Mémoire

Charles Jacquet - 2781-12-00

February 23, 2017

Abstract

Malicious software (malware) are plaguing this computer age. But how to avoid them ? An actual solution is threat information sharing where companies share lists of Indicators Of Compromise (e.g. IPs, mail addresses, urls, malware hashes and so on) with each other. An interesting tool is the MISP (Malware Information Sharing Platform), a platform that allows companies to share these IOCs but also malware analysis. Sometimes, the information is somehow confidential, but still interesting to be shared, thus we need to find a way to share these secret information without disclosing them entirely. The goal of this master thesis will be to analyze the state of the art and to implement different solutions working with MISP in order to compare their performances.

1 Interaction with this Latex Document

In order to facilitate comments, I've added some new commands explained by[1]:

- unsure => What I need to check
- change => What have to be modified
- info => Add simple comment
- improvement => Indicate a possible improvement

2 Steps

- Establishing a small state of the art. (IOC - Sharing - Standards - Cryptography)
- Misp-workbench analysis to understand what's behing
- Implementation [5] on MISP (with modifications)
- Implement additional ideas on top of the paper
- Performance analysis of these implementations

[1] http://tex.stackexchange.com/questions/9796/how-to-add-todo-notes

3 MISP and Thread Sharing

Nowadays, cyber specialists agree to say that Information Sharing is needed to defend our computer systems against digital threats and attacks. That's why a lot sharing platform appeared like DShield, Critical Stack's Intel, Microsoft's Interflow, AlienVault's Open Threat Exchange, ...
Here I focused on the MISP database which is, as said on their website, an open source software solution for collecting, storing, distributing and sharing cyber security indicators and threat about cyber security incidents analysis and malware analysis. MISP is designed by and for incident analysts, security and ICT professionals or malware reverser to support their day-to-day operations to share structured informations efficiently.
This system is based on a set of "Events" and each event contains a set of IOCs called attributes.
To explain it, easily, we can take the example of an analyst that gets an information to share. First, he creates an event. Then he splits all information into fine grained attributes (e.g. ips, urls, uri, ...). All these fine-grained information are also called IOCs. An Indicator of Compromise can, in this context, be defined as a data that can indicate an infection by a "event" if found on a computer system.

4 Information Sharing State of the Art

The idea of this section is to sum up all different articles read at the beginning of the master thesis. These are articles are linked to the subject but sometimes not directly but helped me to the understanding of the subject, and especially to discover ideas to create some possible solutions.
These articles can be divided into some sections, the first one is data sanitization, this is quite interesting even if it could not be applied in our case (as explained in a later section).
Then we have articles on confidencial database and S2P computations

I have to complete these parts with articles

Add the explaination on the article on ip sharing at the end

Add also a section of other kind of detection system like the DNS one of http://conferences.sigcomm.org/imc/201

One of the first articles that presents some concerns about privacy in sharing security alert is [3]. More precisely, they were concerned about protecting site-private topology, proprietary content, client relationship and site defensive capabilities or vulnerabilities.
This was done in 2 steps, the first one, called data sanitization consist in removing confidential data and remove useless information. We don't take the chance of revealing information to an attacker if this one is not needed.
The second one is the correlation/aggregation work were alerts are linked together for analyses purpose.
Before explaining deeper how they sanitize data, it's interesting to first focus on how they get them!
They have used three different categories:

- Firewalls : They consider all "deny" as a possible attack

- IDS : They remember logs of attacks that the IDS has found
- Anti-viruses softwares : gives also some interesting logs

They based their analyses on data coming from DShield and Symantec's DeepSight.

Let's come back on data sanitization, as already explained, we first remove all useless information, then, we can hash all confidential data!
The advantage of leaving this work to the company is to avoid the need of trust on the repository.
This technique is quite well working if, the data has a certain size. But, on the other hand, it's not useful for IP addresses, if an attacker is targeting a company, it has to precompute only 256 or perhaps 65536 IP address hashes. Thus this is not brute force resistant!
For each alert, we have two different IPs, the source IP (ip_src) and the destination IP (ip_dest). We can classify all these IPs in two categories:

- Internal IPs : IPs that belong to the company
- External IPs : IPs external to the company

The first category is, of course, the one that we want to protect and in order to do so, these IPs are hashed with a HMAC algorithm. While the second one is hashed by a simple hash algorithm like SHA-1.
The result is that we can compare all SHA-1 hashed IPs together while only companies can decrypt their own internal IP addresses.
It is an efficient technique as they receive millions of IPs all the time. And, as the attacker is not able to see if the IP is hashed by HMAC or SHA-1, he has to test all hashed IPs against a precomputed table which is not feasible.
They are also using another set of protections like the randomized threshold for publication of an alert but I'm not going to spend time on it.
In sanitization, the also round all timestamp to the nearest minute in order to add some uncertainty!
The second step is the correlation, they spoke about historical trend analyses, source/target-based analyses and event-driven analyses but some other articles are more interesting for the correlation principle, thus I'm not going deeper in it.

Then, [8] was also working on confidential data sharing starting from the first article, but they came up with a new interesting idea, instead of hashing confidential data, why not generalize it and do probabilistic correlations.
(They also used a technique to create probabilistic attack scenario which is a set of alerts that are put together to create a bigger attack).
Guided alert sanitization with concept hierarchies:
For example, if we have an IP 192.168.1.123/32, we can generalize it to 192.168.0.0/16.
The depth of the generalization is chosen thanks to the entropy or the differential entropy technique explained in [1].
Alert Correlation:

They focused on defining similarity functions between sanitized attributes and building attack scenarios from sanitized attributes

This article was interesting for seeing a technique of data obfuscation. And then to create correlation analyze but, it's difficult to apply that technique in order to create a database of confidential data !

I've explained some solutions that can be applied to IP addresses or file (just hashing them). But, what if we could do the same with all network packets and still getting some privacy!
That's the goal of [4] ! Today, it's not enough to analyze IPs, URLs and so on. We need to go deeper in it, that's why they propose a technique based on the byte distribution of the packets.
They used PAYL and Anagram [7], systems that they have created and which are really useful in these analyses.

But, if we have some confidential information, instead of sharing with everyone, we could simply select organizations with which we have benefit to share. But for that, each organization needs to have their database and to be able to do some secure two-party computations.
The interest is to get some metric, for example, if there are only IPs in the databases, if we can get the intersection, and if the cardinality of this intersection is non-negligible, we know that it could be interesting to share with them!
The paper written by Freudiger and al [2] focused on this problem by working on a DShield dataset. They've experimented some strategies to know if it could be useful to share or not with another company. And then, they also experimented to share the whole dataset of the company, only the data set linked to the intersection just found or only, the intersection (just to get a rough idea of what they have in common ...).
Their conclusion were intuitively expected but still interesting :

- More information we get on an attacker, the better the prediction are !
- The chosen collaboration strategy has a really big impact (some of the strategies are really useless)
- Collaboration with companies improves not only the predictions but also removes a lot of false positive.
- Sharing only about common attackers is almost as useful as sharing everything!

5 Sum up

The first idea imagined was a simple proof of work database. But, even if it's an interesting way of remembering and sharing data while avoiding total disclosure of the data set. It has a big disadvantage as well, we cannot really

use data for analysis since an enormous amount of computation is needed to get all information, and then, how could we compare them?
Thus, I had to go deeper and to look into scientific papers. As there are not a lot of information about MISP in this kind of literature , I had to find another starting point. The one I found was DShield. This is also a kind of sharing platform. Even if the way they get or use data is quite different, the general concept is the same or at least for my approach.

there is now a new paper [6]

Then, I could find a lot of techniques in order to generate blacklists, detect event correlations and protect companies with it.
I also found privacy concerns about sharing data. And thus, a lot of separate techniques. For example, if there is privacy problem with sharing files, we can simply hash them. It's not possible to brute force in order to get back the starting file but, for IP addresses, since in general, there are only 256 or 65536 IP addresses for a specific company, we could create a table with all possible hashes and test them all!
That's why it was interesting to see how they handled these problems but, as these techniques were built in view of a special purpose, they have some disadvantages for us.
For example, if we need to store the whole data set, proof of work database with bloom filter to make some analyze could be enough but if the idea is to make correlation between data, this idea is not good enough and we need to focus on other techniques.
But on the other hand, an other widely used technique is data sanitization as seen with articles examples. But again, some problem arise when we want to analyse data. Wath could we do with an ip where the most signicant bit could have been modified by noise. Or with an ip without modified with concept hierarchies ?
Sanitized data loose all the information needed in these process so we need to find one another technique.
That's why I created a set of questions that could help me to know what I need to focus myself on :

- When we spoke about privacy, do we speak only about data privacy or also about source anonymity ?
  ⇒ Here the question is not to know who had seen it before, but just to determine what is going on. We only need to have like a hashstore of data to be sure that data cannot be recovered ! But we need to find a way to be sure to securely share IPs !

- How to define the sate of an IOC ?
  ⇒ This is done in misp and has no impact on my work

- How MISP is used by companies? What is really the difference with DShield ?
  ⇒ DShield is used just by getting automatically a lot of data in order to discover as soon as possible all big attack in order to block them ! While MISP is to analyze event and share it with other companies

- In MISP, event correlations are done but, are they working on attack scenarios ?
  ⇒ I don't think so, but I forgot to ask this question
- Are misp allow some groups (collaboration) to know the intersection of their attributes with other group in order to know if it's worth to share ?
  ⇒ No, at least not for the moment !

6 Useful Cryptographic Functions

Todo

7 On what I worked

We have a data set of malicious data, IOCs, events and we want to share them. And it is already done by the MISP project. But now, if we want to distribute these information to every one?
It would be really nice, every computer specialist could check on computers to discover infections, problems and moreover fix it thanks to previous analysis contained on MISP. But there is a problem, some information are confidential and we need to have some privacy concerns while sharing.
Actually, it is quite easy to understand this fact. If a company have data, they have it so they don't want to share it! Even more if it can be confidential but in the other hand, if they can avoid infection, or detect it with information from other companies, there, they are interested!
But of course this means that someone need to share information, thus, how could we share these information without leaking any confidential data ?
Sanitization is a good idea but, if we do sanitization in order to still be able to recover data, an adversary could do the same. So Sanitization to protect data would modify data up to make them unusable for every one.
But what if we could find a way to share only if the user has really knowledge of the event and can share some, or is really infected by and need the information. While an attacker could not be able to discover anything of the data set.
I will consider this problem but with two different kind of solution. The first part is when the database could be shared because an attacker could not get information from it (Easier to get data, so perhaps we cannot put all data on it).
And the second approach is one that still need a server to respond but allowing more privacy.

But still, the idea that a common user have access to data while an attacker, which is a specialist cannot seems infeasible in a lot cases. The attacker always ends with the data but, if he takes 1 second, 1 day, 1 week or 3 months, years and so in is different because, we can then think about how many time data are

valid?

IP dynamique, fixed , malware removed, ...

8 Misp-Worbench hashstore

One idea already implemented is a redis[2] hashstore located in the misp workbench project[3] and is aimed to get all hashed IOCs. The result is that only the ones who have seen the IOC can get the associated information.
Back on the small data problem, if we consider an attacker that want to try every possible IPs, for IPv4 it represents 4.228.250.625 different IPs that needs to be tested. Even if it is a lot it is still feasible ! Moreover, not all IPv4 need to be tested, for an example we can avoid private subnet like 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16 which already represent 17.891.328 addresses.

To complete ! Mais il y a une grosse partie du misp-workbench que je ne comprends pas et que je ne vois pas à quoi ça sert vu que je n'ai pas accès à la db .. J'aimerai pouvoir voir comment ça fonctionne pour pouvoir l'implem dans l'autre

9 Limitation

it is intrinsically impossible to fully hide the IOC while still allowing a subscriber to evaluate the rule.

Explain why

10 Ideas

10.1 Bloom filter

Bloom filter[4] is a widely used solution.But unfortunately thanks to correlation we could get back the whole dataset.
Misp-workbench is similar to a bloom filter with only one function.

Explain bloom filter

Check this in the code because it is said in the documentation that it's a bloom filter

10.2 Machine Learning

The idea is quite the same as the bloom filter, but, here we want to privilege the privacy! By that I mean, in the bloom filter, there are False Positive, True Positive, True Negative, but NO False Negative.
Here the idea is to accept False Negative and analyze how it impacts the database.
To stay simple, we can see an address ip as a bit sequence. If we use the entire set of ip in the database to train an algorithm as support vector machine (svm). Then it could be interesting to check the base concept representation (BCR) to analyze if this technique could be interesting.

Je dois m'attarder un peu plus sur cette solution

In order to test this technique, I would train an algorithm on the whole set of data. In standard machine learning system, we would use a test set different

[2] http://redis.io
[3] https://github.com/MISP/misp-workbench
[4] https://en.wikipedia.org/wiki/Bloom_filter

from the training set in order to avoid overfitting but in our system, we need overfitting.
After that, the idea would be to test random ip and compare the result.

10.3 Secure Two Party Computation (Intersection)

Must create a list of all ip needed to request. Use the algorithm to compute each part and get to know the intersection. But need and idea to limit the size !

10.4 Proof of Work Database

Here, we want to keep a database, we don't want any False Positive nor False Negative. Thus if we keep a simple database as the hashstore, it's possible to compute every hash and test them all!
But what if we can avoid that by adding computation, this make brute force more difficult but still not unfeasible. But at least with this technique we can avoid precomputation techniques.

hash cache pour les mail contre les spams

The database server choose a random key.
This key need to change every ??x seconds/minute/hours?? and every ??100?? access. (we can imagine a system like the one of bitcoin (hash with specific properties to find))

then in order to get the answer, the request must be like
(|| is a concatenation)
hash(IP)||hash(IP||key)
With a LONG hash function

10.5 All in one request

Here the goal is quite different, Imagine the case of a forensic detection of what had gone wrong.
We can do it differently, we do a full analyze of the machine, then, once we have the whole data set that we went to test, we send everything and the data system just answer with the id of the triggered event but we don't know what triggered it thus we don't really know the content.
Of course with this, we could just send requests of 1 elements, so to thwart that, we simple could make the request difficult (proof of work), also with a big delay to be able to request again from the same ip and moreover, we could make impossible to make request with less than 10 IPs (need to add random other ip).

10.6 All in one request with S2P

The idea is the same as the one of all in one request but, in the case that the user doesn't want to send everything that he have visited, they could use

a secure two party computation intersection algorithm. With that, the server only receive the intersection between his dataset and the one of the user.

10.7 Private Sharing of IOCs and Sightings [5]

This paper consider a cryptographic approach to hide the details of an indicator of compromise. They consider two different phases, the first is to share these IOCs and the second one is to privately reporting the sightings of IOCs.
First, they define an IOC as a propositional formula where the propositional variables are defined over features or observables. They also claim that every IOC can be expressed in the Disjunction Normal Form (DNF) without any negation (e.g. destIP= 198.51.100.43 $\wedge$ destPort = 80).
They store IOCs by hashing the concatenation of all information but, if we consider that IPv4 brute force are feasible, IPv4||port_number is still feasible (Most part of the port are the same as 80, 443, ...).

Erreur! I have to change that because I didn't really understood the scheme when writing that !

startPad = ’
x00’*16

Add subscriber id when creating the file for a specific id :)

10.7.1 Create a rule

- Create a salt and an iv
- password = all IOC's value joined by a comma
- create the key with from the salt and the password
- encrypt the message (CAO) in (aes, ctr) but add a starting padding startPad (used to see if the decryption match)
- create the rule with ConfigParser
- return the rule or write it into a file

10.7.2 match a rule

- parse the rule or all rules in a file
- for each one test to encrypt the attribute with the salt + value and see if the decryption is correct (startPad)

(

"hmset" pour redis) use the token as the client id ! Create 2 backends:
-> from database ==> create rule files and redis dump
-> from web api + token ==> create rule files and directly into redis

I don't know wich intermediate format to use, redis dump if easy, or the rule files like them

Then create the matching system

Both system will be really similar to what I need, so I will only have to do few modifications !

11 My Implementation

11.1 [5]

11.2 Additional choice

11.3 Generalization

12 Chosen Crytographic System

13 Security Discussion

14 Benchmarking

15 Further Work

- Sightings
- additional crypto systems
- additional benchmark
- ...

References

[1] Cover, T. M., and Thomas, J. A. Elements of information theory, 1991.

[2] Freudiger, J., De Cristofaro, E., and Brito, A. E. Controlled data sharing for collaborative predictive blacklisting. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment* (2015), Springer, pp. 327–349.

[3] Lincoln, P., Porras, P. A., and Shmatikov, V. Privacy-preserving sharing and correlation of security alerts. In *USENIX Security Symposium* (2004), pp. 239–254.

[4] Parekh, J. J., Wang, K., and Stolfo, S. J. Privacy-preserving payload-based correlation for accurate malicious traffic detection. In *Proceedings of the 2006 SIGCOMM workshop on Large-scale attack defense* (2006), ACM, pp. 99–106.

[5] 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.

[6] 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.

[7] Wang, K. *Network payload-based anomaly detection and content-based alert correlation*. PhD thesis, Columbia University, 2006.

[8] Xu, D., and Ning, P. Privacy-preserving alert correlation: a concept hierarchy based approach. In *21st Annual Computer Security Applications Conference (ACSAC'05)* (2005), IEEE, pp. 10–pp.