB.
9) If VL == L then continue
10) Fetch the RV of the next platform to be visited from the SGB.
11) If the RV is acceptable for the mobile agent (level predefined by the creator of the mobile agent) then
    • CP ++
    • compares the fetched RV (in step 10) and RV of current platform. If the RV of next platform is smaller, then send all the result collected from the previous platform to the SGB
    • migrate to next platform
12) Else continue with step 10.

Step 7 specifies that if the VL is high, mobile agents will send the result, which it collects from each previously visited platform(s), to the SGB before migrating to the next platform. Similarly, step 8 specifies if the VL is medium, then the mobile agent sends the results to SGB after n/m platform before migrating to the next one. For VL = L it will keep the result with itself only.

If a mobile agent arrives at SGB in step 2 or 5, this means that the AP is behaving maliciously; it sends a warning message and decreases the RV value of that platform, renews the mobile agent from its saved copy and sends it to the other platform. In both cases it increments the Vulnerability Level Counter VLC for the domain. The VLC is used to decrease (High ⇒ Medium or Medium ⇒ Low) or increase (Low ⇒ Medium or Medium ⇒ High) the Vulnerability Level of the domain.

The SGB adds up the CP (number of platform visited by a mobile agent) value into the CCP (Cumulative Counter of Platform; CCP = Σ CP, indicating the number of honest APs visit by mobile agents in a domain. CP is contained by each mobile agent, while CCP is maintained by the SGB for the whole domain. When SGB gets a mobile agent, it performs the calculation CCP = CCP + CP. If the mobile agent is returned in the middle of it execution, that is when the hash code differs at any platform, then the SGB perform CCP = CCP + (CP-1) to update the CCP value. It subtracts 1 from CP because the current platform is malicious.

The SGB identifies an adversary platform by comparing two hash codes, HC_{B} from step 6 at platform B and HC_{C} from step 4 at platform C, in step 5 (see section A). HC_{C} is calculated at the present platform C after the execution but before the result from the agent execution is received by the agent. HC_{C} is calculated at the previous platform B after the result has been received. By not including the results in the hash code comparison, a legal agent modification is masked. If any difference is found, then the current platform C would be malicious. One step before, B has already calculated and compared the hash code with the hash code calculate by platform A, and no difference was found. This develops a trust of SGB on B that it calculates the hash code of mobile agents with the results collected from B correctly. Therefore if C calculates a hash code which is not equal to the previous, then C is malicious.

B. Process for changing the vulnerability level

/* VLC is the number of time APs in the domain are found malicious, CCP is the number of time APs in the domain are found honest*/

/*T_{CCP}, T_{VLC} is temporary and used for this algorithm only and incremented each time when the CCP and VLC is incremented*/

1) T_{CCP} = CCP; // one time only
2) T_{VLC} = VLC; // one time only
3) if (DV >= 20%) 
   • VL ++; // (Low ⇒ Medium ⇒ High)
   • T_{VLC} = 0; T_{CCP} = 0;
4) if (VL != Low)
   /* If the number of times the AP(s) are found malicious
   are less than or equal to 5% of the number of times the
   AP(s) are found honest, then perform up-gradation */
   if (VLC <= 0.05 * CCP )
   VL := // (High => Medium => Low)
5) Broadcast the new VL to all APs

C. Reputation value behaviour and Restriction state

There are two points at which the SGB can determine that
a platform is malicious:
- If the SGB receive a mobile agent at step 2 or 5.
- When the mobile agent is returned to SGB after complet-
ing its journey.

In both above cases it increments the VLC (Vulnerability
Level Counter) and RC (Reputation Counter) for the AP that
was observed as malicious. At each increment it recalculates
the Reputation Value RV of each platform. Suppose X is the
total number of mobile agents that visit a platform and Y is
the number of mobile agent that are found modified, then the
Reputation Value is given by

\[ RV = 1 - \frac{\sum_{i=0}^{Y} (i)}{\sum_{j=0}^{X} (j)} \]

The equation above shows that with a larger value of RV
the AP is safer to visit as compared to a platform with a
smaller R value. An agent platform is in restriction state if
it changes more than r% of visiting agents. This percentage
value represents the threshold value after corssing it the AP is
in restriction state. Higher percentage reflects sticker security
then lower percentage. Once the AP is in restriction state its
RV value can be improved as follows:

1) Send a dummy mobile agent to the AP
2) If the mobile agent is modified on return
   - y ++, z = 0 and recalculate RV
   - if (RV < 0.4)
     - Terminate AP and send notice to AP owner.
3) else
   - z ++ //z is a counter initialized with zero
   - if (z == y) x ++ and recalculate RV
   - else goto 1
4) if (RV >= 0.6 AP's restriction state is over
   z is a counter initialized with zero at the beginning

IV. DISCUSSION

This infrastructure is also useful for e-commerce appli-
cations where different companies have installed their host
machines (agent platforms) within a domain, to provide their
services to visiting mobile agents. Penalizing for malicious
behavior of agent platforms can e.g. be done by financial
punishment.

A threshold level for consistent malicious behavior could
also be introduced: by crossing this limit an agent platform
would face exclusion from that domain or heavy monetary
fine. Time constraints can be added to this mechanism as well.
That is, the SGB would wait for a specified amount of time
for the mobile agent that has already been sent to an AP. If
it doesn’t receive the mobile agent’s response by that time
it will decrease the reputation value of platform and transmit
the mobile agent to another AP (which is possible, as the SGB
retains a copy of each mobile agent). This copy would be
deleted when the mobile agent has successfully migrated
away from that domain.

When any AP is in restriction state then the SGB sends
dummy mobile agents to this AP. These dummy mobile agents
would be copies of actual mobile agents that had visited the
domain in past, preferably such, which had been modified. As
there might be a situation in which the AP only alters certain
type of mobile agents, keeping copies of those and reusing
them in future would better help in determining whether an
AP has changed its malicious behavior or not. The SGB has
to be careful that the dummy mobile agents would not make
commitments with the APs). These commitments, if occurred,
would bind the agent’s owner, who has no knowledge of
this duplication. The SGB should be equipped with such
a mechanism that it would determine which mobile agent
has to just perform computations (like searching a database)
and retrieve the results. It might be possible that the SGB
creates dummy agents by itself, which will not make any
commitments and just copy some fields from affected mobile
agent like their ID or intermediate results.

Before execution of a mobile agent at an AP, it calculates
its hash code. Practically, this hash code is not of the whole
mobile agent but for its critical part only. This critical part
includes creator information, the itinerary, and its data store
(in which the mobile agent puts the information it collected
from previously visited platforms) carried by the mobile
agent. Creator information includes creator identity as well as
assigned VL and RV values. This information is signed by
the SGB, so that individual platforms cannot modify these values
secretly.

Mobile agents can save the time that they spend on agent
platform when requesting the RV value of next agent platform
from the SGB (step 10 in mobile agent algorithm). This can
be achieved by mobile agents when they enter in the domain,
it can examine the RV values of all agent platform on the SGB
and create an itinerary list of those agent platforms that possess
an acceptable RV value for mobile agent. But this approach
has a drawback: there is a possibility that RV value of any
agent platform(s) which was acceptable for mobile agent at
the time it made the route list, but can be change (increased or
decreased) after its journey has been started. We will address
this issue in future work.

The certain benefit of this mechanism is that when the
mobile agent finds the VL value to be within an acceptable
level, then it is most likely that it will find most of the
platforms in that domain to possess acceptable reputation
levels as well. Therefore, when a mobile agent decides not
to visit a certain domain, it saves the time in migrating there
and what it would spend in inquiring the reputation levels of the individual platforms.

V. SCENARIO FROM PRODUCTION AREA AND SIMULATION

We consider a scenario (shown in Figure 2) in which a company that offers different kinds of production services takes production order from its clients. The company collects these orders from clients that are connected to each other through the Internet. For this job it creates mobile agents that move from one client to another. A mobile agent is assigned an acceptable Vulnerability Level and Reputation Value by the owner, before dispatching. Each client and the company are domains which contain a set of APs. A client is composed of different departments; each department has its AP that is used to communicate with the mobile agents. Each department requiring production services from the company will give orders to the mobile agent when it visits them. The SGB is a third party in which each of the client and company place trust. OMNet tool is used for simulating the production scenario described above. A model is setup containing multiple domains that are connected with each other through local and wide area network. In this simulation 100 mobile agents are sent over the network, which want to visit all those platforms. There are four domains in this simulation. Each domain contains five APs. Initially, all the domains are assigned a low Vulnerability Level (VL=1) and APs a high Reputation Value (RV=1). The Vulnerability Level increases every time, when 15% of the mobile agents are found as modified maliciously. The increase in vulnerability level means that this domain is more vulnerable to performing malicious changes in the mobile agents than one having a lower vulnerability level. Similarly, the highest reputation value is one, and if it is less this means that this AP has performed malicious changes in the past. All mobile agents are assigned with low vulnerability level, 60% of mobile agent have VL= Low and 40% have VL= Medium. Similarly, high reputation values are assigned to mobile agents by the creator; 50% of mobile agent have RV=1, 30% have RV=0.95 and 20% have RV=0.90.

With the above set up, the network is simulated by considering three cases, which are:

- Best case: Identical to the initial set up where no AP is malicious.
- Average case: The case in which only a few APs are acting maliciously.
- Worst case: A setup in which a majority of APs are evil.

In the average case, approximately half of the APs perform malicious changes on the mobile agents. With every malicious change detected by the SGB, there is a decrement in the Reputation Value of the AP. When the SGB detect that 20% of mobile agents are changed maliciously, then the Vulnerability Level of that domain is incremented by 1 by the SGB. Similarly, in the worst case we assumed that approximately most of the AP behaves maliciously.

![Fig. 2. Domain based security in Production Area](image)

VI. RESULTS

The periodicity of agent platforms is defined as the size of a set of agents in which one agent is maliciously changed. Increasing periodicity means that malicious nature of the platform appears with larger period; for example, with periodicity 5 the platform will change every fifth mobile agent that visits it.

Figure 3(a) and 3(b) shows the number of mobile agent visited by each platform in average and worst case. With increasing periodicity, the number of APs visited by the mobile agents also increases, since the platform changes lesser and lesser mobile agents. For larger periodicity values, the curve increases with smaller slope, since the effect of changes becoming more and more weak in respect to the total number of visited platforms. This can be seen in 3(b) (where 100 mobile agents visit the agent platforms). Here, for periodicity values larger than 20 the slope of the curves is very low.

![Fig. 3. Average No of AP visited by mobile agents with different periodicity](image)
At a certain value of periodicity the number of mobile agents that are maliciously modified will become less and RV and VL values are acceptable for most of mobile agents. In the best case, where no modifications by the APs will take place, the periodicity approaches to infinity and all mobile agents visit approximately each platform.

Figure 4 shows the number of agent platforms visited by mobile agents in worst and average case scenarios depending on the RV threshold value assigned to mobile agents by their owner. In the experiment, initially all the reputation values of APs is High. Then, some agent platforms begin to change the mobile agents maliciously, according to the periodicity assigned. As a result, the RV values of these agent platforms start decreasing. The mobile agents with high RV threshold values visits less APs (start of graph) as compared to the tail of the graph) since some APs behave maliciously and their RV values changes. APs having low RV suffer in serving less or no mobile agents in future as a penalty. Here it is assumed that serving larger number of mobile agent is an incentive for AP.

The results of these graphs depends upon the RV value of each AP, VL value of the domain and assigned RV, VL threshold values of each mobile agent. It can further be observed that this mechanism is successful in protecting the mobile agents and shows that there is a tradeoff between the number of agent platform visited by mobile agents and the level of security, desired by the agent owner (expressed in the RV and VL threshold values).

VII. CONCLUSION

With the proposed scheme, mobile agents can be protected from the possible attacks of malicious platforms by deciding whether to visit an agent platform or not. This decision is made by the mobile agent at the security manager before arriving at the platform. The agent first checks if the VL value of the domain is acceptable, then it selects all those agent platforms in its itinerary list whose RV values are acceptable. Scenarios from the production area were simulated with RV and VL values changing according to the platform’s periodicity. In these simulations, average and worst case scenarios are considered, according to the number of malicious agent platforms. This mechanism is useful where the mobile agents visit multiple sites that are connected through a network (which can be private or public). This mechanism saves turnaround time of mobile agents, as it allows a decision by looking at the VL value whether it has to visit whole domain or not.

Still needed is a global agreement upon the rules for calculating and interpreting the vulnerability level and reputation values in the form of standards (e.g. in terms of choice of the free parameters in the described algorithms). This would allow mobile agents, when moving over the Internet, to have the same understanding of VL and RL values for all domains and included agent platforms.

REFERENCES