

# Preserving Obliviousness Characteristic of Honeypot database

S. K. Gupta<sup>†</sup>   Anand Gupta<sup>‡</sup>   Renu Damor<sup>†</sup>   Vikram Goyal<sup>†</sup>   Sangeeta Sabharwal<sup>‡</sup>

<sup>†</sup> Indian Institute of Technology, Department of Computer Science & Engineering, New Delhi, India

<sup>‡</sup> Netaji Subhas Institute of Technology, Division of COE, New Delhi, India

skg@cse.iitd.ernet.in, nsit\_anand@yahoo.com, renu.damor@yahoo.com, vkgoyal@cse.iitd.ernet.in, ssab23@yahoo.com

## Abstract

The concept of honeypot has been explored by many researchers in the network domain. We have proposed database honeypots and have given its architecture[5]. The use of such a honeypot is to identify potential attackers before an attack actually takes place. The privacy policy as advocated in [6] is expected to deny the access to a suspected user. However, one would like to identify the suspected user who might have entered into the system under some disguise (masquerading). We propose that a suspected user be provided with synthetic information (in place of denial of access) with the help of which the administrator could confirm the suspicion. In this paper, we give certain characteristics of such a honeypot namely, luring, determination of suspicion and transparency to the user and give techniques for it being oblivious to the user.

## 1 Introduction

Till now honeypots [2, 3, 4] have been proposed as an early warning security system tools in network domain where security involves prevention of unauthorized access to network or data. The general use of these honeypots is to watch suspicious activities by probable attackers and later confirm or deny their suspicion. These network honeypots enhance the security by detecting and then inferencing the intention from suspected user moves. A lot of work on privacy and prevention of unauthorized access in databases have been proposed since the pioneering work by Rakesh Agrawal[6] proposing a framework of privacy policy. The privacy policy comprises of a set of rules which are expressed and stored using language such as P3P [10] or EPAL [11]. The enforcement of privacy policy has been proposed both at application level and at query level. We feel that the privacy of data is not foolproof. There can be a user who can masquerade into the system as an authorized user with access right and violate the robust privacy policy [7,8,9]. For instance, a hospital collects personal information from a patient with the promise that his information will be disclosed to only those for whom he has given his

consent. A famous actress comes to the hospital with the intention of getting good treatment and with a belief that her privacy will not be breached ends up in loosing her private information in the following way: A news company bribed the nurse to intrude and get the facts about the actress. The nurse is able to masquerade (using some illegal means) as doctor and obtains private information (which she is otherwise not authorized and allowed). She then sells it to the news company. Hence the privacy get breached which leads to compromise. Such privacy breaches can not be safe guarded even after having robust privacy policy and any other strong prevention system. To the best of our knowledge, this issue has not been considered in literature so far. A new kind of honeypots called context dependent honeypot have been advocated [5] to address this need. The idea of context honeypots database is to watch such suspected users and then to confirm or deny the suspicion. The context honeypot is built for a specified malafide intention. The utility of the honeypot depends on how well it is able to confirm the suspicion. We propose certain characteristics of a honeypot to help it in achieving its purpose. One of the characteristics, that the context honeypot should be oblivious to the suspected user, is addressed here. Being oblivious to a user means that user should not suspect that he/she has been put in honeypot. If a user becomes aware that he has been put in a honeypot, then he can accordingly behave in such a way which would render confirmation of suspicion more difficult or impossible. We therefore propose that the synthetic information made available does not arise suspicion to the user, that is, honeypot is oblivious to the user.

The organization of the paper is as follows: Section 2 introduces the characteristics of context honeypot. Section 3 discusses various techniques of obliviousness. Finally in section 4 it is followed by conclusion and future work that completes the paper.

## 2 Characteristics of a Context Honeypot

The nature of context honeypots is quite different from the nature of network honeypots. While network honeypots mainly address the issue of security (particularly, denial of service), context dependent honeypots address the issue of privacy. They are expected to confirm the suspicion of a

potential intruder.

We identify the following characteristics of a context honeypot in order to address the desired functionality.

- (a) Lure a suspected user.
- (b) Oblivious to a user.
- (c) Confirm the suspicion.

We describe the three characteristics in this section and discuss (b) in section 3 in more detail.

Although the above three characteristics are not wholly independent. We first address them individually (independently). We shall address the issue of their interdependency later.

## 2.1 Lure a suspected user:

A suspected user may be provided with synthetic data (not factual) which is likely to be helpful in confirming the suspicion, seemingly fulfilling his/her malafide intention. By luring an intruder, honeypot serves following purposes:

- The intruder can be watched and the forensic information (about the intruder) can be gathered so that it can be used as proof against him in future.
- By studying the activities of intruder, vulnerability in current system can be explored and security level can be enhanced.

Luring in general is defined as an act of attracting with promise of pleasure or reward.

In network domain [4], luring of an attacker can be accomplished in several ways, depending upon malafide intention of the user. Luring involves simulation of personality of any operating system such as Red hat Linux, Microsoft Windows NT etc. as per need. Luring may involve configuration of different services such as HTTP, SMTP, SSH etc. Practical application of luring is Honeyd. Honeyd [12] simulates virtual hosts on the network. In order to lure an attacker, an entire network topology is simulated within one machine having multiple hops, packet losses and latency. Establishment of many virtual routers can also be seen.

We redefine luring with reference to context honeypot as follows:

**Definition 1** *It is an act with the help of which a suspected user may be provided with synthetic data which is likely to be helpful in fulfillment of his/her malafide intention.*

For instance, in the example mentioned in section 1, nurse is a suspected user and this suspicion over nurse is an external input for context honeypot. Now context honeypot is expected to provide an attractive synthetic information about the actress in such a way which would help an administrator confirm suspicion.

Luring in databases requires essentially two things. These are:

- Malafide intention.
- Type of suspected user.

## 2.2 Oblivious to a suspected user:

If a suspected user who has directed to honeypot for confirmation of suspicion becomes aware of the fact that he/she has been put in honeypot, it would be self-defeating and would defeat the purpose. Therefore, an important characteristic of a context honeypot is that it should be oblivious to the suspected user. Indeed, absence of this characteristic would make honeypot totally useless. Absence of it might make the system more vulnerable than it was before. Therefore in this paper, our focus is on the property of context honeypots that a suspected user (probably masquerader?) who has been put in honeypot should not be aware of being put in the honeypot. We focus this property in section 3 of the paper.

## 2.3 Cause anticipated change in behavior:

The purpose of context honeypot is to identify a potential intruder. We propose to accomplish this by providing synthetic information to a suspected user. It is expected that this synthetic information is likely to cause a change in behavior which would help distinguish between benign and malicious user. For example, we suspect (through some external stimuli) that a bank customer is suspicious. We propose that he may be provided with a synthetic (reduced) available balance in his account. If the user is genuine, then he/she would contact bank to get it rectified. The anticipation process will be context and specific malafide intention dependent. It may involve intense observation of the suspected user and may be a multi-step process. This is the most important component of context honeypot. The building of anticipation for context honeypot involves following parameters

- Creation of a user profile on the lines of cookies which depends on <userid, role, pastqueries, otherattributes >
- Building of a profile and identifying parameters for comparison
- Comparison of current and past user profile.
- Specific malafide intention

**Current Action** The action taken by the user in response to the synthetic data provided by the honeypot is called current action. It is denoted by  $A_{current}$ . There can be  $k$  malafide intentions say  $(m_1, m_2, \dots, m_k)$  and we presume that for each intention, there can be an infinite set of actions for synthetic data shown. We will take a finite set of actions from that infinite set to make the honeypot database more focused. It means  $m_1: (a_{11}, a_{12}, \dots, a_{1k})$ ,  $m_2: (a_{21}, a_{22}, \dots, a_{2k})$  and so on. For example, suppose that a user is a student in a college and wants to see his grade. He queries and sees the result. The result shows him that he got E grade in one subject and that brings his average grade in academic very low. Now to modify this scenario in his favor, there can be many actions. Such as, one can choose to reappear in the exam, or alter the grade either

through talking to professor and convincing him to alter his grade or the grades of the others so that his standing in class can come up. All the above actions are benign. If he masquerades as Professor and alter his and/or others grades then this action termed as malafide. This way all finite actions are grouped under malafide or benign. Thus the current action by the user is tagged accordingly.

**Current Query** The query issued by the user is which is still unanswered called current query. Its is represented by the symbol  $Q_{current}$ .

**Past Query** Any answered query that has been issued by the user in the current session is called Past Query. It is represented by the symbol  $Q_{past}$ . All past queries can be obtained by the query logs which the information system maintains.

**Anticipated Action** The action which a malafide user is expected to take on the basis of disclosed synthetic data is called Anticipated Action. It is denoted by the symbol  $A_{anticipated}$ . The anticipation policy contains all the anticipated action for any synthetic data. As in the college student instance where the malafide intention is to upgrade his ranking by wrong means. All the finite possible actions are stored. The anticipation policy contains all the anticipated action for any synthetic data. So any action taken by the user is termed as anticipated action.

**Anticipation Policy** Anticipation Policy is a set of rules which defines all permissible actions that can be taken by a malafide user provided with some synthetic data. These rules are represented in table 1.

Table 1: Anticipation Policy Table

Initial State	Next State	Query	Synthetic Data	Action
$S_{start}$	$S_0$	$Q_0$	$D_0[0]$	$A_1$
$S_0$	$S_0$	$Q_0$	$D_0[1]$	$A_2$
$S_0$	$S_1$	$Q_1$	$D_0[2]$	$A_3$
...	...	...	...	...

### 3 Techniques of Obliviousness

As stated earlier, context honeypot operates by providing synthetic information to a suspected user to determine confirmation. However, obliviousness is an important characteristic. If a suspected user is provided with such synthetic information which enables him/her to determine whether he/she has been put in honeypot, then the user can take step to bypass the honeypot process. So, the synthetic information should be such that the user can easily believe it. For this, the synthetic information should

(a) in range, and

(b) otherwise not available to the suspected user.

For example, if range of salary domain of a Professor is between 18k and 25k and if a suspected user who asks for salary of a professor provided with an answer is 30k or 5k, then he/she can suspect to have been put in the honeypot. However, if the actual salary of a Professor is 22k, but the suspected user is provided with 24k, then there is less chance of user becoming suspicious. Similarly if a suspected user can otherwise determine the salary, then salary component should not be changed. For example, if a suspected accountant asks for salary, then it should not be changed, however we may change address since the accountant is not given access to address. The proposed synthetic table should contain values which satisfy the range constraints for each domain effectively by taking care of issue (a).

We propose the following two step process to address the issue raised in (b) above.

*Step 1:* We expect that the specific malafide intention of the suspicious masquerader is supplied externally (through some stimulus, either manual or otherwise). We may either be provided with some information as to who has possibly masqueraded as whom (albeit with a degree of suspicion) or we may have to determine who has masqueraded (Since an access is logged as suspicious, we know that whose login was masqueraded or what masqueraded has taken place).

*Step 2:* Determine what information can be released for this suspicion (from the synthetic table) which satisfies the oblivious property of the honeypot.

#### 3.1 Determine the Suspicious Masquerader

The determination of suspicion will assist in the synthetic information to be disclosed to the suspicious user. We propose the following two environments :(i) The identity of the masquerader is supplied through some external stimuli, or (ii) we attempt to determine it approximately by an iterative process. We illustrate by the following example.

Consider a hospital domain where the user type and the role involved are as shown in figure 1.

User Type	Role
A	Doctor
B	Nurse
C	Billing Officer

Figure 1: Role and User

Here the different user types are A, B and C .Their

Table 2: Hospital Database: Attributes of each Role

<p><b>Doctor</b>                  (DocId(R), PatientID(R), Present and Past History of Illness(R/W), Patient_Age(R), Patient_Sex(R), Marital_Status (R/W), NurseID (R/W), Nurse_action (R/W), Nurse_Report (R), TechnicianID(R/W), Test_Id (R/W), Test_Report (R), medicineid(R/W), Patient_Diagnose_over(R/W) )</p> <p><b>Nurse</b>                  (NurseId(R), PatientID(R), Patient_Age(R), Patient_Sex(R), Nurse_action(R), medicineid(R), Nurse_Report(R/W), Patient_Diagnose_over(R/W) )</p> <p><b>Billing officer</b>                  (BO_ID(R), PatientID(R), Test_Id(R), DocId (R), NurseId (R), medicineid(R), Test_Rate (R/W), Doctor_Rate(R/W), Nurse_rate(R/W), Medicine_rate (W),Bill_status )</p>
---

attribute sets are represented in table 2.

**Privacy Policy:**

The privacy policy is illustrated in table 2 consisting doctor , nurse and billing officer as individual. The access rights are indicated against the attributes for the individual (identified by corresponding id).

This attributes for each role along with their read and write access are shown.

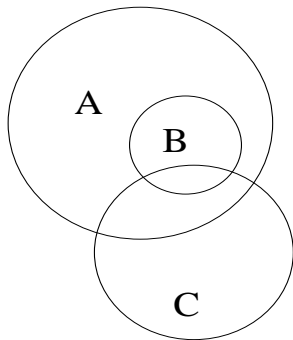


Figure 2: Venn Diagram: Representing Hospital System

**Environment 1**

In this case, we have been supplied (externally) the probable identity of the masquerader. We further assume , for the sake of simplicity, following assumptions.

- Each Role has exactly one user.
- Data is static(No volatility).
- One masquerader masquerades only one authorized user at a time.

The proposed approach is described for the environment in

section 3.2.

**Environment 2**

In this case, we have to determine the suspected masquerader before proceeding further. For our example, the possible combinations are given in figure 3. We further assume that masquerader is unknown and we are here to determine who it can be in addition to the assumptions made for environment 1.

Masqueraded User	Masquerader
A	B C
B	A C
C	A B

Figure 3:

For identifying the masquerader we have proposed some fabricated loops. If the masquerader makes a mistake then according to rule he will be assumed as a particular masquerader. The number of loops with different functionality can be increased or reduced so that the dynamic nature exists. As for now we will discuss 4 loops:

**Notation**

Let N(WV) can be

- (i) a fact
- (ii) one variable unknown
- (iii) two variable unknown

W denotes the masquerader.

V is the user whose login is masqueraded.

This notations will be used throughout in all loops.

**Loop1**

From our example in figure 3 all possible combinations for N are :

S : {(AB), (BA), (AC), (CA), (BC), (CB)}

Here, in this case , W is unknown and V is Known. Our aim is to explore W and find out who can be the possible masquerader. For example, suppose the user B is victim. So, from the set S all possible combinations are {(AB), (CB)}.

Now, our goal is to identify whether it is user A or C. Suppose, unknown user issues a query which belongs to domain of user A (His own domain instead of domain of user B). Then, it can be taken as an indication that possible masquerader is user A. Since, our focus is on obliviousness property of honeypot, so system will continue to show that

Table 3: IP access rules

IP ID	IP Address	Masqueraded User
IP1	192.168.42.226	C,B
IP2	192.168.42.223	A
IP3	192.168.42.221	B

data which comes under domain of  $(A \cap B)$ , instead of denial of displaying of data. As per privacy policy, data which belongs to domain of  $(A - B)$  must be hidden, while the data which comes under domain of  $(A \cap B)$  will be original data. Data which belongs to the domain of  $(B - A)$  is shown with synthetic values. This is represented by the following statement:

$$M_{AB} \leftarrow [ Q'_{original}(A) \cap Q'_{original}(B) ] + \text{Hide} (Q'_{original}(A) - Q'_{original}(B)) + \text{Synthetic} (Q'_{original}(B) - Q'_{original}(A))$$

The figure 4 shows through Venn diagram the construction of logical view.

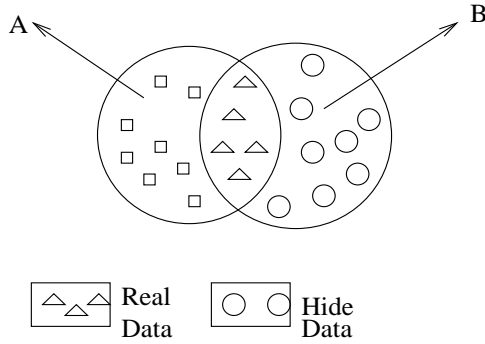


Figure 4: Venn Diagram: Loop 1

**Loop 2**

If user W tries to access from the location for which user V is not authorized, then privacy policy will deny such access immediately. If there is an immediate access from some other location, for which user V is authorized, then user W can be interpreted as possible masquerader. For example, table 3 depicts general access location rules for all legitimate users. Now, if user A tries to access login of user B ( $\bar{A}B$ ) from IP2 then privacy policy will deny such access, as user B does not have the access right from IP2. Immediately after this access, if user A tries to access login of user B from IP1 or IP3 then it can be assumed that the possible masquerader is user A and he is masquerading as user B.

**Loop 3**

Table 4 displays time schedule for undertaken hospital system. This schedule enlist all users along with their permitted time hours. If there is an attempt to access login of user V (as per notation) at time t and time t is not part of permitted time schedule then this is an indication that login of user v is being masqueraded. In order to determine

Table 4: Time Schedule

Rule ID	Schedule	
	IN	OUT
T1	9:00	17:00
T2	17:00	1:00
T3	1:00	9:00

Table 5: User Availability status on Holiday ( $\checkmark \leftarrow$  Available,  $\times \leftarrow$  Not available)

User	Holiday Status
A	$\times$
B	$\checkmark$
C	$\times$

possible masquerader, we look at the permitted time schedule of each user individually (By using figure 5,6,7). We try to locate those users, for whom t is part of permitted time schedule. Such user/ users can be possible masquerader/masqueraders.

For example,  $N(WV)$ , in this expression, V is user B and W is to be determined.

If login of user B is accessed at 18:00 then user B is not authorized to access his/her login as per schedule. This access leads to an indication that login of user B is being masqueraded by someone. In order to locate possible masquerader, we look at those users who were authorized to access at time t. In our case, t belongs to time slot T2 and for this time slot T2, only user c is authorized. Thus we may conclude that user c is possible masquerader.

Rule ID	Allowed
T1	$\checkmark$
T2	$\times$
T3	$\times$

Figure 5: User A

Rule ID	Allowed
T1	$\checkmark$
T2	$\times$
T3	$\checkmark$

Figure 6: User B

Rule ID	Allowed
T1	$\times$
T2	$\checkmark$
T3	$\times$

Figure 7: User C

**Loop 4**

Table 5 displays availability status of each user on holiday. If there is an attempt to access login of V and V is not available on this day, then it leads to an indication that login of V is being masqueraded. In order to determine masquerader, we check availability status of other users. All those users who were available on holiday comes under suspicion. For example, If there is an attempt to access login of user A, then possible masquerader is user B as he is available on that day.

Table 6: Properties from Hospital Domain

<ul style="list-style-type: none"> <li>• <math>A \cap B : B</math></li> <li>• <math>A - B : Present\ and\ Past\ history\ of\ illness,\ Investigation\ Report</math></li> <li>• <math>B - A : \phi</math></li> <li>• <math>A \cap C : TestID\ \&amp;\ medicineid,\ etc</math></li> <li>• <math>A - C : present\ and\ past\ history\ of\ illness,\ age,\ sex,\ marital\ status\ and\ other\ fields</math></li> <li>• <math>C - A : Bill\ value\ detail\ / status</math></li> <li>• <math>B \cap C : medicine\ id,\ test\ id</math></li> <li>• <math>B - C : Patient\_Age,\ Patient\_Sex,\ description\ of\ nurse\ action</math></li> <li>• <math>C - B : Bill\ value\ detail\ / status</math></li> <li>• <math>A \cap B \cap C : B \cup (A \cap C) : PatientID,\ Nurse\_id,\ medicineid</math></li> <li>• <math>A - (A \cap B + A \cap C - A \cap B \cap C) : present\ and\ past\ history\ of\ illness</math></li> <li>• <math>B - (A \cap B + A \cap C - A \cap B \cap C) : \phi</math></li> <li>• <math>C - (A \cap B + A \cap C - A \cap B \cap C) : Bill\ value\ Detail\ / status</math></li> </ul>
---

### 3.2 Constructing Synthetic Matrix

The strength of obliviousness property depends on how well we are able to identify the masquerader. In notation,  $N(WV)$ , Information about  $W, V$  is already determined in section 3.1. Next step is to determine what synthetic information must be provided to masquerader. This provided information will be constructed in form of synthetic matrix denoted by  $M_{AB}$  described as :

$$M_{AB} \leftarrow R \cup T$$

$R \leftarrow$  original data

$T \leftarrow$  Synthetic data

Domain of  $R$  is determined by  $A \cap B$

Domain of  $T$  is determined by  $B - A$

Accordingly, a logical view has been created in this section so as to satisfy oblivious property. The Venn diagram shows the domain of legitimate data of each user. In figure 8 the intersection shows the common domain part for which each of them are authorized to access. The Table 6 describes the Mathematical part in co relating the figure 2.

There after the query  $Q$  posed by masquerader( $W$ ) is passed through privacy policy with respect to masqueraded user( $V$ ). We are in receipt of query  $Q'$  and divert it to constructed synthetic view. The construction of this logical view is explained in Box 1.

### Box 1: Synthetic Information Matrix

#### Terms Used

Let  $Q(A) \leftarrow$  Discloses Domain of Data related to  $A$

and  $Q(B) \leftarrow$  Discloses Domain of Data related to  $B$

$Q_{original}(A) \leftarrow$  Discloses complete original information related to  $A$ , for which  $A$  is legitimate

$Q_{original}(B) \leftarrow$  Discloses complete original information related to  $B$ , for which  $B$  is legitimate

$M_{AB} \leftarrow$  Complete Domain of Data consisting of either (i) synthetic information or (ii) synthetic information + original information so  $B$  masquerades as  $A$ .

$Q' \leftarrow$  Query issued by user which gets modified due to privacy policy related to that user

**Choice of  $M_{AB}$**

$$R = Q_{original}(A) \cap Q_{original}(B)$$

/\* Identifying common attribute and their respective domain values for which masquerader and masqueraded user are eligible \*/

$$T = Synthetic \leftarrow (Q_{original}(A) - Q_{original}(B))$$

/\* Converting the domain of values to Synthetic for which masquerader is not eligible \*/

if  $R = \phi$  then  $M_{AB} \rightarrow$  All the Data of  $A$  is made synthetic

Else  $M_{AB} \leftarrow [T + R]$

Query  $Q'$  is diverted to the table of  $M_{AB}$

### 3.3 Interaction of Masquerader

In previous section, we have already determined who is possible masquerader and how synthetic information should be given to him in order to preserve obliviousness property. In this section, we show interaction process of masquerader using box 2. Each access is recorded in the log file. If history of masquerader exists and again masquerader poses another query for same malafide intention then current response must be consistent with previous responses (in terms of previous used domain and previous shown values.)

In order to determine response, we need to consider two cases as follows using venn diagram (Figure 8):

Case 1: if domain of  $A$  legitimacy and  $B$  legitimacy is NULL.

Case 2: if domain of  $A$  legitimacy and  $B$  legitimacy is NOT NULL

(a) Domain of intersection is shown with original information.

(b) Domain specific to  $B$  but does not include the intersection part should be made synthetic.

(c) Domain specific to  $A$  but does not include the intersection should be hidden.

These parts are joined in order to respond the given query posed by masquerader.

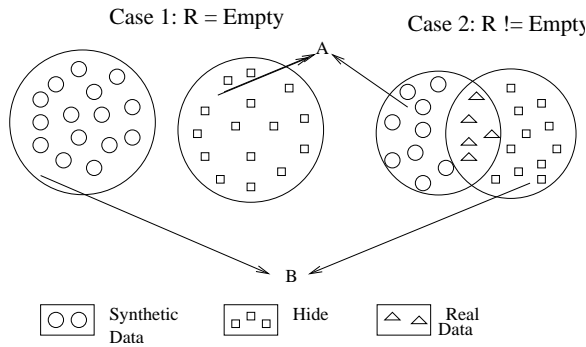


Figure 8: Venn Diagram: A Masquerade as B

### Box 2: Interaction Algorithm

#### Term Used

$Q \leftarrow$  Query posed by a user  
 $Q' \leftarrow$  Query issued by user which gets modified due to privacy policy related to that user  
 $Q_{original}(A) \leftarrow$  original information posed by Query  $Q$  for which A as legitimate and authorized  
 $Q_{original}(B) \leftarrow$  original information posed by Query  $Q$  for which B is legitimate and authorized  
 $M_{AB} \leftarrow$  Matrix Logical view for a masquerader B who masquerades as A  
 $Syn_{AB} \leftarrow$  Result view because of Query  $Q'$

#### Query Interaction

##### Step1 : First Session

/\* If a particular masqueraded user and masquerader combination does not exist in Log \*/  
 $R = Q_{original}(A) \cap Q_{original}(B)$   
 if  $R = \phi$   
   then  $Syn_{AB} = Q'(M_{AB})$   
   else  $\{ T = Synthetic \leftarrow (Q_{original}(A) - Q_{original}(B))$   
        $Syn_{AB} \leftarrow [T + R] \}$   
 if  $(Syn_{AB} \subseteq M_{AB}) \cap (Syn_{AB} \supseteq M_{AB})$   
 then  $\{ Z_{AB} = Syn_{AB}$   
 Record in Log Complete =  $\checkmark$  }  
 else  $\{ Record in Log Complete = \times \}$   
 /\* Show the Result which is in  $Z_{AB}$  \*/

##### Step2: Second Session

Check log file for a particular masqueraded User and masquerader  
 /\* If a particular masqueraded user and masquerader Exists in Log \*/  
 $Z_{AB} = \cup_{i=1}^n Syn_{AB}(i)$   
 if  $(Q(M_{AB} \subseteq Z_{AB}) \cap Q(M_{AB} \supseteq Z_{AB}))$   
 then  $Q'(Z_{AB})$   
 else  
   Begin  
      $R = Q'(Z_{AB}) \cap Q'(M_{AB})$   
     if  $R = \phi$  then  
        $Syn_{AB} = Q'(M_{AB})$   
     else  $\{ T = (Q'(M_{AB}) - Q'(Z_{AB}))$   
        $Syn_{AB} \leftarrow [T - M_{AB} + R]$   
        $Z_{AB} = Syn_{AB} \}$   
     Store in log  $Z_{AB}$  and show the result  
   End  
 Show  $Q'(Z_{AB})$

Table 7: Log File

Masqueraded User	Masquerader	Result	Complete
A	B	$Syn_{AB}$	$\times$
...	...	...	...

## 4 Conclusion and Future work

A lot of research work has been carried out in the field of security and privacy policy. However, no system is fool-proof and there can be an intruder, who accesses system as masquerader. Such access leads to the privacy breaches even after implementation of robust privacy policy. So, it is felt that there is need to detect such possible breaches in the system, which can determine vulnerabilities and therefore improve privacy.

In this paper, concept of context honeypot is proposed along with its characteristics. Context honeypots provide synthetic information to suspected users with an aim to determine the suspicion. One of the basic characteristics is *Oblivious to the user*. Obliviousness of context honeypot needs to be preserved.

If a suspected user becomes aware that he has been put in the honeypot then he can take appropriate step which can defeat the purpose of the honeypot. We have addressed the issue of obliviousness in this paper based on knowledge of certain parameters, i.e. the masquerader and the masqueraded user.

We are currently working on other characteristics of honeypots i.e. luring and determination of suspicion mentioned in this paper and their interaction in context honeypots.

## 5 Acknowledgment

Our thanks to Surendra Tomar, Pratibha, Sheetal Tewari of Database Group in I.I.T Delhi for their invaluable suggestions. We wish to thank the N.T.R.O. and Ministry of Information Technology, Government of India for funding the project.

## References

- [1] Marcel-Jan, Pete Finnigan Ever used database honeypots? Available at <http://www.petefinnigan.com/forum/yabb/YaBB.cgi?board=>
- [2] Feng Zhang; Shijie Zhou; Zhiguang Qin; Jinde Liu Honeypot: a supplemented active defense system for network security Network security Proceedings of Fourth International Conference on Parallel & Distributed Computing Applications and Technologies, 27-29 Aug. 2003, PDCAT 2003 pp. 231-235
- [3] Scottberg, B., Yurcik, W., Doss, D. Internet honeypots: protection or entrapment? International Symposium on Technology and Society, ISTAS'02, 6-8 June 2002, pp. 387 - 391

- [4] Spitzner, L. Honeypots: Catching the insider threat Ninteenth Annual Computer Security Applications Conference, ACSAC 2003, pp. 170 - 179
- [5] Gupta, S.K.; Gupta A.; Goyal, V.; Sabharwal, S.;Patra, B.; Damor, R. Context honeypot : A framework for anticipatory privacy violation Submitted in Fourth International Conference on Bridging the Digital Divide, AACC 2006, Nepal, December 2006
- [6] Agrawal, R; Kiernan, J.; Srikant, R.; Yirong Xu Hippocratic databases Proceedings of the Twenty-eight International Conference on Very Large Bases, 2002, pp. 143-54
- [7] Goyal, V.; Gupta, S.K.; Meshram, I.; Gupta, A. PRINDA: Architecture and Design of Non-Disclosure Agreements in Privacy Policy Framework Data Engineering Workshops, 2006., Proceedings, 22nd International Conference, 03-07 April 2006, pp. 90 - 90
- [8] Goyal, V.; Gupta, S.K.; Saxena, S.; Chawla, S.;Gupta, A. Query Rewriting for Detection of Violation through Inferencing Privacy, Security and Trust 2006., International Conference on Privacy, Security and Trust 2006, 30 October-01 November 2006
- [9] Gupta, S. K.; Dubey, S.; Goyal, V. ; Gupta, A. A system to generate Privacy Log for testing Privacy Violation Detection System. National Conference on Innovation in Communication and Information Technology ICIT 2006, 7-8 July 2006, pp
- [10] Cranor,L.; Langheinrich, M.; , Marchiori, M. ; PresslerMarshall,M.; and Reagle, J. The platform for privacy preferences 1.0 (P3P1.0) specification W3C Recommendation, April 2002.
- [11] Ashley, P.; Hada, S.; Karjoth, G.; Powers, C.; Schunter, M. Enterprise privacy authorization language 1.2 (EPAL 1.2). W3C Member Submission, November 2003.
- [12] Provos, N. Honeyd: A Virtual Honeypot Framework Freely Available software; downloaded from : <http://www.citi.umich.edu/u/provos/honeyd/>