SECURE COLLABORATIVE PROCESSING ARCHITECTURE FOR MITB ATTACK DETECTION

Hani Qusa and Shadi Abudalfa

Information Technology Department, University College of Applied Science, Palestine

ABSTRACT

In this paper, we take a distributed architecture called Semantic Room (SR) which is capable of correlating events coming from several organizations participating in the SR, developed in the context of the EU Project COMIFIN, and we add privacy capability to the SR.. The SR architecture consists of Edge Gateways deployed at each financial institution and a set of private clouds that form the SR collaborative processing system (CSP). Edge Gateways perform data pre-processing and anonymize data items, as prescribed by the SR contract, using Shamir secret sharing scheme. Anonymous data are sent to the CPS that aggregates information through MapReduce-based computations. The anonymous data resulting from the collaborative computation are revealed to the financial institutions only if suspicious cyber threat activities are detected. In this paper we show how this SR can be leveraged for detecting Man-In-The-Browser attacks.

KEYWORDS

Collaborative environments, MapReduce, Privacy, Secret sharing, Man-in-the-Browser, Critical infrastructures.

1. INTRODUCTION

The increased "webification" of critical infrastructure services such as home banking, remote payments, etc., on one hand improves the availability and user-friendliness of those services, on the other hand however exposes them to a variety of Internet-based sophisticated security attacks. The complex proprietary infrastructures that "no hackers knew" are now replaced by common used systems where vulnerabilities are documented and well-known. Although the sophistication of cyber attacks has increased over time, the technical knowledge required to exploit existing vulnerabilities is decreasing [13].

Attacking tools are often fully automated and the technology employed in many attacks is simple to use, inexpensive and widely available. Because of the increased sophistication of computer attack tools, a higher number of actors are capable of launching effective attacks against IT critical infrastructures; that is, the range of possible attackers is not limited to skilled crackers anymore; criminal organizations, terrorist groups using the Internet for making money or causing any types of damages are augmenting, with serious and measurable consequences for any critical infrastructure: lost revenue, own time, damage to the reputation, damage to IT assets, theft of proprietary data or customer sensitive information [4].

* This work was done while the corresponding author was accomplishing his Ph.D study in University of Rome “La Sapienza”, Rome Italy.

DOI : 10.5121/ijnsa.2013.5507

In general, these coordinated attacks are extremely difficult to detect by single critical infrastructures in isolation, since their evidence is deliberately scattered across different administrative domains. Thus, there is the urgent need to combine and correlate a large volume of data from multiple distributed sites in order to have a more comprehensive view of malicious activities that may occur and that would have gone undetected if considered in isolation.
A number of research works are devoted to the study of collaborative systems for detecting massive large scale security threats [18, 17, 16]; however, possibly competitor distrusting organizations (e.g., different financial institutions) can be reluctant to fully adopt such collaborative approach as this imply sharing information that can be potentially sensitive. In these competitive contexts, a trusted and controllable environment should be provided, which provides specific guarantees that a number of requirements on the management of the data are continuously met.

In this paper we describe a collaborative abstraction named Semantic Room. Semantic Room enables the construction of collaborative and contractually regulated environments where distributed event aggregation and correlation on data provided by organizations participating in the Semantic Room can be carried out with the aim of monitoring widely distributed infrastructures and providing early detection of attacks, frauds and threats. Each Semantic Room has a specific strategic objective to meet (e.g., botnet detection, Man-in-the-Browser) and is associated with both a contract, which specifies the set of rights and obligations for governing the Semantic Room membership, and different software technologies that are used to carry out the data processing and sharing within the Semantic Room. Thanks to this latter characterizing element of a Semantic Room, we were able to instantiate different processing and sharing systems within it: the systems aimed at detecting a variety of cyber attacks such as inter-domain stealthy port scan [1], MiTM, Botnet-driven HTTP.

Session hijacking [2] also showing the effectiveness of the collaborative approach as reported in [1], [2] and [3]. However, none of these instances consider privacy issues in the data sharing and correlation: data flow into the SR in clear and the result of the computation is finally distributed to the SR participants revealing which participant contributed to the discovery of the attacks. This disclosure could contrast either with national privacy-related legislation or with privacy clauses included in contracts that SR participants can sign with their customers and with those included in the SR contract. This can be a clear disincentive in the diffusion of semantic rooms among financial players.

Regarding this observation, we propose the design a novel SR instance that employs a distributed software architecture through which distrusting SR participants can execute collaborative computations on massive sets of input raw events (e.g., network packet traffic, financial transactions) while preserving the privacy of sensitive data. Such data can be revealed if and only if a set of specific conditions are verified corresponding to some contract infringement. The architecture includes edge gateways, co-located at each SR participant, executing a first level of aggregation and filtering, and a collaborative processing system formed by a set of private clouds that correlate events in a batch way. Our solution introduces a novel combination of MapReduce-based distributed parallel processing that supports massive data processing and Shamir's secret sharing schema that guarantees the privacy of sensitive data (as in [10],[9]). Once an event has been aggregated by the gateway, the latter generates the shares of the secret embedded in the event and disseminates shares to the set of private clouds that store these shares in large scale storage (i.e., HDFS [6]). Once a batch has been completed a MapReduce [12] computation starts at each public cloud; this computation is instrumented by one or more complex queries, the same at each cloud, that are specified in a high level similar to SQL named HIVE [7]. Each of these

queries looks for anomalies, specific dangerous patterns, signatures etc. of a particular cyber attack or fraud. The shares that satisfy the query are ordered and passed, one at a time, to a reconstruction unit that collects necessary shares of a data item from other clouds. When a sufficient number of shares is received, the cloud is able to reconstruct the secret. In this paper we show how this SR can be leveraged for detecting Man-In-The-Browser attacks.

The rest of the paper is organized as follows. Section 2 presents the Semantic Room abstraction, Section 3 introduces an architecture for a privacy preserving semantic room. Section 4 details a privacy-preserving room architectures to detect Man-in-The-Browser attacks. Section 5 describes some related works. Finally, section 6 concludes the paper.

2. THE SEMANTIC ROOM ABSTRACTION

We consider a European Union (EU) project called Collaborative middleware for Monitoring Financial Critical Infrastructure abbreviated as CoMiFin [19]. The CoMiFin project provides the service model for a federation of financial institutions formed in so called Semantic Room (SR) for the sake of information processing and sharing. The financial institutions participating in a specific SR are referred to as the members of the SR.

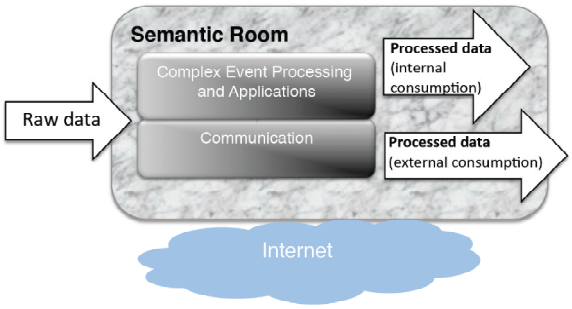

Fig. 1. The Semantic Room Abstraction

The SR abstraction is defined by the three following principal elements (the interested reader can refer to [19] for further information):

- *Contract:* each SR is regulated by a contract that defines the set of processing and data sharing services provided by the SR along with the data protection, privacy, isolation, trust, security, dependability, and performance requirements. The contract also contains the hardware and software requirements a member has to provision in order to be admitted into the SR;

- *Objective:* each SR has a specific strategic objective to meet. For instance, there can exist SRs created for implementing large-scale stealthy scans detection, or SRs created for detecting Man-In-The-Middle attacks;

- *Deployments:* each SR can be associated with different software technologies, thus enabling a variety of SR deployments. The SR abstraction is in fact highly flexible to accommodate the use of different technologies for the implementation of the processing and sharing within the SR (i.e., the implementation of the SR logic or functionality). In particular, the SR abstraction can support different types of system approaches to the processing and sharing; namely, a centralized approach that employs a central server (e.g., Esper [5]), a decentralized approach where the processing load is spread over all the SR members (e.g., a DHT-based processing), or a hierarchical approach where a pre-processing is carried out by the SR members and a

selected processed information is then passed to the next layer of the hierarchy for further computations.

As shown in this fig. 1, the SR abstraction supports the deployment of two components termed Complex Event Processing and Applications, and Communication which can vary from SR to SR depending on the software technologies employed to implement the SR processing and sharing logic, and a set of management components, that together form the marble rectangle in Figure 1 and that are exploited for SR management purposes (e.g., management of the membership, monitoring of the SR operations). SR members can inject raw data into the SR. Raw data may include real-time data, inputs from human beings, stored data (e.g., historical data), queries, and other types of dynamic and/or static content that are processed in order to produce complex processed data. Raw data are properly managed in order to satisfy privacy requirements that can be prescribed by the SR contract and that are crucial in order to effectively enable a collaborative environment among different, and potentially competitive, financial institutions.

Processed data can be used for internal consumption within the SR: in this case, derived events, models, profiles, blacklists, alerts and query results can be fed back into the SR so that the members can take advantage of the intelligence provided by the processing (Figure 1). SR members can, for instance, use these data to properly instruct their local security protection mechanisms in order to trigger informed and timely reactions independently of the SR management. In addition, a (possibly post-processed) subset of data can be offered for external consumption. SRs in fact can be willing to make available for external use their produced processed data. In this case, in addition to the SR members, there can exist clients of the SR that cannot contribute raw data directly to the SR but can simply consume the SR processed data for external consumption (Figure 1). SR members have full access to both the raw data members agreed to contribute to by contract, and the data being processed and thus output by the SR. Data processing and results dissemination are carried out by SR members based on obligations and restricts specified in the above mentioned contract.

3. SECURE SEMANTIC ROOM

Figure 2 illustrates a specific instance of an SR which is capable of preserving the privacy of sensitive data during collaborative events processing. The architecture consists of two main components: a so-called Edge Gateway, and a Collaborative Processing System (CPS). The edge gateway transforms raw data into events, here as CPS detects anomalous behaviors.

These components are used in three different phases of the data processing. The phases work as in a pipeline and are described next.

3.1 Pre-processing phase

The Edge Gateway is located at the service provider site. It is responsible for (i) protecting sensitive data items, as prescribed by contracts service providers established with their customers (see Figure 2), and (ii) injecting anonymized data to the Collaborative Processing System. We have designed the Edge Gateway component so as to embody two principal modules; namely, the privacy-enabled pre-processing and data dissemination modules.

Privacy-enabled pre-processing module.

In the privacy-enabled pre-processing module, raw data of service providers are pre-processed by filtering unnecessary data and/or aggregating data according to specific formats necessary for the successive private event processing phase (see Section 4 for details). In addition, aggregated data

are given to a privacy preserving algorithm which anonymizes sensible data according to specific contractual clauses.

The algorithm is based on the Shamir's *(k, n)* secret sharing scheme [15], which permits to shares a secret *s* among *n* participants in a way that the secret can be easily reconstructed if and only if any *k* out of the *n* participants make their shares available, where $k \leq n$ provides the strength of the scheme. This is achieved by generating a random polynomial *f* of degree *k - 1* defined over a prime field $\mathbf{Z}_p$, with $p > s$ and such that $f(0) = s$. The polynomial is used to generate n different shares, s_1; s_2,…, s_n, where $s_i = f(i)$. The vector of shares is denoted by *[s]*. The secret can be reconstructed exploiting the Lagrange interpolation technique.

Our architecture assumes that a certain number $w < n$ of participants are semi-honest; that is, they do follow the collaborative processing protocol, but they are \curious" and attempt to infer as much information as possible from the computation itself. In principle, semi-honest participants can even coordinate themselves in order to increase the chances of getting private data. In order to neutralize the semi-honest activities we set $k = w + 1$. The privacy preserving scheme embodied in the Edge Gateway first divides the aggregated data into two parts, a sensitive data part *s* and a non sensitive part *u*.

Shamir's scheme is then applied to *s* and the produced list of shares *[s]* is sent to the data dissemination module together with the *u* part.

Data Dissemination module.

This module is in charge of disseminating private data to all the participants in the form of events. The dissemination occurs periodically, i.e., every fix time window. The beginning and end of each period is demarcated through special signaling messages. The module sends elementary information in the form of a triple *(hash([s]); s_0; u),* where *hash()* is a perfect hash function, i.e., a function with no collisions. It is worth noting that the hash function takes all the shares as its argument: for any two secrets *s, t, hash([s])=hash([t])* iff *s=t*.

For the purpose of data ordering, the data dissemination module of service provider *i* manages a vector seq_{ij} of sequence numbers associated with the participant *j*, which is reset at the beginning of a new dissemination phase. The entry seq_{ij} represents the sequence number of the last pair sent by *i* to *j* and it is increased of one unit before each transmission.

Each triple *(hash[s], sj, u)* sent by the data dissemination module of service provider *i* to Participant *j* is tagged with the pair *(i, seq_{ij}).* We assume that all the communication channels are secure, FIFO and reliable. The tuple *(i, $seq_{ij,}$ hash[s], s_j, u)* defines an event.

3.2 Private processing phase

The Collaborative Processing System (CPS) is responsible for (i) collecting private data sent by Edge Gateways of service providers, (ii) properly aggregating and processing the private data and (iii) sending back a result of the computation in an unanonimyzed form to service providers.

The CPS can be thought of as a federation of private clouds. Each private cloud is owned by a service provider and deployed for the sake of collaborative complex data processing. A private cloud consists of the set of hardware and software resources made available by the provider. It communicates with other private clouds in the federation only during the reconstruction phase (see below) using secure channels. Within a private cloud two processing units can be identified, as shown in Figure 2: a private processing unit and a reconstruction unit.

Private processing unit. The goal of this unit is to aggregate and correlate private large datasets coming from the Edge Gateways and to notify anomalies to all the participants.

In our design, the j-th private processing unit receives events (i, seqij, hash[s], sj, u) from Edge Gateway i, i = 1,…, n. The private processing unit is constructed out of a distributed network of processing and storage elements hosted in the private cloud. As a share acts as a shadow of the original secret, any participant has all the necessary data to make correlations. The processing elements manipulate and aggregate those data as follows. The processing elements are components of the MapReduce framework [12]: a centralized Job Tracker coordinates the execution of mappers and reducers on each resource of the private cloud through a collection of Task Trackers deployed on those resources. Mappers and Reducers process the data according to a specific processing logic. We use a high level query language in order to define the processing logic. The language compiles into a series of MapReduce jobs. Specifically, the language supports SQL-like query constructs that can be combined into flow and specify the data patterns to be discovered on the set of input data. A query engine is in execution inside each private processing unit: the engine retrieves the data in the storage elements and aggregates them according to one or more SQL-like queries.

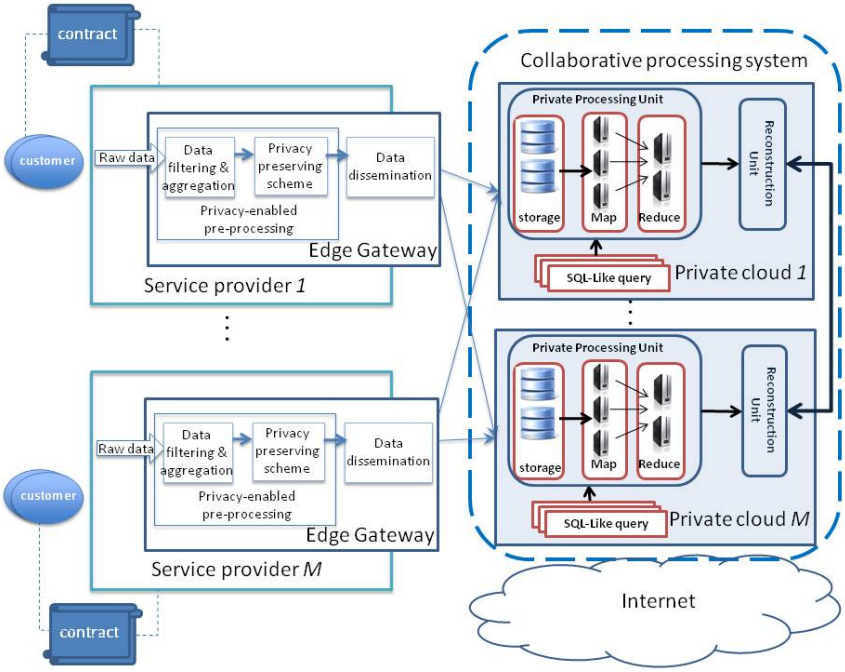

Fig. 2. Contract-based secure processing architecture

The final result from reducers is an ordered sequence of shares. The ordering is carried out exploiting the \order by" constructs made available by the majority of SQL-like languages for data processing (e.g., HIVE [7], JAQL [8]). An external protocol could be also used as an alternative; it first orders the shares in groups, according to the lowest id of the participant from which the shares were sent, and then inside a group it orders according to the sequence number of the shares.

Reconstruction unit.

The reconstruction unit is responsible for communicating with the other reconstruction units of the private clouds of the federation in order to rebuild the secret. Each reconstruction unit sends the output of the query, i.e., an ordered list of shares, and waits for receiving a similar list from all the other participants. Each unit then applies the Lagrange interpolation algorithm to reconstruct the original secret. The reconstruction algorithm is organized as sequence of reconstructions. The first interpolation is applied using the first share in the lists received from the participants, the second interpolation is applied using the shares in the second position, etc.

4. THE SR COLLABORATIVE MITB

4.1 Man-in-the-Browser attack description

Banks deploy web servers for running web-based online banking services. Web servers run outside the bank firewall (in DMZ) and are thus exposed to a number of security threats. In the financial context a relatively new type of attack is the Man-in-The-Browser (MiTB). It is a variant of Man-in-the-Middle capable of stealing login credentials, account numbers and various types of financial information. In MiTB attacks, licit bank customers use web banking services in order to carry out their own financial transactions. Attackers leverage malware such as Trojan Horses. These typically wake up when customers visit a target site, and function by transparently (a trojan horse does not interfere with the normal use of the browser) capturing and modifying information (e.g., customer's credentials) as they move between the browser's customer interface and the Internet. Unlike more traditional phishing methods that employ links in the body of emails to direct users to fake web sites and prompt them to enter secure data, the MiTB simply captures data as the user opens the browser. This personal data is then sent directly to an FTP site to be stored, where it is sold to the highest bidder. Users are completely unaware of their data is being hijacked, since they are actually interacting with a legitimate site. Examples of well-known MiTB attacks include Zeus and Silentbanker Trojans, which have been successfully installed on millions of PCs [11].

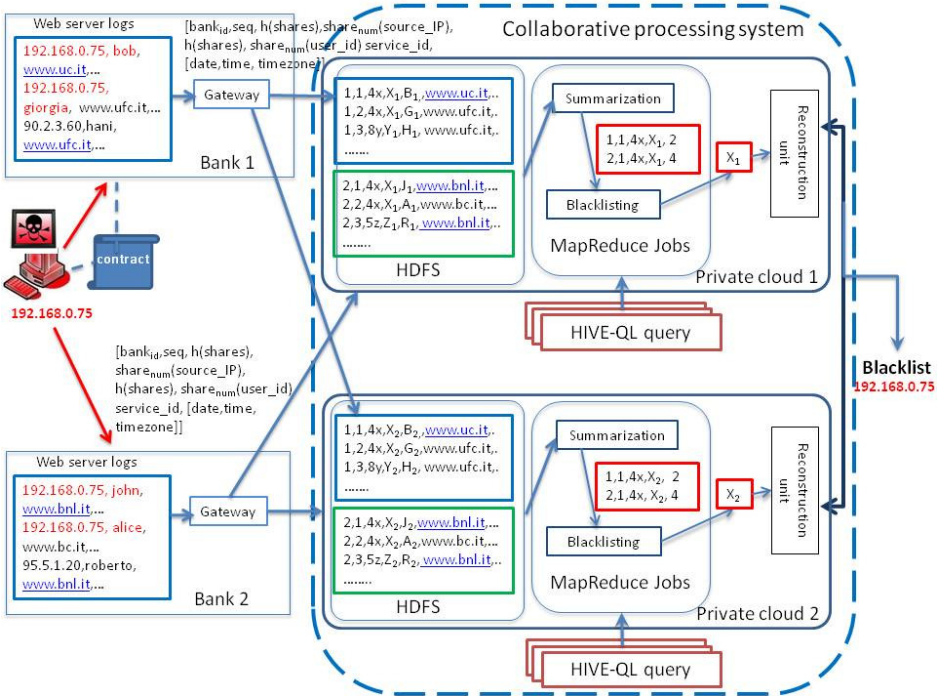

Fig. 3. Processing steps of the contract-based architecture for collaborative Man-in-The-Browser detection

4.2 The added value of collaboration in detecting MiTB

Detecting MiTB attacks is extremely difficult as they appear normal transaction between a server and a licit (already known and registered) customer, and traditional virus-scanning systems are not fully effective in recognizing the presence of these Trojans since they operate between the browser security protocols and input of the user. Standard security measures normally will not even reveal the presence of the MiTB virus.

To this end, we instantiated the SR architecture to discover such type of attack; that is, to discover if bank's licit servers are receiving a high number of connections on behalf of different customers from the same device. The underlying hypothesis here is that customers usually access their financial services, also provided by different banks, from a limited number of devices, and that the same device is not used by a large number of customers to access financial services in a short time period. The attack is then detected by identifying statistical anomalies related to the activities of a single source of network traffic that successfully logs into several services (such as home banking web sites) provided by different banks and using different identities (this might be an indication of stolen user credentials). Once identified, the source IP addresses are included in a blacklist.

4.3 MiTB SR: processing steps

At the first step, data coming from sources (i.e., web server logs) deployed at the network of each bank participating in the collaborative environment are collected and forwarded to the bank's Edge Gateway (see Figure 3).We have implemented the Edge Gateway component as a Java program that pre-processes the incoming data by filtering and aggregating them (for instance, we filter the http code of the request). These data are then anonymized through the secret sharing

privacy scheme. In our case study, we consider the example in which contracts state that the source IP address and the user id are the most valuable data items and must be maintained private during the computation.

The Edge Gateway produces shares (and hash of shares) for each IP address and user id of the filtered and aggregated data. In particular, in the example of Figure 3 two banks participate in the collaborative Man-in-the-Browser processing system: two shares are thus produced by the Gateways for each IP address and user id. All the remaining data (e.g., financial web site, date, time, time zone) the Gateways manipulate are unchanged as knowing their values during the computation cannot compromise the privacy of customers.

At the end of the pre-processing phase the Edge Gateway produces a stream of data in the form: $<bank\_id, seq_{bank\_id,shareNum}, hash(all\_shares(sourceIP)), sharei(SourceIP), hash(all\_shares(user\_id), share_i(user\_id), service\_id, [date,time,timezone]>$ where $i \in [1, n]$. The data dissemination component of the Edge Gateway sends these data to each corresponding private cloud in the collaborative system as described in previous Section. The data are stored in a portion of the HDFS storage system deployed at each private cloud. They are then processed by a collection of Hadoop's MapReduce jobs [6] handled by our architecture. The processing logic for Man-in-the-Browser detection is expressed in HIVE-QL [7]. With HIVE-QL we divide the processing logic into two flows. The first flow is named Summarization flow; it is responsible for correlating the data and further aggregating them according to statistical anomalies that can be discovered. In our case study, for each IP address, the summarization flow counts the number of distinct accesses done by distinct customers (distinct shares of user id) to different financial services (distinct service id) in the fix time window. The output of this step, as shown in Figure 3 is a collection of summary data in the form $<bank\_id, seq_{bank\_id,shareNum}, hash(all\_shares(sourceIP)), share_i(SourceIP), Num\_Access>$ where $i \in [1, n]$.
The data are then injected into the Blacklisting flow which counts the number of accesses and verifies whether that total number exceeds predefined thresholds. If the check is positive, the Blacklisting produces a stream of data in the form $<share_i(SourceIP)>$ *where* $i \in [1; n]$. This result is ordered according to the ordering protocol (or SQL-like ordering constructs) previously described.
The ordered shares are then passed to the reconstruction unit of the private cloud. We have implemented a Java distributed protocol for reconstruction purposes that takes the ordered shares and starts the reconstruction algorithm as described above.

5. RELATED WORKS

Collaborative processing of data produced by multiple independent sources while preserving privacy is getting momentum. Very recent works have proposed interesting systems which are briefly described in the following.

AIRAVAT [14] is a MapReduce-based system that integrates mandatory access control (decentralized information flow control) and differential privacy technique in order to guarantee the privacy and security during a MapReduce computation. AIRAVAT provides enhanced mappers and reducers that incorporate the earlier mentioned privacy-enabled techniques in order to guarantee privacy during a MapReduce computation. In contrast to this approach, our solution is a non-intrusive one: we run MapReduce jobs that manages encrypted data obtained by means of secure multiparty computation techniques.

SEPIA [10] is a multi-purpose privacy preserving computation system. In SEPIA a group of individual networks collaborate by providing data through so-called input peers. Input peers anonymize data using Shamir's secret sharing scheme: they send shares of a data to be maintained

secret to a (usually smaller) set of privacy peers. The privacy peers perform the actual computation; they can be hosted by either a subset of the networks running the input peers or external parties. The result of the computation is reconstructed and sent back to the input peers provided that specific conditions are satisfied. In our solution, we don't use distributed computation. Instead of that we use SQL-like query to process the data located in each CPS separately, where the data in each CPS is a complete private view of the original data as described before. In this sense our architecture uses open source technology improving thus security and reducing costs.

6. CONCLUDING REMARKS

In this paper we took an abstraction, namely Semantic Room, developed in the context of COMIFIN EU project which enables the construction of trusted collaborative environments for the protection of critical infrastructures from cyber threats and we proposed a novel SR architecture which is capable of collaboratively correlating raw data coming from web logs of financial institutions with the aim of detecting Man-in-The-Browser attacks while preserving the privacy of specific data items. Privacy guarantees are included in the SR contract and reflect the privacy requirements imposed by customers' of SR participants. Financial organizations participating to the semantic room can be deployed across administratively and geographically disjoint domains over the Internet.

We are currently conducting an experimental evaluation of our Semantic Room implementation. Preliminary results show that we are able to detect suspicious Man-in-The-Browser activities in approximately 1 minute. This cost is much higher (roughly one order of magnitude) than the ones obtained by semantic rooms that correlates data without any privacy preserving mechanism in the context of port-scan detection and Session Hijacking detection [2] and [3] . This gave us a preliminary feedback about the cost of implementing privacy preserving computations. We are planning to carry out an extensive experimental evaluation in order to assess more in detail the costs of our privacy mechanisms in terms of both throughput and detection delay when varying the batch window size. In addition, our current implementation may suffer from some privacy guarantee limitations due to the possible link-ability of the shares to the SR participants.

REFERENCES

[1] Aniello, Leonardo and Lodi, Giorgia and Baldoni, Roberto. Inter-domain stealthy port scan detection through complex event processing. In Proceedings of the 13th European Workshop on Dependable Computing, pp. 67-72, 2011, Pisa, Italy.

[2] Elie Bursztein, Chinmay Soman, Dan Boneh, and John C. Mitchell. 2012. SessionJuggler: secure web login from an untrusted terminal using session hijacking. In Proceedings of the 21st international conference on World Wide Web (WWW '12). ACM, New York, NY, USA.

[3] Roberto Baldoni, Giuseppe Di Luna, and Leonardo Querzoni. Collaborative Detection of Coordinated Port Scans. ICDCN proceeding , pp 102-117. 2013.

[4] G. Lodi, L. Querzoni, R. Baldoni, M. Marchetti, M. Colajanni, V. Bortnikov, G. Chockler, E. Dekel, G. Laventman, A. Roytman, Defending financial infrastructures through early warning systems: the intelligence cloud approach, in: Proceedings of the 5th AnnualWorkshop on Cyber Security and Information Intelligence Research, CSIIRW '09, ACM, New York, NY, USA, 2009, pp. 18:1-18:4.

[5] Where Complex Event Processing meets Open Source: Esper. http://esper.codehaus.org/, 2009.

[6] Hadoop. http://hadoop.apache.org/, 2011.

[7] Hive. http://wiki.apache.org/hadoop/Hive, 2011.

[8] Jaql. http://www.jaql.org/, 2011.

[9] D. Bogdanov, S. Laur, and J. Willemson. Sharemind: A Framework for Fast Privacy-Preserving Computations. In Proc. of the 13th European Symposium on Research in Computer Security: Computer Security, ESORICS '08, pages 192-206, Berlin, Heidelberg, 2008. Springer-Verlag.

[10] M. Burkhart, M. Strasser, D. Many, and X. Dimitropoulos. SEPIA: Privacy-Preserving Aggregation of Multi-Domain Network Events and Statistics. In USENIX Security Symposium. USENIX, 2010.

[11] Entrust. Defeating Man-in-the-Browser: How to Prevent the Latest Malware Attacks against Consumer and Corporate Banking, March 2010.

[12] D. Je_rey and S. Ghemawat. MapReduce: simpli_ed data processing on large clusters. Commun. ACM, 51(1):107-113, 2008.

[13] H. F. Lipson. Tracking and Tracing Cyber-Attacks: Technical Challenges and Global Policy, 2002.

[14] I. Roy, S. T. V. Setty, A. Kilzer, V. Shmatikov, and E. Witchel. Airavat: security and privacy for MapReduce. In Proc. of the 7th USENIX conference on Networked systems design and implementation, NSDI'10, Berkeley, CA, USA, 2010. USENIX Association.

[15] A. Shamir. How to share a secret. Communications of the ACM, 22:612-613, 1979.

[16] Y. Xie, V. Sekar, M. K. Reiter, and H. Zhang. Forensic analysis for epidemic attacks in federated networks. In ICNP, pages 143-153, 2006.

[17] G. Zhang and M. Parashar. Cooperative detection and protection against network attacks using decentralized information sharing . Cluster Computing, 13(1):67-86, 2010.

[18] Maher Salem and Ulrich Buehler. Mining Techniques in Network Security to Enhance Intrusion Detection Systems. International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.6, November 2012.

[19] Roberto Baldoni and Gregory Chockler. Collaborative Financial Infrastructure Protection. Springer, 2012.