COLLABORATIVE DEFENCE FOR DISTRIBUTED ATTACKS (CASE STUDY OF PALESTINIAN INFORMATION SYSTEMS)

Hani Qusa

Information Technology Department
University College of Applied Science - Palestine

ABSTRACT

In this paper, we develop a comprehensive approach for protecting national Palestinian information systems. We do not restrict our attention to protecting each individual organization, but rather focus on the entire ecosystem as a whole. Therefore, the developed system will be opened for participation for all Palestinian governmental and non-governmental organizations who are interested in improving their security and protection against current threats and security attacks targeting Palestinian information systems. The results will help in raising the awareness about information security for participating organization.

KEYWORDS

collaborative environments, MapReduce, critical infrastructures, Intrusion detection system, DoS Attack.

1. INTRODUCTION

The new technological trends, the need to provide a competitive services, and the overlap in the businesses in several fields have brought about a shift from the internal organization's infrastructure to engagement in a truly global ecosystem characterized by many cross domain interactions and heterogeneous systems and data. Therefore, the dependency of several governmental and non-governmental organizations on cyber-infrastructure of critical infrastructures encounters an increasing demands and undergoes a profound technological and usage changes.

However, the sophistication of cyber attacks has increased over time and tools and techniques of attacks are easy and widely spread. This comes to fact that the technical knowledge required to exploit existing vulnerabilities is decreasing. For example, organizations that suffer from Distributed Denial of Service (DDoS) attack, know that they have been attacked, but they cannot easily distinguish the group of IP addresses that commit this attack alone, which create a big challenge for organizations to defend themselves. The perpetrators of these attacks, whether motivated by the prospect of financial gain or because they see such an attacks as means of garnering publicity or otherwise pursuing a political cause, benefit from sharing technology and other information among themselves. Therefore, protecting these infrastructures in the face of faults and malicious attacks is crucial to ensure stability, availability, and continuity of the key electronic services and individual businesses worldwide.

DOI : 10.5121/ijnsa.2014.6603

Collaboration on the data level by aggregating data distributed among different organizations is considered as an important analysis in the context of the security for these organizations. Collaboration for security implies combining and correlating large volume of data from distributed organizations in order to have a more comprehensive view of malicious activities that may occur separately and that would have gone undetected if considered in isolation.

In this research, we embark on a research agenda aiming to develop a comprehensive approach for protecting national information systems of Palestinian organizations. We do not restrict our attention to protecting each individual organizations, but rather focus on the entire ecosystem as a whole. Therefore, the developed system will be opened for participation for all Palestinian governmental and non-governmental organizations who are interested in improving their security and protection against current threats and security attacks targeting Palestinian information systems. The results will help in raising the awareness about information security for participating organization.

Our specific objective in the research timeframe will be to devise a scalable distributed monitoring system that will provide the relevant IT components of participating organizations with early notifications about faults and other potentially malicious activity originating at remote sites (possibly belonging to other critical infrastructures) thus enabling those components to trigger the necessary protective mechanisms in a timely fashion.

2. RELATED WORKS

Several systems have been developed in order to protect organizations in collaborative way such as [1][2][6][7][8][9][10][11]. However, customization of these solutions in order to provide the necessary protection for the Palestinian information system is not feasible. This returns to the fact that these information systems are not so advanced to have such protection with more capabilities than required. In other words, one main attack that threatens the Palestinian information systems is the Denial of Services attack. This attack has little impact on financial transaction on the Palestinian information systems. However, the delay impact caused by this attack on the Palestinian community desires to provide the required protection. Thus, customization of an existing solutions or using commercial one will be considered as an overhead cost for these organizations that cannot be afforded. While designing a special system that fits to the existing infrastructure will have a good impact with minimum cost.

3. SYSTEM ARCHITECTURE

3.1 System properties and assumptions

In fact, the security tools used in Palestinian organizations are very simple and depends mainly on commercial firewalls. This exposes them to several types of attacks that cannot be controlled by only firewalls. Therefore, the development of Collaborative Intrusion Detection System (CIDS) for building collaborative defence for Palestinian information system is very important issue as we as it has several challenges.

Furthermore, the old infrastructure of most Palestinian organizations makes difficulty in using existing collaborative intrusion detection systems which requires at least a good services to run on it. We assume that using a Distributed File System (DFS) that can be installed on several individual machines can help in processing the required data.

Finally, the geographical distribution nature of the Palestinian Territories imposes a real challenge which desires a need to adopt distributed collaborative defence system. Another challenge is the differences in the data model used in these organizations. This requires a special normalization level that helps in unifying the form without losing the benefit of the data.

3.2 Threat Model

We consider threat model that takes into account one major type of attackers who try to evade to systems by running port scanning against Palestinian organizations running information systems. The objective of these attackers is to suspend the electronic services provided by targeted organizations. This attack is well-known as the Denial of Service (DoS) attack. More sophisticated attacks of DoS is the distributed one, where the attacker depends of several controlled machines to execute the attack concurrently from these controlled devices, which in know as Distributed DoS (DDos).

In order to carry out such an attack, these sophisticated attackers often run port scanning against single organization at low rate. The attackers use this strategy against several organizations in the same time. By adopting this strategy, it's difficult for individual intrusion detection systems of these organizations to discover such an attacks.

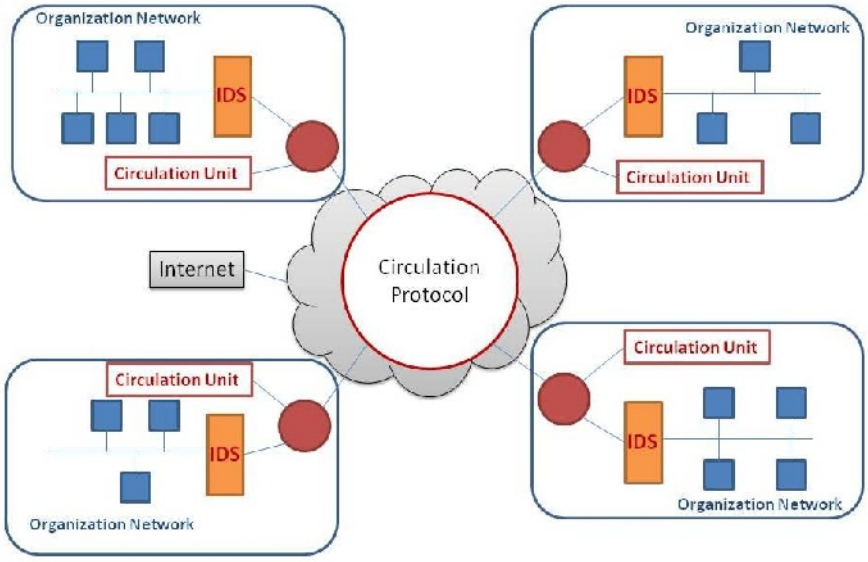

Fig. 1: System Architecture

3.3 System Model

We propose a system model that employs multiple individual detectors installed in different organizations. In each organization a local intrusion detection system IDS is installed in order to monitor the traffic of that organization. The proposed system consists of *n* organizations that use these individual IDSs in order to form Collaborative Intrusion Detection System CIDS. The system will exchange the suspected behaviours among the individual IDSs in order to cooperatively discover malicious behaviours and detecting the sources of these malicious activities. Thus, the system doesn't rely on any external party nor has a centralized node for

processing the data. This form a hierarchical architecture of collaborative intrusion detection system. Fig. 1 shows the architecture of the system.

Practically, the processing of the data in the system is carried out by set of components distributed over the participating organizations. Each organization contains two components; 1) the local Intrusion Detection System (LIDS) and, 2) The Circulation Unit (CU). The phases work in pipeline and are described next in more details.

The first component (LIDS) monitors the network traffic of the organization in order to produce a blacklist of suspicious behaviours. The suspicious behaviour means an IP address that carries out a port scanning activities against that organization.

The second component, CU , are connected together using a routing table. The CUs are responsible for exchanging the suspicious lists among organizations. These suspicious list are generated by the local IDSs. Furthermore, the CUs are responsible for correlation process over the collected data.

4. SYSTEM IMPLEMENTATION

The processing of the data is executed in two phases according to the system architecture. The implementation of the data includes deploying two sub-systems; one for the local processing that uses the LIDS component. And the Collaborative Processing which use the Circulation Units (CU) to process the resulted data from the individual LIDS cooperatively. The processing in the two phases are described next.

4.1 Local Processing Phase

During this phase, each participating organization processes the local traffic network in order to check for anomalies. This phase is implemented by the private processing unit that carried out all the necessary operations. The unit is a private cloud constructed for this purpose that uses a Distributed File System (DFS). Specifically, the system adopt Hadoop DFS (HDSF) [3] in order to store and to process the network traffic data. The system uses a high level query language in order to define the processing logic. The language is compiled into a series of MapReduce jobs [12].

We adopt SQL-like query processing called HIVE [4], that specifies the data patterns to be discovered on the set of large input data. A query processing engine inside each private processing unit retrieves the data in the storage elements and aggregates them according to one or more SQL-like queries. The main output of the query is a filtered and aggregated according to the suspicious IP addresses. All participating organization generate a general format of data that in order to be used in the next phase. The general format of the output is a list of IP addresses and the count of the appearance of this IP addresses in the network traffic of each organization, i.e. *<IP, IP_Count>*.

4.2 Circulation Phase

This module is in charge of collecting the previously generated lists from all participating organizations. The data collection occurs periodically, i.e., every fix time window. The circulation units agrees on disseminating the list of IP addresses according to a specific threshold. For example, if the threshold is set to *100*, then all the aggregated *IP* records with *IP_Count* ≥ *100* will be disseminated. This threshold is denoted as local threshold ($L_{Threshold}$).

The start and the end of each period is determined according a special threshold. Practically, let L_i be the list produced by Participant P_i. and let all participating organizations be connected as a logical ring running over the Internet. Then, participant P_i starts the collection of lists L_j from all the other participants by sending a start-collection message in the form of an empty token T_i. The token is circulated along the ring until all the participants have appended their lists to it. After that, P_i removes the token. This protocol is an enhancement of a previous protocol that was introduced in [1].

In order to satisfy a level of privacy that avoids link-ability between the data and the owner of this data, the first participant that puts the list inside the token is determined at random. Specifically, when participant P_j (where i != j) receives T_i for the first time then, if the token is empty it adds its list L_j to T_i with a probability p_{start}; otherwise, it adds the list for sure. P_i removes the token from the ring when the token passes through it for the second time nonempty and unmodified. This ensures that all participants have added to the token their lists. All participants executes this protocol in the same window of time. The full algorithm is listed in Algorith1

Finally, the circulation units in all participating organizations starts the correlation process over collected data. Another agreed threshold called collaborative threshold ($C_{Threshold}$) is used to decide the black list of IP addresses that have carried out malicious behaviour against participating organizations.

Algorithm1: Circulation Protocol

Input: pr_{start} = probability of start sending
Output: Each participant P_i will compute: $T_{total} = \bigcup_{i=0}^{n} T_i$

Uses:
$n_{id} = node_{id} = i$
A[n] = ;

upon event <init> do
    foreach j 1 → n do
        A[j] = 0;
    $T_i = \Phi$
    T_i.send();

upon event <TokenReceived> do
$n_{id} = T_{recevied}.id$;
    if n_{id} == n_{id} then
    if (T_{prev} == $T_{recevied}$) then
        $T_{Deliver} = T_{recevied}$;
    else
        $T_{tosend} = T_{prev} = T_{received}$;
else
    if ($T_{received}$ == Φ) then
        $pr_{sending}$ = *random()*;
        if ($pr_{sending} \geq pr_{start}$) then
            $T_{tosend} = T_{received} \cup T_i$;
        else
            $T_{tosend} = T_{received}$;
    else
        if (A[n_{id}] == 0) then
            $T_{tosend} = T_{received} \cup T_i$;
            A[n_{id}] = 1;
        else
            $T_{tosend} = T_{received}$;
T_{tosend}.*send()*;

5. PERFORMANCE EVALUATION

This section provides an evaluation of a prototype implementation of the proposed system. The proposed architecture has been implemented in Java and run on a cluster of four 2.3 GHz dual processor physical machines, equipped with 4GB of RAM each one. The physical machines are connected to a LAN of 1Gbit, running ubuntu linux.

The network traffics of the participating organizations were simulated using a log generator that produces a traces for the network traffic in the same format as generated by web server. For

example, the generated log contains TCP connection in a trace which is represented as a tuple <source, destination> with an associated timestamp with granularity up to seconds, and state information (Dropped or Accepted). The generated logs are injected to the LIDS of the participating organizations.

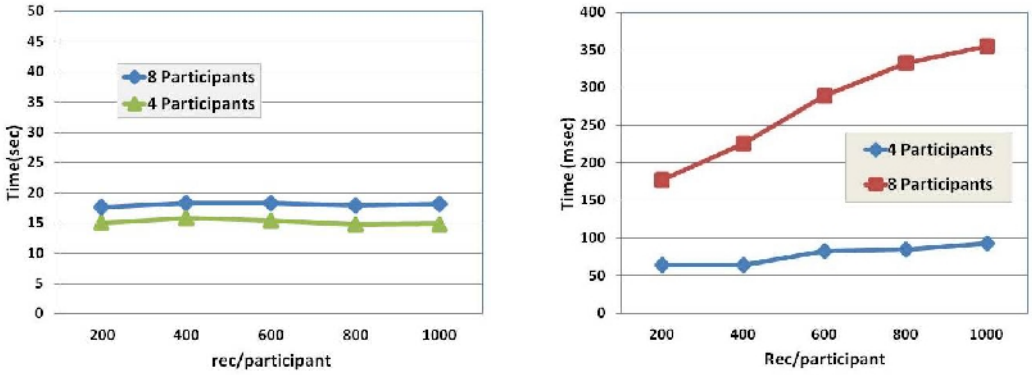

Fig. 2: (a) LIDS processing time, (b) Circulation throughput

Inside the processing unit, the data is processed by a collection of Hadoop's MapReduce jobs inside each participant. The processing logic, which represents our data pattern, is expressed in HIVE-QL statements. Specifically, the language supports SQL-like query constructs that can be combined into flow and specify the data patterns. The query engine retrieves the data in the storage elements and aggregates them according to one or more SQL-like queries. The HIVE-QL statements are compiled into series of MapReduce jobs.

We evaluate the system throughput as a function of the number of records and number of participants that the system can process and serve in a specific period of time with and without applying the proposed privacy-preserving mechanism.

Figure 2(a) shows the throughput of this phase. The time for processing the data is an average of ≈16 sec in all cases. This is because HIVE is used mainly for processing large dataset and it requires a high start-up time, while in our experiments, the dataset comprises 1000 records in case of 8 participants with 1000 record per each one which is not big enough in terms of HIVE. While Figure 2(b) shows the throughput of the circulation phase. Depending on the local network for experimentation indicates mostly linear and very slow increasing in the delay regarding the size of the data.

6. CONCLUDING REMARKS

In this paper we described a prototype for collaborative defense against coordinated attacks. The system is proposed to protect national information systems of the Palestinian organizations. The proposed system is able to collaboratively correlating raw data coming from web logs of Palestinian organizations in order to detect port scanning activities which are used in preliminary phase of conducting DDoS attacks.

We are currently conducting an experimental evaluation of our system implementation. The initial results show that ability of the system to detect suspicious port scanning activities. We are planning to carry out an extensive experimental evaluation to the accuracy of the detection and to consider other types of attacks that threaten Palestinian information system. The future work will

take into consideration other issues with a deep investigation such as the privacy of the data and the scalability of the system. Furthermore we aim to investigate more in the scalability of the system by considering the number of the data manipulated and the number of participants in the systems.

ACKNOWLEDGEMENTS

This work was supported and funded by the scientific research council in the ministry of education & higher education in Palestinian national authority. The author would like to thank reviewers for their thoughtful comments to improve this article.

REFERENCES

[1] Qusa, Hani, and Shadi Abudalfa. "SECURE COLLABORATIVE PROCESSING ARCHITECTURE FOR MITB ATTACK DETECTION." International Journal of Network Security & Its Applications 5.5 (2013).
[2] Roberto Baldoni, Giuseppe Di Luna, and Leonardo Querzoni. Collaborative Detection of Coordinated Port Scans. ICDCN proceeding , pp 102-117. 2013.
[3] Hadoop. http://hadoop.apache.org/, 2011.
[4] Hive. http://wiki.apache.org/hadoop/Hive, 2011.
[5] Jaql. http://www.jaql.org/, 2011.
[6] 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.
[7] D. Je_rey and S. Ghemawat. MapReduce: simplified data processing on large clusters. Commun. ACM, 51(1):107-113, 2008.
[8] 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.
[9] Y. Xie, V. Sekar, M. K. Reiter, and H. Zhang. Forensic analysis for epidemic attacks in federated networks. In ICNP, pages 143-153, 2006.
[10] G. Zhang and M. Parashar. Cooperative detection and protection against network attacks using decentralized information sharing . Cluster Computing, 13(1):67-86, 2010.
[11] 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.
[12] Gates, Alan F., et al. "Building a high-level dataflow system on top of Map-Reduce: the Pig experience." Proceedings of the VLDB Endowment 2.2 (2009): 1414-1425.
[13] H. F. Lipson. Tracking and Tracing Cyber-Attacks: Technical Challenges and Global Policy, 2002.

AUTHORS

Hani Qusa is a Assistant professor in Information Technology Department at University College of Applied Sciences. He received his master and PhD degrees in 2009 and 2013 respectively in computer engineering from university of Rome “La Sapienza”. Additionally, he published several topics in the area of network security, participated in international workshops and exhibitions, and contributed with national and international projects. His research interests include privacy preserving in collaborative environments, secure multiparty computation protocols, distributed computing and secure distributed systems.