MISP - The Design and Implementation of a Collaborative Threat Intelligence Sharing Platform

Cynthia Wagner
Fondation RESTENA
2, avenue de l'Université
L-4365 Esch-sur-Alzette,
Luxembourg
cynthia.wagner@restena.lu

Alexandre Dulaunoy
CIRCL - Computer Incident
Response Center Luxembourg
41, Avenue de la Gare
L-1611 Luxembourg,
Luxembourg
alexandre.dulaunoy@circl.lu

Gérard Wagener
CIRCL - Computer Incident
Response Center Luxembourg
41, Avenue de la Gare
L-1611 Luxembourg,
Luxembourg
gerard.wagener@circl.lu

Andras Iklody
CIRCL - Computer Incident
Response Center Luxembourg
41, Avenue de la Gare
L-1611 Luxembourg,
Luxembourg
andras.iklody@circl.lu

ABSTRACT

The IT community is confronted with incidents of all kinds and nature, new threats appear on a daily basis. Fighting these security incidents individually is almost impossible. Sharing information about threats among the community has become a key element in incident response to stay on top of the attackers. Reliable information resources, providing credible information, are therefore essential to the IT community, or even at broader scale, to intelligence communities or fraud detection groups.

This paper presents the Malware Information Sharing Platform (MISP) and threat sharing project, a trusted platform, that allows the collection and sharing of important indicators of compromise (IoC) of targeted attacks, but also threat information like vulnerabilities or financial indicators used in fraud cases. The aim of MISP is to help in setting up preventive actions and counter-measures used against targeted attacks. Enable detection via collaborative-knowledge-sharing about existing malware and other threats.

Keywords

Threat intelligence management; IT security; collaborative information sharing; trust; incident response

WISCS'16, October 24 2016, Vienna, Austria

ISBN 978-1-4503-4565-1/16/10...$15.00

DOI: http://dx.doi.org/10.1145/2994539.2994542

1. INTRODUCTION

The number of new threats and incident indicators are constantly increasing and there is no indication that this trend will stop soon. Detecting and handling these threats individually has become almost impossible, since accurate classification or reliable taxonomies of threats differ within existing solutions and often the distribution of information is limited or restricted to selected users. This poses major constraints.

In the era of 'generation Y' or 'generation social media', individuals who grew up with technologies to become so-called digital natives, sharing and collaboration within a community has become an attitude towards life. Recently, this trend of sharing all kind of information within a community can also be observed for the IT-community. Promoting collaboration and information sharing is critical in community driven domains such as IT. On one hand due to the sensitiveness of data, and on the other by sharing information, new threats can be identified more quickly in a joint-effort and response can be adequately coordinated throughout the whole community. Therefore, the need for having reliable information sharing platforms in place will be a key to successful collaboration and incident response in the near future.

This paper presents the Malware Information Sharing Platform, also called MISP, and provides an overview of its technical implementation. The aim of this project is to provide a platform, where actors of private or public IT-communities can share information and IoCs about existing threats from various domains. Such as cyber security, finance, etc., to contribute to a better over-all security understanding.

The paper is organized as follows: Section 2 discusses recent works that deal with the handling of threat intelligence collection and sharing. Section 3 provides the motivation for MISP, describes the most important technical modules like the sharing models and the synchronization process. In section 4 the actual MISP platform is briefly described. Section 5 shows the actual results about usage and relevant statistics. Some future work and conclusions are given in section 6.

2. RELATED WORK

Information sharing is a major asset in the IT world and has gained significant importance in the area of research too. Large companies selling threat intelligence within their commercial solutions have gained a large market share, as for example IBM, Dell secure Works, Crowdstrike, McAfee, CISCO, CheckPoint and many more.

Dandurand et al. [5] explain that the most important requirement for a successful threat intelligence system is the facility to share information, automate information sharing and the ability to generate, refine and control data. In [5], these requirements were extended by defining a concept of knowledge management for the area of cyber security by adding needs. These include the ability for collaboration and human and/or machine interfaces for automation, to cite only a few. In [15], the difficulty and motivation for information sharing is discussed; like trust issues and the problem to keep the online community active to contribute. [3] gives an overview about challenges encountered in the domain of threat intelligence and tries to summarize the requirements and needs to build successful threat intelligence platforms. It is also highlighted that there are some requirements discussing the added valued of shared data and privacy, respectively law issues for these systems.

For sharing information, a lot of effort has already been put in structuring information by introducing different kinds of data formats and transport mechanisms. For example in [2], STIX/TAXII has been introduced to combine human to machine data to share information. In [6] the Incident Object Description Exchange Format, IODEF is described. It provides a data sharing framework for computer security incident teams by combining text with structured data. A similar approach is introduced in [14].

Beside the various existing data formats and transport mechanisms, several technical implementations of threat intelligence platforms exist. In [10], a model to represent the topology of sharing by using a graph model is introduced that applies parameters like added-value of information and trust/repudiation. In [12], a new method to assess the threat level for a piece of malware is presented, where scoring factors weigh the malware to evaluate its level of threat. Another method is presented in [1], where a threat intelligence platform is designed that uses a publish-subscribe communication model by combining STIX to the Extensible Messaging and Presence Protocol (XMPP).

Evaluating and representing large quantities of information is also a major problem in the daily management of information sharing platforms. In [20] for example, a data mining approach based on similarity metrics is presented to identify statistical patterns and other relations in shared information as for example real incident tickets.

Another important point in information sharing is the usability and user experience (UX) for existing platforms. In [17], a systematic study is presented that highlights human elements, while using information sharing platforms. Therein it discusses major user experience requirements for improving the usability of this kind of platforms.

Recently, many guidelines, best practices and summaries on existing platforms have been published. In [11], guidelines for information sharing as well as the benefits and challenges of information sharing are discussed. In [18], a survey on the implementation and organization of information sharing platforms was realized to discuss the overall dimension of information sharing. It was concluded that the effectiveness of the platforms could be increased by having a strongly active sector-oriented community; within which incidents could be shared rapidly with experience reports. In [8], a case study for information sharing has been performed in order to identify issues and hurdles in organizational, technical and legal domains. An outcome of this survey indicated that information sharing remains a group activity and that there is a real need to reduce the number of false positives. In [9] by ENISA, a summary on the threat landscape is provided. It discusses and encourages both, secure communication and information sharing between CERTs.

3. OVERVIEW OF MISP

The following section describes the motivation for the sharing model as well as the major technical modules. Among others the graph modular approach and the redundancies that were implemented for the MISP platform.

Before focussing on the technical side of the platform, the term ‘information’ in the context of the MISP platform should be defined. In this paper, information that can be shared is defined as any kind of relevant indicator for threats, IoCs, and all other kinds of information from various domains such as cyber security, finance, etc.

3.1 Data model

The data model describes the standard description format for creating events in MISP. The main motivation was to have a simple and convenient format while at the same time enabling more complex requirements. An advantage of this simple approach is that a user can decide him-/herself the level of granularity of information that he/she wants to share. For example, a user can describe an event with multiple attributes while providing as much information as possible, or he/she can only put a minimum of information for an event.

Another reason for this model was to have a flat model to ease the work of parsing and to avoid ambiguity (e.g. STIX). Composite observables in STIX are very often flattened and neglected by the parser which introduces rejected observables to be included. The main objective is to rely on a minimum viable data format and extend it as the need for additional complexity arises instead of trying to capture all possible future requirements in advance. A new entry in MISP is called an **event** object. An event can be defined as a set of characteristics and all kinds of descriptions for an IoC, including attachments, etc. These characteristics and relevant information are called **attributes**. Event attributes for example are IoC date, threat level, comments, organisation,...

Attributes are mainly defined by two fields, **category** and **type**. The main difference is that the **category** field describes what the attribute represents, such as targeting data, network activity, financial fraud, etc. whereas the **type** field describes how the attribute represents the chosen category. Some examples for attribute types are checksums (md5, sha1), filename, hostname, ip-address, email source and destination, etc. The actual payload of the attribute

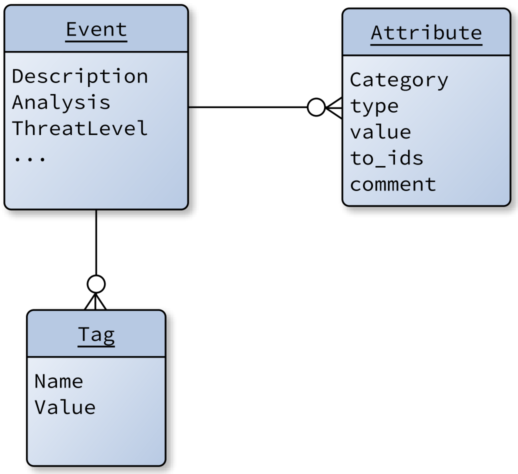


Figure 1: Simplified event representation in MISP

is in the value field and additionally in the case of malware samples or attachments in the base 64 encoded data field.

Furthermore, an event can also have tags. A simplified representation of this data model is given in Figure 1.

3.2 Sharing models

The motivation for sharing information can be manifold, since humans have contradicting needs in a sense of 'security versus relatedness'. On one side, people that share information about occurred threats and incidents within a community would prefer keeping it secret. On the other side, by sharing information, new insights or similar information, as well as possible response actions, can be extracted from this community.

Intrinsic motivation, as described in the self-determination theory, [7] explains that humans can perform or initiate actions without the need of external, but for internal rewards. In this case this means, people explicitly share information about threats or incidents within a community (relatedness) in order to gain information about new threats that are published by others (security).

3.2.1 Sharing levels

MISP relies on the voluntary action of its community to share information and indicators. Furthermore, the level of reach of the content is left to the sharer, who can select various sharing scenarios, as described below:

- **organization only**: Only members of an organization are allowed to see an event.
- **community only**: Users of the MISP community can see the event, including organizations that run MISP servers that synchronize with that server.
- **connected communities**: Users of the MISP community, including organizations on this MISP server, as well as MISP servers synchronizing that server. This also includes hosting organizations of servers that connect to these servers.
- **all**: The shared content is shared within the whole MISP communities.
- **Sharing Group**: A distribution list approach that can include a set of organisations and remote MISP instances. This setting allows for granular distribution as well as the option to entrust partners with an extending role within the sharing group.

3.2.2 Proposals

In order to ensure the integrity and veracity of the data distributed by MISP, the modification of events is only permitted to members of the creating organisation. However, one of the key aspects of successful information sharing is a focus on collaboration and providing the user base with a feedback loop. Proposals allow users to make suggestions for changes to an event, created by another organisation. Proposals are an integral part of data that is distributed among MISP instances and will be further described in the pull and push mechanism section. A user can suggest a proposal to an event that was created by a different organisation on a remote instance. This proposal is reported back to the original creator of the event, who may accept or discard it. Either way, the outcome of this decision will be propagated back to all interconnected instances.

Typical uses of this feature are for example the notification of false positives to an event creator, asking for an error correction, or simply completing an existing event by additional findings.

3.3 Taxonomies

User experience collected from older MISP versions showed that people do not want to spend too much time to fill in fields in web forms or to copy and paste information. A complicated user interface was one limiting factor of information sharing. Hence, the free text importer feature was introduced. A user can copy and paste raw data into a single field that is then fed through an algorithm relying on heuristics to match the attributes. The resulting attributes are presented to the user who has to validate the matchings.

Interactions with MISP can be done with a REST (REpresentational State Transfer) interface. A Python library (PyMISP)[1] is available and allows to interact with MISP APO. Tools like Cuckoo sandbox[2] and Viper analysis[3] supports MISP to allow a bidirectional (pushing and/or pulling) information.

These features, in conjunction with the steadily increasing number of users, overwhelmed some others which lead to the requirement of filtering events. This requirement is also useful for handling information classification. The classification is often bound to internal, community or national classification schemes. Another common problem is the description of the events or the mapping of events into categories. This is a complex task since the number of categories is not always known in advance. A typical example here is; the types of attack as they evolve and change quickly.

[1] https://github.com/CIRCL/PyMISP
[2] https://www.cuckoosandbox.org/
[3] https://github.com/viper-framework/viper

Experience has shown that these challenges are often related to the context and thus, the users of the MISP software. A centralized pre-defined set of definitions that satisfying all the potential users is a hard challenge and so, a distributed approach based on machine tags was introduced. Tags can be defined per MISP instance and are exportable. This allows the reusing of tags from other MISP instances.

The freedom of defining tags quickly lead to a situation where tags were redefined making filtering complicated. To overcome this problem, a new concept of tagging was introduced, the taxonomies.

A taxonomy is based on the triple tag solution that was introduced by Flickr[19]. The triple tag structure has a namespace, predicate and value. In the example :
{**admirality-scale** : **source-reliability** = **'fair'**}, admirality-scale is the namespace, source-reliability is the predicate and 'fair' the value. A clear advantage of this concept is the still human readable format of the machine tags. The repository of taxonomies for the open source community[4] includes taxonomies modeling national, intelligence, law enforcement, csirt classifications and many others domains. In case that none of the predefined taxonomies fits the description of an event, the user can formulate his/her own taxonomy. This introduces a notion of folksonomy into MISP and keeps the tagging structure more organic.

3.4 Synchronization protocol

In the following section, the synchronization protocol will be further explained. The algorithm used in MISP is based on a trial-and-error approach, where the main focus was put on efficiency, accuracy and scalability. The final algorithm implemented in MISP resulted in simple models called pull, push and cherry-pick technique.

As MISP is a distributed set of instances, events are assigned a universally unique identifier (UUID) each. Beside this, events may contain one or more attributes, which also have uniquely assigned UUIDs.

3.4.1 Pull

The pull mechanism allows a MISP instance to discover available (and accessible as defined by the distribution rules) events on a connected instance and download any new or modified events.
During the entire synchronisation procedure, events are converted into a JSON representation for transfer, which consists of a set of events with the associated meta data. A quick run-through of the major logical steps of the algorithm is as follows (additional tasks such as access right checks are omitted for simplicity's sake):

1. Create a filter list based on the synchronisation filter rules to be passed to the remote instance.

2. Request the JSON output of the event index from the remote instance and pass along the generated filter parameters.

3. The remote instance will generate this list taking into account any filter rules that the administrators of the remote instance may have created to filter the data outgoing to the instance that, in this case, is initiating the pull. This means that the list of events ends up being filtered by both the content consumer and the content provider.

4. Compare each event by its UUID to a potentially existing local copy. If no local copy exists or the local copy is out of date, add the UUID to the list of events to be pulled.

5. For each of the events to be pulled do the following:

   (a) Fetch the event JSON using its UUID from the remote instance.

   (b) If a local version of the event already exists, do the following as an edit, if not as a new event creation.

   (c) Capture or update the related objects (such as related tags, sharing groups, organisations involved with either the event directly or the attached sharing groups, etc.).

   (d) Save each of the attributes attached to the event. If an event is being edited, update attributes with the new data only if the local version is older.

   (e) Finally publish the event, which will notify users and propagate it further to interconnected instances (if applicable according to the event distribution settings and the synchronisation rules of the instance).

6. Once all events have been pulled, the second phase of the synchronisation begins, the synchronisation of proposals.

7. Request a JSON containing all proposals from the remote instance.

8. The remote instance will compile a JSON with all proposals that have been made to events visible to the requestor instance and return it.

9. Loop through each proposal and do the following:

   (a) Check if the proposal already exists locally. If it does and the local version is not outdated then the next proposal is processed.

   (b) If the proposal does not exist locally, a new proposal will be created, otherwise the existing proposal gets edited.

   (c) Capture or update the creator organisation of the proposal.

   (d) Once a proposal is saved, members of the event creator organisation are notified via e-mail.

10. If no more proposals are left to be processed then the pull procedure terminates.

3.4.2 Push

The push mechanism of MISP allows one instance to convert an event or a list of events to a JSON format that is then sent to a remote instance. This can be triggered either by initiating a full push of all applicable events to a single instance or simply by publishing an event, which would trigger a push for that specific event alone, but to all connected and eligible instances. The algorithm works as follows:

[4] https://github.com/MISP/misp-taxonomies

1. Fetch the version number from the remote instance and if the remote instance is at least a minor version behind, block the push and log an error message. MISP cannot ensure that the remote instance is capable of handling the event.

2. Generate a list of events that are eligible to be pushed to the remote instance (based on the distribution settings and the filter rules on the synchronisation link).

3. Iterate through each of the event IDs that are eligible, convert them to MISP's JSON format and POST them to the event creation API of the remote end.

4. At this point, there are several possible outcomes for the POST request:

   (a) If the event does not exist on the remote end and can be created, the remote instance returns the newly created event and the push of the next item commences.

   (b) The event already exists and can be edited, the remote side will match the event by UUID to a local event and return the URL that is to be used to update the event. The instance initiating the push can then push the event to the new URL which will result in an event edit.

   (c) The remote instance blocks the event (for example if the event is already up to date, is blocked by a local filter or blacklist, etc.)

5. If an event is saved, the remote MISP will capture all related objects and create them locally or update any eligible objects (organisations, sharing groups, tags, proposals) that exist already.

6. After saving an event, be it from a creation or an update, applicable users will get notified about it by e-mail and MISP will initiate a push towards each interconnected instance that is eligible for the event.

3.4.3 *Cherry Picking*

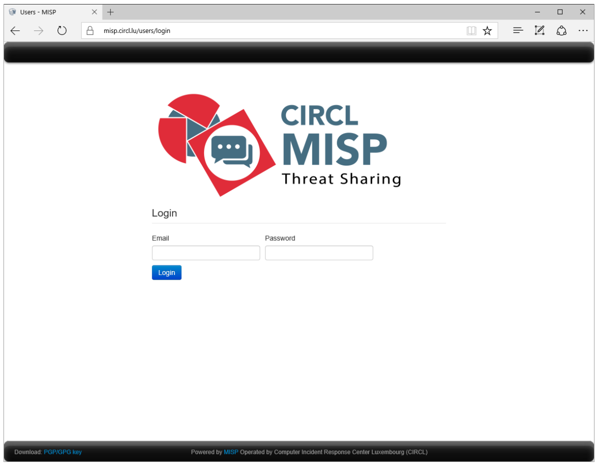
MISP also provides an alternate pull method that allows users to selectively pick and choose events that should be pulled to the local instance. To facilitate this, administrators can browse interconnected instances using a similar UI to the local event index, explore individual events using a view similar to the event view and download specific events. The actual mechanism for fetching events this way is the pull mechanism described earlier, but with an event ID set as a target parameter.

Since this creates an issue in regards to keeping the cherry picked events up to date, a sync update function allows administrators to restrict the data pulled to a subset containing only events that already exist locally, ignoring all new events. This again uses the default pull mechanism, but all event UUIDs that do not exist locally get discarded during the filter process.

3.4.4 *Feed system*

The synchronisation system works well for interconnected MISPs, but there are scenarios when a direct link between MISP instances is not feasible (for example when dealing with air-gapped systems) and in some cases content providers might want to share their data either indirectly to clients or open it up to a wider audience. To support these use-cases, MISP has a built-in Feed functionality.

A configurable feed generator script generates a dump of the selected events in separate JSON files along with a manifest file that includes the metadata of each event contained in the feed dump. The output can be simply served via a web server and other MISP's can browse the contents via the UI similarly to how the cherry picking works. Administrators can then choose to pull the feed, create filter rules to pull a subset of the feed or simply cherry-pick data that they deem useful for their instance. Alternatively for air-gapped systems, the feed dump can be distributed out-of-bound and served locally by the recipient for ingestion by their own internal, air-gapped MISP.



Download: PGP/GPG key Powered by MISP Operated by Computer Incident Response Center Luxembourg (CIRCL)

Figure 2: The login screen of MISP

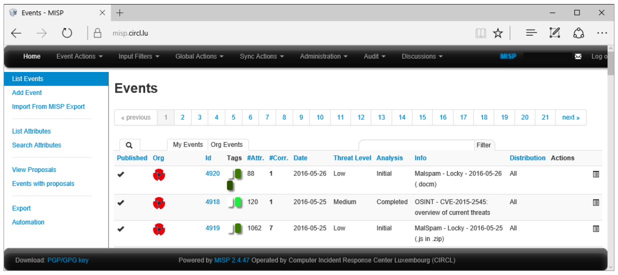
4. MISP - THE TOOL

The following section briefly describes the interface of MISP and provides some additional information. Figure 2 shows the login interface to the MISP Platform that can be accessed by the link **https://mispriv.circl.lu**, but new users need to register at CIRCL first to get access to the platform. The platform is meant for private sector companies including; ICT, antivirus, industrial, financial and other sectors [4].

The index page, similar to a dashboard, represents a relevant part for the MISP user experience. It shows an index of all recent activities and events that were submitted including the corresponding status. Figure 3 shows the index page after login. It regroups the different events and provides a menu bar to the user to select actions, such as add an event, list attributes, export information, etc. An extended user guide [13] describes the use of MISP and explains the different steps to share information on the platform.

5. USAGE AND STATISTICS

In the following section, some statistics and usage will be presented. The numbers presented in Table 1 reflects a snapshot from 16th June 2016 of one MISP instance dedicated to the private sector [4] regrouping mostly private companies willing to share information.



Events

Published	Org	Id	Tags	#Attr.	#Corr.	Date	Threat Level	Analysis	Info	Distribution	Actions
✓		4920		88	1	2016-05-26	Low	Initial	Malspam - Locky - 2016-05-26 (.docm)	All	
✓		4918		120	1	2016-05-25	Medium	Completed	OSINT - CVE-2015-2545: overview of current threats	All	
✓		4919		1062	7	2016-05-25	Low	Initial	MalSpam - Locky - 2016-05-25 (.js in .zip)	All	

Download: PGP/GPG key   Powered by MISP 2.4.47 Operated by Computer Incident Response Center Luxembourg (CIRCL)

Figure 3: The MISP events index page - the default view after login

N	Description	Number of instances
1	Events	3 769
2	Attributes	421 868
3	Correlations found	151 209
4	Proposals active	36 569
5	Users	797
6	Organisations	409
7	Discussion threads	159
8	Discussion posts	280

Table 1: MISPpriv sharing information in the private sector

It can be observed that on that date more than 3 700 events have already been created in the MISP database. These events refer to a set with more than 420k attributes. It can also be observed that this large number of events is generated out of a community of 400 organisations.

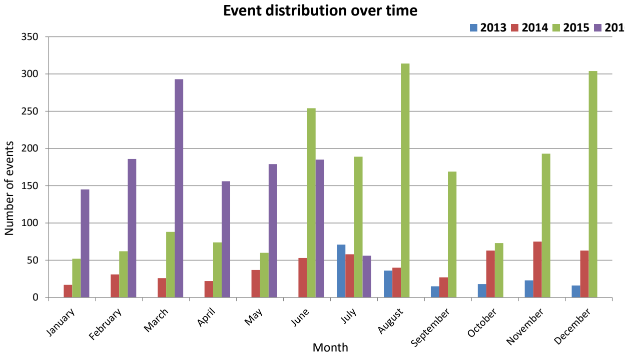


Figure 4: Distribution of events per month from 2013 to 2016

In 2013 and 2104, 50 events per month were quite common. In 2015 and 2016, these rates significantly increased to peak rates of 300 events per month. Figure 4 shows that over time, MISP has become more popular and more people and organisations are ready to share IoCs and other relevant threat information.

Figure 5 shows the number of attributes affiliated to an event. The number of attributes for an event is not fixed, but adaptive. The user can choose for himself the number of attributes, depending on the state of the event or knowledge about the event, for a precise description. This explains the variation of attributes for the events.

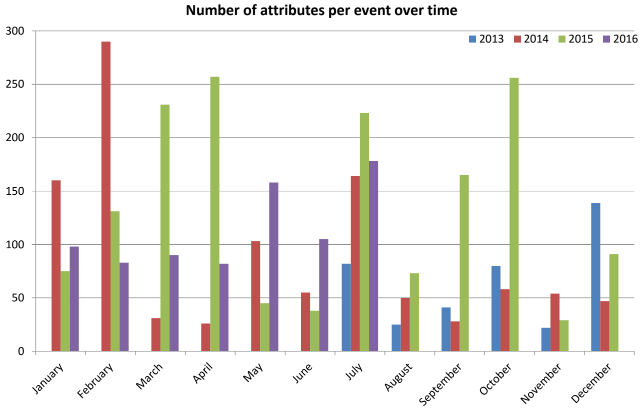


Figure 5: Average number of attributes per event per month

Attribue type	frequency
md5	99446
hostname	67313
ip-dst	40040
sha256	33887
sha1	26501
domain	25761
url	23585
link	21441
ip-src	137277
filename	3804
filename|sha256	3683
filename|sha1	3620
text	3614
malware-sample	3475
mutex	3452
comment	2003
filename|md5	1486
email-src	912
yara	678

Table 2: Top 20 used attribute types

In the MISP instance for the private sector [4], 65 attribute types are used. Most of the threats are related to malware such as hashes and host names, helping users to detect malware samples.

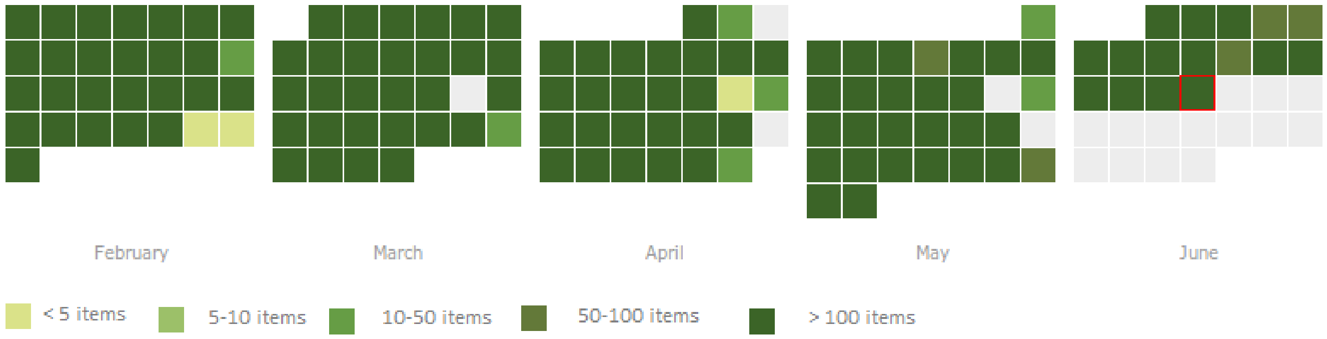


Figure 6: Misppriv activity heat map until June 16, 2016

.

Due to space constraints not all of them can be described in detail, but Table 2 shows the top 20 attributes used to describe an event.

However, additional requests of attributes can be submitted to the open source development community[5]. Recently, non technical attributes are emerging such as IBAN numbers and other information about threat actors. For example, IBAN numbers of money mules involved in financial abuse are shared. These IBAN numbers are mainly interesting for banks and accountants, who could block or check wire transfers to these accounts often executed by attackers using financial malware.

In order to show the large usage of CIRCL's MISP private sector on a daily base, a heat map of the activities in MISP is represented in Figure 6. This heat map shows the overall activity of MISP for a period of 4.5 months, from February to June 16th, 2016. Each calendar day is represented by a square in a green color.

The five different gradients of green color represent the number of instances added to MISP on a given day. The lightest gradient of green represents less than 5 items added a day. The next one, 5 to 10 items, followed by 10 to 50 items. The second darkest green represents 50 and 100 items and the last, the darkest green, more than 100 items that were added on a day.

From the heat map can be concluded that the MISP instance is continuously used during 2016 with some exceptions. For squares represented in gray there are no events existing. Less activity can be observed end of March for the Easter weekend. The same can be said for the weekend of May 1st, which is a national holiday in most european countries and for the weekend of Whitsun.

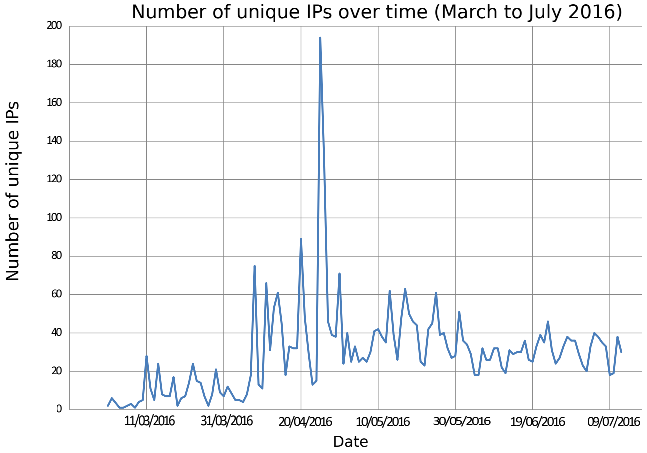


Figure 7: Distribution of MISP installations per day

When a MISP server is installed, the instance does not include any information that could be shared and therefore starting out for new users is often hard. To ease the usage of MISP, CIRCL provides a feed of events that can be easily shared; such as OSINT events and/or attributes that are classified as TLP:white[6], unclassified information that can be distributed without any restrictions. Hence, the information in this feed is already on the Internet.

Figure 7 represents the number of unique IP addresses that installed MISP on a daily base. In general, 20 to 40 unique IP addresses can be observed daily. The peaks can be explained by the fact that MISP was discussed on twitter and a large armada of bots tried to access the feed.

6. FUTURE WORK AND CONCLUSIONS

Nowadays, sharing information has become a precious resource of information within the IT-community, but not restricted to, since attackers share information among their peers too, therefore it is essential as an IT-community to share information in order to stay informed on new emerging threats.

[5] https://github.com/MISP/MISP

[6] TLP: Traffic Light Protocol, is a protocol for the classification and distribution level for sensitive information

In this paper, a threat intelligence sharing platform has been presented, where users from the IT community and other communities at large, can share their information on incidents or other artifacts in a trusted environment.

Future work is manyfold. In a future iteration process, the MISP replication and synchronization protocol will be analyzed for its efficiency. Another step is the information quality of the shared information, respectively information classified as false-positives or false-negatives.

To deal with these issues, MISP should not only be a vast platform with information, but also include quality requirements, therefore, future work is the implementation of a correlation evaluation system. A possible quality evaluation method could be for example scoring from the crowd [16].

In order to evaluate the large datasets that are generated by MISP, data mining techniques for structured data can be used in a future step to evaluate the shared information efficiently to observe local trends and improve MISP.

MISP is a tool that should meet the permanently changing and evolving requirements of the IT-community and should be considered a useful support for incident analysis, mitigation and response and thus evolve over time.

7. ACKNOWLEDGMENTS

We would like to express our deepest acknowledgments to all involved parties for supporting this initiative and their motivation to actively contribute to the evolution of the Malware Information Sharing Platform initiative. Special thanks to Raphaël Vinot and the whole CIRCL-team for reviewing this paper.

8. REFERENCES

[1] S. Appala, N. Cam-Winget, D. McGrew, and J. Verma. An actionable threat intelligence system using a publish-subscribe communications model. In *Proceedings of the 2Nd ACM Workshop on Information Sharing and Collaborative Security*, WISCS '15, pages 61–70, New York, NY, USA, 2015. ACM.

[2] S. Barnum. Standardizing cyber threat intelligence information with the structured threat information expression (stix). Technical report, MITRE Corporation, 2012.

[3] S. Brown, J. Gommers, and O. Serrano. From cyber security information sharing to threat management. In *Proceedings of the 2Nd ACM Workshop on Information Sharing and Collaborative Security*, WISCS '15, pages 43–49, New York, NY, USA, 2015. ACM.

[4] CIRCL. Misppriv. https://misppriv.circl.lu, 2016.

[5] L. Dandurand and O. Serrano. Towards improved cyber security information sharing. In *Cyber Conflict (CyCon), 2013 5th International Conference on*, pages 1–16, 2013.

[6] R. Danyliw, J. Meijer, and Y. Demchenko. The incident object description exchange format, 2007. IETF, RFC5070.

[7] E. Deci and R. M. Ryan. Intrinsic motivation and self-determination in human behavior. In *Perspectives in Social Psychology*, pages 11–40, 1985.

[8] J. C. Haass, G.-J. Ahn, and F. Grimmelmann. Actra: A case study for threat information sharing. In *Proceedings of the 2Nd ACM Workshop on Information Sharing and Collaborative Security*, WISCS '15, pages 23–26, New York, NY, USA, 2015. ACM.

[9] U. Helmbrecht, S. Purser, G. Cooper, D. Ikonomou, L. Marinos, E. Ouzounis, M. Thorbrugge, A. Mitrakas, and S. Capogrossi. Cybersecurity cooperation: Defending the digital frontline. Technical report, ENISA, October 2013.

[10] J. Hernandez-Ardieta, J. E. Tapiador, and G. Suarez-Tangil. Towards improved cyber security information sharing. In *Cyber Conflict (CyCon), 2013 5th International Conference on*, pages 1–28, 2013.

[11] C. Johnson, L. Badger, and D. Waltermine. Guide to cyber threat information sharing [draft]. Technical report, NIST, April 2016. NIST Special Publication.

[12] M. Maasberg, M. Ko, and N. L. Beebe. Exploring a systematic approach to malware threat assessment. In *49th Hawaii International Conference on System Sciences (HICSS)*, pages 5517–5526, 2016.

[13] MISP-Contributors. User guide of misp malware information sharing platform, a threat sharing platform. https://www.circl.lu/doc/misp/book.pdf, 2016.

[14] K. Moriarty. Real-time inter-network defense (rid), 2012. IETF, RFC6545.

[15] S. Murdoch and N. Leaver. Anonymity vs. trust in cyber-security collaboration. In *Proceedings of the 2Nd ACM Workshop on Information Sharing and Collaborative Security*, WISCS '15, pages 27–29, New York, NY, USA, 2015. ACM.

[16] M. Noll and C. Meinel. Design and anatomy of a social web filtering service. In *Proceedings of the 4th International Conference on Cooperative Internet Computing*, CIC, pages 35–44, 2006.

[17] T. Sander and J. Hailpern. Ux aspects of threat information sharing platforms: An examination and lessons learned using personas. In *Proceedings of the 2nd ACM Workshop on Information Sharing and Collaborative Security*, WISCS '15, pages 51–59, New York, NY, USA, 2015. ACM.

[18] F. Skopik, G. Settanni, and R. Fiedler. A problem shared is a problem halved: A survey on the dimensions of collective cyber defense through security information sharing. *Computers & Security*, 60:154 – 176, 2016.

[19] A. Straup Cope. Machine tags. flickr. https://www.flickr.com/groups/api/discuss/72157594497877875/, 2007.

[20] B. Woods, S. Perl, and B. Lindauer. Data mining for efficient collaborative information discovery. In *Proceedings of the 2Nd ACM Workshop on Information Sharing and Collaborative Security*, WISCS '15, pages 3–12, New York, NY, USA, 2015. ACM.