Master's Thesis: Small Report and Planning

Charles Jacquet - 2781-12-00

November 18, 2016

1 Introduction

This small report is intended to briefly explain the goal of this master's thesis, explain what has already been done and the planning for the remainder year. The planning part made me realized I don't have as much time as I thought.

2 Threats Sharing

Nowadays, in computer security, we need more than simple malware detectors or anti-viruses. New security threats suddenly appear and we have to react as soon as possible. This is why a new kind of security levels appeared and is called threat sharing.

The idea behind is to share a threat when we discover one, we allow then, every other member of the sharing organization to defend themselves against it but also to put in common all information they have. This results in fast recoveries and protections.

A lot of sharing platforms emerged from this idea. One of them is called MISP, Malware Information Sharing Platform and Threat Sharing [2]. One of its specificities is of being an open source project [1]. We can also find a set of tools on the same repository [2].

This platform is maintained by the Computer Incident Response Center Luxembourg (CIRCL [3]) which is one of the companies helping me in my work. The second one is Conostix [4] which provides security and system services in Luxembourg.

[1] `https://github.com/MISP/MISP`

[2] `https://github.com/MISP/`

[3] `https://www.circl.lu/`

[4] `http://conostix.lu/`

3 IOCs

In the last section, I spoke about threats. I will subsequently be referring to a threat by the name of event (to be coherent with MISP). Then, an event is composed of a set of attributes.
An attribute is an information linked to an event. Thus if we find it on another computer system, we know that it is probably infected by this event. As an attribute somehow indicates the possibility of being infected, we also call it an Indicator Of Compromise (IOC).
But, an IOC is either a value with a specific type or a concatenation of values. Let's see some example of fictional IOCs:

- ip => 192.168.0.2
- hostname => hostname.com
- hostname|port => hostname.com|80
- ...

All different possible types are specified in the MISP guide[5]

4 Privacy Preserving IOCs Sharing

IOCs are commonly considered as sensitive data. This induces companies with confidential data to be unwilling to share IOCs. Especially when in some case, a user needs to have these IOCs on a file that he can bring with him to check a computer.
Nevertheless, they have no problems to share data if, the user already knows it. Which means that a user that has seen the IOC on his computer could be able to discover that another organization have seen it and then, to get back information they have, but never otherwise.

This is really difficult because we need to find a way to give data without giving them. A first implemented idea is the misp-workbench[6] which is a hash-store implemented by CIRCL. They hash all IOCs, and put them into a redis database.
Then, a user possessing an information can hash it and makes a request but not otherwise.
Even it works quite well with big sized data, it is not the same with small ones and IPv4 is a good example of it.
It is actually feasible to create a table with all possible IPv4 with their hashed value. Then an attacker could brute force the database and get to know all IPs stored in the system with their information.

[5] `https://www.circl.lu/doc/misp/`

[6] `https://github.com/MISP/misp-workbench`

Another idea is explained by van De Kamp and al in [1]. The first part of my master's thesis was to implement it.
This algorithm has two steps, the first one is to create a set of rules where each one represents an IOC and is stocked as a file.
A rule has two important information, the first one is the list of the concatenated value types. While the second one is the encrypted information.
What we want, is that only a user having seen the right values can decrypt the IOC. For that, as we have all attribute types, we can concatenate our values in the same order before using this value in a key derivation algorithm. If the values are the right ones, the key just found allows decrypting the information which is, in our case the id and uuid of the IOC. The information thus allows a user to access the event on the MISP web API.
The second step is looking for matches by trying to decrypt each rule with seen values.

Additionally, they have added a different salt for each rule message. This salt ensure that an attacker is not able to precompute all possible decryption keys. After that, they also added the id of the user in the key generation (I'm using the token instead). Thanks to this idea, if rule files are leaked, we know which user is the responsible for it.

My implementation is located in a github repository [7]. I still need to improve it for memory utilization, visualization and other kinds of implementations.
Once all these things will be done, I'll have to clean the code and to test performance with real cases.

5 Allow Everyone to Query

Conostix has implemented a server like system based on DNS called Blackns. The interest is that every computer can make a dns query without installing anything. The idea is to use the solution explained just before with this system . Unfortunately, I suppose that it will not fit because it needs a lot of computation. For each check, we need to generate a key for each rule before trying to decrypt the rule's information with the key hoping for a match. Though It could work and that is why I want to implement it, but I have to find a way to ensure a protection against DOS attacks.
But I've found a set of other kinds of ideas that could be interesting in this case as we don't need anymore to hide every information in a file format as done with the rules and moreover, a server can also do some additional computation on data before sending them.

[7] `https://github.com/charly077/thesis/tree/master/encrypted_rules_implem/`

6 Planning

I can divide what I still want to do into 4 different steps.

- End of the first implementation
- Second implementation with Blackns
- Performance analyses on real data
- Write the report

I want thus to end this first implementation for the end of December, before my Blocus and exams.
I also want for the same time to know how I will analyze the performances and have a clear idea of the second implementation.

I then want the second implementation to be finished for early March. And the performance analyses for the end of March.
I will then use the Easter Holidays to write the report and I want it to be finished to the 16/04. This is in order to get enough time to make corrections proposed by the companies and then by the teacher.

For now, I'm going to continue to work every Thursdays and Fridays on it up to the end of this quadrimester. I have also done a more detailed planning on trello : `https://trello.com/b/805BZSWx`.

References

[1] van de Kamp, T., Peter, A., Everts, M. H., and Jonker, W. Private sharing of iocs and sightings. In *Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security* (2016), ACM, pp. 35–38.

[2] Wagner, C., Dulaunoy, A., Wagener, G., and Iklody, A. Misp: The design and implementation of a collaborative threat intelligence sharing platform. In *Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security* (2016), ACM, pp. 49–56.