Toward Forensics Aware Web Design: Case Study: Low Fare Local Hajj Booking Web Services

Khalid A. A. Al-Shalfan
Al-Imam Muhammed Ibn Saud Islamic University
Kashalfan@Imamu.edu.sa

ABSTRACT

Muslims, who would like to subscribe in the Hajj (Islamic pilgrimage to Mecca) have to register in the Hajj Ministry or agency in their countries, in order to book Hajj permit for organization and security reasons. The Saudi government, organizes the registration of Hajj via a set of web services that enables the registration on the Low Fare Local Hajj (LFLH) program. Hajj registration system includes several observations and errors related to forensics requirements preservation especially the identity and privacy and therefore, if an electronic evidence is required for investigation matter, it will not be admissible or at least admissible with low proven power. In this paper, we apply Fi4SOA framework to the LFLH real motivating example. First, in design phase, we depicts and extract forensics and business requirements of LFLH example. In addition, we establish the SABSA matrix including all requirements, strategic, and physical operations. Secondly, in run time phase, we translate some LFLH rules and events into TESLA rules and events and shows how to infer them and detect any forensics or business malfeasance.

KEYWORDS

Hajj booking, web services, forensics and business requirements, Digital forensics.

1 LOW FARE HAJJ RESERVATION

The Hajj is an Islamic pilgrimage to Mecca (in Saudi Arabia) and the largest gathering of Muslim people every year. It is one of the five pillars of Islam, and a religious duty which must be carried out at least once in a life time by every Muslim who is physically and financially capable of undertaking the journey, and can support his family during his absence.
A Muslim who would like to subscribe in the Hajj has to register in the Hajj Ministry or agency in their countries in order to book Hajj permit for organization and security reasons.
Citizens of KSA as well must register to Hajj in order to book Hajj permit. The government organizes the registration of Hajj via a set of web services that enables the registration on the Low Fare Local Hajj (LFLH) program. The LFLH is a program launched by the KSA Hajj ministry characterized by its low Hajj cost. The registration capacity in LFLH in the year of 2014 is about 41 thousands pilgrims [1] in a country of more than 28 million of citizens.
The LFLH web service is launched before a month from Hajj days. Since the capacity is limited, all available places are booked within the first few days. The Figure 1 shows the structure of LFLH web services.
The booking of Hajj permit in LFLH begins by logging into the local hajj web site [2]. The user has to choose between requesting new booking, register confirmation payment after the booking acceptance, or cancel the booking reservation in the case of withdraw.

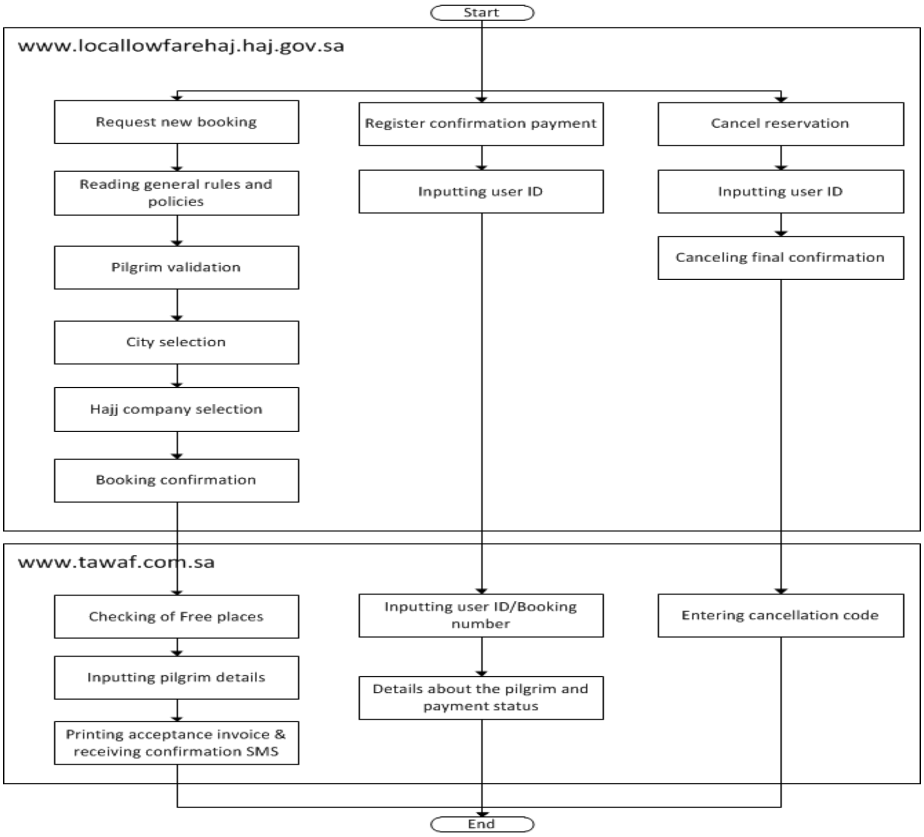
Start
www.locallowfarehaj.haj.gov.sa
Request new booking
Reading general rules and policies
Pilgrim validation
City selection
Hajj company selection
Booking confirmation
Register confirmation payment
Inputting user ID
Cancel reservation
Inputting user ID
Canceling final confirmation
www.tawaf.com.sa
Checking of Free places
Inputting pilgrim details
Printing acceptance invoice & receiving confirmation SMS
Inputting user ID/Booking number
Details about the pilgrim and payment status
Entering cancellation code
End

Figure 1. Low Fare Local Hajj web services Structure.

Requesting new booking passes through reading and confirming the general rules and policies of Hajj including who are authorized and the quality of services as well as the costs. The next step, the user has to input his identity number to verify his right of the hajj. Since each citizen in KSA has the Hajj right one time each five years. Also the government requires other conditions and policies which are:

- Not be less than the age of the applicant (15) fifteen years.
- Women must be logged in demand with her Muharram.
- That does not exceed the number required booking them during a single visit to (10) members up to a maximum.

In addition, the agreement includes the service program of LFLH and the price set of the program according to categories of camps.

One time the pilgrim satisfies the registration conditions, he/she is asked to choose the living city in order to search only on Hajj Company in that city. Then the available Hajj Company is listed including the cost and the offered category of each one. The user chooses the one satisfying his ambitions and confirms the initial booking. Besides, the hajj portal web site redirect the user to the hajj company web page in common site for all companies called tawaf.com.sa [3]. The company first inform the user about available places exists yet. The user is asked to deliver required information (name, birth date, ID number, phone number). Finally an acceptance SMS (message) and electronic primary confirmation invoice is received by the user.

The user has to pay the amount to the hajj company within 48 hours, if not the registration is canceled automatically. After payment, the user inputs payment details in the hajj company system and enable him to pursuit the payment status. The hajj company sends a payment confirmation SMS if the user is satisfied and agrees with everything. One time the registered pilgrim wants to decline his registration, the Hajj portal system enables his to cancel his reservation by inputting his ID number and confirm the cancelation by writing the SMS received code in the dedicated filed in the hajj company system.

The described Hajj registration system includes several observations and errors related to forensics requirements preservation, especially the identity and privacy and therefore, if an electronic evidence is required for investigation matter, it will not be admissible or at least with low proven power.

Registration scenarios that breaches forensics requirements are described below.

Scenario 1: Suppose that a user1 made a mistake when inputting his ID number, in hajj portal and the hajj portal by coincidence find that the ID number satisfies the registration conditions and therefore redirect him to hajj company system. The user then inputs other details and confirm the registration.

In this case, the user 1 will not be able to get the hajj permit since his given information are contradictory. The one having the right entered ID number will not be able to register since he/she is registered in the system by the first user. In both cases, they will lose the hajj permit especially because the number of places are limited and millions of people are reserving in the same time.

Scenario 2: Now let us suppose that the user1 intentionally inputs an ID number for user2 that he knows it is the wrong information in hajj company system. The user1 wants just to block user2 from registering and therefore missing the low cost hajj permit opportunity. In this case, the user2 must wait 48 hours in order to book again and in that time almost all available places are reserved. The hajj system does not provide any procedures to track and catch user1.

Scenario 3: The payment tracking feature in hajj system enables the users to pursuit their reservation status, by inputting their ID number in hajj system and automatically the user is redirected to hajj company system. The hajj company system presents the payment and reservation status related to the inputted ID number. In this page, the user can change the ID number and therefore he/she is able to see the reservation status, private information (name, birth date, phone number…) of other users, and if that user registered or not in the low cost hajj program.

In scenario 1, the identity of the user is breached accidentally and no procedures are provided to detect this mistake. This wrong identity is used after in the accomplishment of the remained steps successfully. The wrong ID number is detected only when giving hard copies to the Hajj Company. The scenario 2 poses the problematic of not taking into account the forensics matter in the Hajj system in order to detect and track suspect users. While in scenario 3, the privacy of users is breached and can be accessible for other users. In the above cases, a forensics layer that integrates built in forensics features in services in design time and monitors the transactions, detects any forensics breaches, and enables the tracking of the suspect is a primordial necessity.

2 Fi4SOA FRAMEWORK OVERVIEW

In previous work of our research team, we proposed a Fi4SOA framework [4]. It aims to forensically sound handle data and automatically find, detect, and track forensics or business breaches. In this section, we recall the main phases encompassed by Fi4SOA framework.

The first phase called design time forensics and business requirements integration. It enables the extraction of forensics/business requirements, the establishment of rules and drivers to preserve forensics/business requirements, and practical strategies and recommendations to integrate them to the targeted application. To this end, we used and adapt Sherwood Applied Business Security Architecture (SABSA) security methodology to extract, establish, and integrate forensics/business requirements. Through SABSA detailed layers, we easily determined forensics and business proprieties and the manner to apply them without conflicts or decreasing application quality of service. In addition, it provides a set of rules and recommendation related to the case for each involved party (hardware, software, or humans) in order to forensically sound process data.

The second Fi4SOA framework phase is named run-time events monitoring. In this phase, based on the forensics and business rules, we monitor events (logs, transactions, etc) and detect any forensics or business malfeasance. When time a threat is detected, the system alerts the administrator and logs all related information and data to the incident. For this sake, we uses Tesla as an event specification language that enables the events and rules formalization. In addition, it provides mechanisms to detect events that matches predefined rules.

Through Fi4SOA framework, we firstly prepare and improve the forensics readiness of the targeted application. Thus, at any time, we can get forensically sound information about an incident or any event. Secondly, the proposed framework provides a real time events monitoring which detects in an early stage any forensics or business malfeasance and therefore increasing the intervention and responses flexibility.

3 APPLICATION OF Fi4SOA TO LFLH

In this section, we apply Fi4SOA framework to the LFLH real motivating example. First in design phase, we depicts and extract forensics and business requirements of LFLH example. In addition we establish the SABSA matrix including all requirements, strategic, and physical operations. Secondly in run time phase, we translate some LFLH rules and events into TESLA rules and events and shows how to infer them and detect any forensics or business malfeasance.

3.1 Design phase

SABSA [5] is an open standard methodology aiming to design and develop a risk-driven information security architecture for enterprises. It provides a set of guidelines and solution supporting critical business initiatives. SABSA methodology consist of 6x6 matrix. Vertically, SBSA matrix comprises six layer, which are contextual security architecture, conceptual security architecture, logical security architecture, physical security architecture, component security architecture, and operational security architecture. These layers represents respectively the business view, architecture view, designer view, builder's view, trademark's view, and facility manager's view (details are not mentioned here due to space limitation and can be found in [5]). Horizontally, SABSA uses six questions "what, why, how, who, where, and when" to analyze six layers in detail. We applied in Aman System Research Team[1] SABSA methodology to digital forensics within web services based infrastructure in order to determine forensics and business requirements without conflicts. In the assets, we determine the different forensics and business attributes to be protected and preserved. The motivations provides guidelines related to the case for maintaining and achieving business and forensics goals. The SABSA process attribute identifies formal and technical solution for the encountered problems. The SABSA people, location, and time determines respectively the involved parties in the business or forensics related matters, their location, and time of execution or availability.

Back to the LFLH example, the table 1 depicts the related rules and recommendations required for forensics and business issues.

Table 1. LFLH Related SABSA Matrix.

Item	ASSETS (What)	Motivation (Why)	Process (How)	PEOPLE (Who)	LOCATION (Where)	TIME (When)
Contextual layer	-Ensuring the preservation and the readiness of forensics	-Ensuring the preservation and the readiness of forensics features and	Framework of operational processing for digital investigation	KSA citizens and	Only KSA citizens	The service is

	features and proprieties in all business area. -Defining internal and external policies that determines the working procedures of any part. - Ensuring the authentication of any user and attributing a role and responsibility to each one. - Protecting the business environment. - Conducting investigation against any violation of enterprise policy.	proprieties in all business area. - Defining internal and external policies that determines the working procedures of any part. - Ensuring the authentication of any user and attributing a role and responsibility to each one. - Protecting the business environment. - Conducting investigation against any violation of enterprise policy.		subscribed Hajj company employees.	are allowed to subscribe in the service.	available only for the month before the Hajj days.
Conceptual layer	-Auditing and evaluating the investigation is a high-level requirement especially in SOA due to the complex, dynamic, distributed, and heterogeneous nature of the interconnected services, which increase the possibility for making errors and collecting evidence in wrong way. Investigation in an SOA environment requires high-level qualifications and skills, which make the process of monitoring and documenting any investigation activities very important. - Having the same test condition and same results is almost under impossible due to the changing, complex nature of an SOA environment. However, the repeatability and reproducibility can be seen as having the same result appearance under close conditions. So that, the results will be considered as repeatable and reproducible if they are accurate (having the same appearance) even if the test conditions are changing.	-The authentication is LFLH web site is based on only the identity number. The identity number is not private and can be known by several persons (explained in scenario 1 and 2). In order to ensure that the registered person is the one having the inputted identity, the authentication should uses a combination of parameters. For example, identity number, birthday, and password. Another solution is the use of fingerprints or face recognition especially technologies related to them are widely used. - One time a user is logged to the Hajj company site using the Hajj ministry web site, he/she can change the identity number and gather private information of other registered users (explained in scenario 3), which is a privacy breach. To deal with this issue, a simple solution is to disable any request of information through the Hajj company web site and present only information related to the inputted identity number in the Hajj ministry site.	Forensics process according to standards	Forensics responsibilities assignment matrix		
Logical layer	The assets includes the SLA, all conducted transactions by the service, the different persons, and tools in contact with the data.	Organizational forensics policy	Forensics service	Forensics trusted framework	Forensics policy authority domain map	
Physical layer		-Inform the user about privacy policies. - Minimize the handling and corruption of original	In the LFLH motivating example, to ensure that the registered person can get access only to their	Role based access control		

		data. - Do not exceed your knowledge. - Write any changes you make to data.	private information, we can disable any query sent from the hajj company web page and warn the users by using information banners for example that any tentative of accessing other users private information is prohibited and cause the judicial follow-up. Also any action taken by the user should be logged and saved in a secure location.
Component layer			
Operational layer			Chain of Custody

3.2 Run time phase

In this section, we only apply TESLA reasoning to some LFLH forensics and business rules and requirements and show how to detect an integrity breach incident.

Event set

Event set includes all user requests, service response, messages, security alerts, and transactions during the service composition. Events are extracted from these resources and for each event; we identify all related information such as event type, record, event task, and contact.

For instance, the Figure 2 shows a soap message request that has a message body representing a method call at service, preceded by optional WS-Addressing headers that provide the URIs of the target service and action and a unique message identifier.

```
<S:Envelope          xmlns:S="http://www.w3.org/2003/05/soap-
envelope"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressin
g">
  <S:Header>
   <wsa:MessageID>
    uuid:6B29FC40-CA47-1067-B31D-00DD010662DA
   </wsa:MessageID>

<wsa:To>http://interiorministry.gov.sa/identityservice</wsa:To
>
   <wsa:Action>http://                       interiorministry.gov.sa
/SubmitId</wsa:Action>
  </S:Header>
  <S:Body>
<GetConfirmation>62518396</GetConfirmation>
  </S:Body>
 </S:Envelope>
```

Figure 2. Soap Message Request.

This message is translated into Tesla event notification as follow:

name *SoapRequestMessage*
number *to =* "http://interiorministry. gov.sa /identityservice" *and action=* "http:// interiorministry.gov.sa /SubmitId" *and GetConfrimation="*62518396*" and MessageID="*6B29FC40-CA47-1067-B31D-00DD010662DA*"*
type *saopmessage*
order nothing
timeStamp "2014-09-24 00:35:35"

Application domain rules

The application domain rules component essentially gathers the policies and rules from the Service Level Agreement (SLA) and web service description (.wsdl). This component translates the SLA and web service description into rules and policies that organizes the relation between different partners and determines the duties and rights of each one. For instance, the government in the LFLH example stipulated four conditions in order to register in the service. Following are the condition and their Tesla rules representation:

- Must not have been previously demanded Booking pilgrimage during the past five years

define BeneathPeriod (periode:"val")
from Hajj(year=$y) and period ($currentYear-$y<5)
where val=$currentYear-$y

- Not be less than the age of the applicant (15) fifteen years.

define BeneathAge (age:"val")
from Age($age>=15)
where val=$age

- For women must be logged in demand with her Muharram.

```
define      NoMuharram ()
from        gender($gender=”women”) and muharram($exist=”no”)
```

- That does not exceed the number required booking them during a single visit to (10) members up to a maximum.

```
define      OverBookingNumber(number:”val”)
from        bookingvisit() and each (bookingnumber >10) within 1 visit from bookingvisit()
where       val=bookingnumber
```

Returning to the LFLH example, we suppose the following rules accompanied with Tesla formulation for each one:

- Inputted data (ID number, gender, name, birthday, and phone number) are only used for the registration purpose by the hajj portal and hajj company web sites.

```
define      privacyDataDeliveryBreaches(service:”val”)
from        registration() and usedIn(serviceId=$x)
            and ($x<> www.tawaf.com.sa) or ($x<> www.locallowfare.haj.gov.sa)
where       val=serviceId
```

The serviceId contains the current service domain name using actually the inputted data.

- Inputted data must be removed in the case of registration cancelation.

```
define      privacyDataRemoveBreaches(service:”val”, data:”val1”)
from        registrationCancelation() and (serviceID=$x)
            and notRemove(data=”dataname”) within ts from registrationCancelation()
where       val=serviceId and val1=dataname
```

The ts represents the maximum time in seconds that a data should be removed after a registration cancelation action.

- Inputted data for confirmed registration must be only retained by the hajj portal and not by the hajj company after 10 days from the hajj accomplishment.

```
define      privacyDataRetentionBreaches(service:”val”, data=”val1”)
from        hajjAccomplished() and (serviceID= “www.tawaf.com.sa”)
            and notRemoveData(data=”dataname”)
            within 10days from hajjAccomplished()
where       val=serviceId and val1=dataname
```

Digital forensics properties

This component deals with the formulation of general and essential digital forensics proprieties enabling the admissibility of the gathered evidence with high proven power. It is not related to special application domain since all digital area shares the same admissibility requirements and only the manner of how to proceed or collect data differs. The forensics policies related to the application domain are determined in the application domain rules components, and this component (digital forensics proprieties) looks only on the relation between proprieties and if there is missing or not considered requirements when handling data. Digital forensics component contains four classes. The forensicProperty class represents the smallest property defining specific requirement. For example, the identity of a record (or evidence) is a specific requirement of the class type authenticity (see Figure 1). To validate a record, it must satisfy all authenticity requirements which are the identity, the integrity, and the authentication. The second class is classType containing all essential forensics requirements type such as the authenticity, reliability, privacy, comprehensiveness, and etc. Some of them are related to the admissibility requirement and the other to the weight of the gathered evidence which represents the third class category. The last class severity is used to determine the forensics property priority and relevance. It focuses essentially in attributing to each property or relation a value showing its severity in the case of property losing. For instance, let’s consider the formulation of the authenticity requirement which encompasses three properties (identity, integrity, and authentication). The identity requires the definition of the different attributes that characterizes the evidence such as date and time of issuing, creation time, author, addressee, subject, and etc. The identity is converted to Tesla as follow:

```
define identityBreaches
(MissingIdentityAttribute=”attr”, record:”ref”)
from record(revordref) and
notsignature(val) or notsiynedby(val)
or
nothascreationtime(val) or nothassubject(val)
where attr=val, ref=recordref
```

The integrity aims to preserve the original data and to keep saved copy from alterations in order to avoid any court data contaminations. Collecting data should adapt methods of data integrity during the storage and transmission to avoid alteration and to maintain its authenticity. So that, the integrity aims to ensure that collected data are protected, not being tampered and complete using hash or cryptographic techniques. All related actions to the record must be logged and preserved such as the names of all persons handling and responsible for the record keeping over time, all technical modifications, annotations, and all actions and policies related to the data retention, disposition and transfer. Thus, the integrity can be verified and therefore enhancing the record trustworthiness. The integrity is converted to Tesla as follow:

```
define integrityBreaches
(MissingIntegrityAttribute=”attr”, record:”ref”)
from record(recordref) and
nothashashvalue(value) or isAltered(val)
or notSiynedby(val) or
nothascreationtime(val) or nothassubject(val)
where attr=val, ref=recordref
```

The authentication aims to allow only authorized persons or software to get access to a document and serve to proof the document authenticity in one particular moment. The authentication is converted to Tesla as follow:

```
define authenticationBreaches
(MissingAuthenticationAttribute=”attr”,
record:”ref”)
from record(recordref) and
isNotauthorized(value) or isNotSkilledin(val)
or isNotCompetentin(val )
where attr=val, ref=recordref
```

Finally, the authenticity that requires the proof of the identity, integrity, and authentication is formulated using Tesla as follow:

```
define AuthenticyBreaches
(MissingAuthenticityAttribute=”attr”, record:”ref”)
from record(val) and
integrityBreaches(val) or identityBreaches(val)
or
authenticationBreaches(val)
where attr=val, ref=recordref
```

The rest of the digital forensics proprieties formalized as the authenticity property forming the essential policies and rules that monitors the system forensics soundness. The events notification and rules must share the same formulization in order to avoid any incoherence between them.

TESLA reasoning system

The system reasoner infers rules and events in order to detect any matched pattern that generates notifications. The system uses domain application and digital forensics proprieties rules and polices forming the knowledge database together with rules and requirement provided by the user or the investigators whom subscribes to the events. The system reasoner contains, aside from knowledge database, a predefined rules describing eventual attacks or forensics breaches scenario extracted from the generated event notification history or provided by experts. The rules include patterns of forensics violation within specified period or ordered sequences of events. It consider also new subscribed rules provided by the end users (service requester, forensics persons, service provider,…) in the aim to inquiry about some events. Subscription to events is very easy using Tesla, as explained in section 3, offering users high flexibility to achieve goals in real or near real time without the need to define previously new rules that contains their request. Following, we portrait some instance of events and rules and how the system reasoner infers them based on the LFLH motivating example.

Let us consider the following event scenario; a user inputs his ID number to the hajj portal system in order to register in the hajj. The system sends the identity to second private service linked with the interior ministry databases in order to verify its permit right for the current year using soap request message. Then the interior ministry replies to the request by sending response soap message containing the request result. This transaction triggers in background the following actions:

The above event history contains six events starting by inputting ID number and finished by verifying it by the interior ministry web service. As forensics requirement, this transaction must firstly inform the user about the parties that use his information, and then during the sending and the receiving of the information by the services. The data must be protected against any violation or tampering in order to keep their integrity, privacy and confidentiality. Thereafter this transaction must be saved and stored in the services database by accurate software, processes, and authenticated skilled persons.
Each of the six events generates the following Tesla event notification:

at t=0:
name *InputingIDNumber*
number *service* = “hajj portal”
and action= “DataInputting”
type *userInputingData*
order nothing
timeStamp “2014-09-24 00:35:35”
at t=1:
name *SoapRequestMessage*
number *to* = “http://interiorministry.gov.sa/identityservice”
and action= “http:// interiorministry.gov.sa /SubmitId”
*and GetConfrimation=”*IDnumber*”*
*and MessageID=”*6B29FC40-CA47-1067-B31D-00DD010662DA*”*
and Hashvalue=”GUODWDS54SDFF98FSS53SFIUID36FS”
type *saopmessageCall*
order nothing
timeStamp “2014-09-24 00:35:40”
at t=2:
name *SoapRequestMessageReceiving*
number *from* = “hajj portal”
*and MessageID=”*6B29FC40-CA47-1067-B31D-00DD010662DA*”*
and VerifyHashvalue=”GUODWDS54SDFF98FSS53SFIUID36FS”
type *saopmessageRecievingandVerification*
order nothing
timeStamp “2014-09-24 00:35:55”
at t=3:
name *VerifyIDnumber*
number *GetConfrimation=”*IDnumber*”*
*and MessageID=”*6B29FC40-CA47-1067-B31D-00DD010662DA*”*
type *VerifIDnumber*
order nothing
timeStamp “2014-09-24 00:36:00”
at t=4:
name *SoapRequestMessageResponse*
number *relatedto = “MessageID:”*6B29FC40-CA47-1067-B31D-00DD010662DA*”*”
and confirmationresponse=”true”
type *saopmessageresponse*
order nothing
timeStamp “2014-09-24 00:36:10”
at t=5:
name *ConfirmorRejectIDnumber*
number *confirmationresponse=”true”*
type *ConfirmorRejectIDnumber*
order nothing
timeStamp “2014-09-24 00:36:12”

And let’s consider that the knowledge database rule contains the following Integrity checking rule:

define integrityNotChecked (messageId=”id”, serviceId:”ref”,missingAttribute=val)
from SoapMessage(id) as SM
or not generatehashvalue(val) as HM within 1s from SM Soap
or not Siynedby(serviceId) as Sby within 1s from SM Soap
or not hascreationtime(val) as CT within 1s from SM Soap
or not hassubject(val) as S within 1s from SM Soap
where messageId=id, ref=serviceId,missingAttribute=val

The above rule integritybreachs includes only two successive conditions, the first soapMessage() aims to define the type of message and the second is one of the integrity requirements attributes (hash value(), signed by(), creation time(), and subject()) since missing one attribute decreases the integrity. Figure 3 depicts the different event detection models for the rule integrityBreaches:

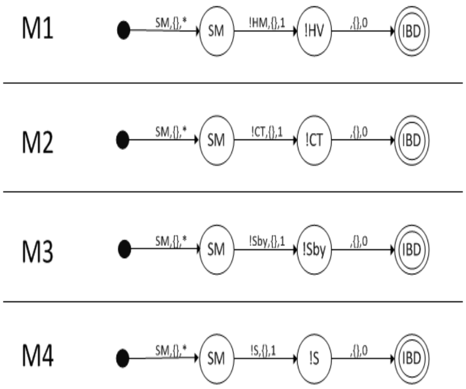

Figure 3. Event Detection Models for The Rule integrityBreaches.

Each transition from state s1 to state s2 is labeled with the set of constraints that an incoming event of type s2 has to satisfy to trigger the transition plus the maximum time for the transition to be triggered. In Figure 3 all models shares the first state SM since they all related to the soap message request event, then any missing attribute within 1 seconds from the creation of soap message request triggers the notification about integrity requirements breaches.

Now, in order to describe the behavior of the event detection automata, we consider only the sequence of events captured by model M1 of the rule integrityBreaches for simplicity reasons.

Table 2. An Example of Event History Occurrences.

T=0	T=1		T=2		T=3	T=4	T=5
Inputting ID number	Generates call soap message	Add integrity hashvalue	Receive soap message	Verify the integrity	Verify the ID number	Send soap response message	Confirm or reject the ID number

Briefly the event processing starts by creating a single instance for each of these automata. Then for each incoming event, it creates new automata instances, or moves existing ones from state to state, or deletes some of them. For instance, the automaton instance M1 is a state SM reacts to the detection of an event e that satisfies the constraints for the transition exiting SM (which is in our example the non-generation of the message hash value within 1 seconds from SM), by firstly duplicating itself, creating a new instance M11, then using e to move M11 to the next state (while M1 remains in the state SM). Those events that do not satisfy currently enabled transition are simply ignored, while automata instances are deleted if they are unable to progress within the maximum time associated to each transition. The integrityBreaches rule is triggered when an instance of the corresponding automaton model arrives to its accepting state represented with double circle (see Figure 4).

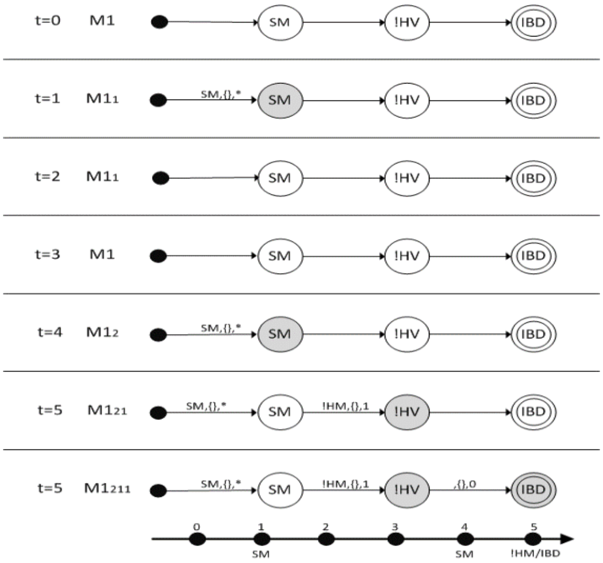

Figure 4. Event Processing Example of The Rule integrityBreaches.

4 CONCLUSION

In this paper, we treated the case of an example of web service non-conformity with forensics requirement using accurate modeling technique, which is SABSA. Whilst using 'SABSA', the implementation malfeasance and suggesting several recommendation to solve the predetermined issues. The importance of this paper reside in its modeling of local, very critical and important web services in Saudi Arabia. Which is Low Fare Local Hajj Booking Web Services. Furthermore, we investigated the rules and recommendation extracted using SABSA into TESLA reasoning language in order to detect and respond to any forensics issues in execution time. We provide an example of TESLA events detection of one forensics property related to the Low Fare Local Hajj Booking Web Services. However in the near future if this research was to occur again often methods and the defined rules would be implemented and integrate them in real web services.

ACKNOWLEDGMENTS

This paper is a partial result of a research project granted by King Abdul Aziz City for Sciences and Technology (KACST), Riyadh, Kingdom of Saudi Arabia, under grant number 11-INF1787-08. We thank AMAN System team (www.amansystem.com) for their help and recommendations.

REFERENCES

[1] H. m. o. KSA, 2014. [Online]. Available: http://www.haj.gov.sa/ar-sa/SitePages/News-Detail.aspx?newsid=1671.

[2] H. M. o. KSA, 2014. [Online]. Available: http://www.locallowfare.haj.gov.sa/LF/home2.xhtml.

[3] H. c. page, 2014. [Online]. Available: http://www.tawaf.com.sa.

[4] G. Cugola and A. Margara, "TESLA: A Formally Defined Event Specification Language," in *Proceedings of the Fourth ACM International Conference on Distributed Event-Based Systems*, Cambridge, United Kingdom, 2010.

[5] A. C. D. L. Johen Sherwood, Enterprise Security Architecture: A Business-Driven Approach, SanFransisco: amazon.com, 2005.