Collaborative repository for cybersecurity data and threat information

Jean Lorchat
Internet Initiative Japan
Tokyo, Japan
jean@iijlab.net

Cristel Pelsser
Internet Initiative Japan
Tokyo, Japan
cristel@iijlab.net

Romain Fontugne
National Institute of Informatics
Tokyo, Japan
romain@nii.ac.jp

***Abstract*—In this paper, we attempt to show how to build a collaborative repository for cybersecurity data and threat information by building on top of a privacy-aware storage system: Tamias. We set the following goals: allow data sharing with a very high level of control over the sharing scope, enhance collaboration of entities that may not know each other but deal with similar threats, and manage different levels of trust between each parties. These levels of trust will define how much information is shared with each entity.**

I. Introduction

The Internet has radically changed the way that people communicate. By providing a worldwide, almost unregulated, information exchange system, it has made possible a large number of innovations, with impact comparable to that of the first newspaper, radio or television broadcasts.

However, the largely unregulated nature of the Internet and the fact that users and corporations alike interconnect their own information systems to this network, has also made it a playing ground of unprecedented size for criminals. Certainly, there are existing countermeasures. But similarly to the arms race existing within the military industry, better defense tools are soon followed by more sophisticated threats.

Of course, many entities are devoted to the study of these threats and the gathering of cybersecurity-related information. However, because of the trust issues raised by the impersonal nature of communications on the Internet, these entities hardly exchange any sensitive data. A Cybersecurity Emergency Response Team (CERT) might provide information about an attack against its network, after redacting the details of the exact targets. By doing so, it makes sure that malicious parties don't try to exploit those victims further. But this might also prevent other operators from getting crucial information about potential *victims-turned-sources* of subsequent attacks.

Also, in many cases, datasets collected by entities monitoring security information are very large. Within this data, one must look for both known and unknown threats. Each entity might have different algorithms to spot these threats, but given the large amount of data, it is likely that some attacks are not detected. So even sharing aggregated information about detected threats might be insufficient in certain cases.

Finally, defense against cyber-threats requires collaboration among defense entities because it is hard to get a global view from within a single entity. But parties might not want to share all their data with all their peers, for the trust reasons we have mentioned previously, lowering the value of the collaboration.

In this paper, we will have a look at the existing threat data exchange systems. We will then show how the Tamias [1] distributed storage system can be leveraged to build a collaborative repository for cybersecurity data and threat information. We will first introduce the Tamias privacy-aware distributed storage system, then detail each of the principles of the system and how they can be adapted to build our repository: identities, stored objects and federation.

II. Related Works

The need for sharing information in the context of cybersecurity has always been evident. In this sense, there are numerous recommendations and a few designs that try to tackle this problem.

In [2], several suggestions are described for efficient data sharing among CERTs, however the approach favored by the authors is to use secure messaging systems such as PGP. While this is an obvious choice for privacy and trust, it is cumbersome to exchange large amounts of data. Also, it does not allow to revoke access when the threat is gone.

Another work [3] known as *fordrop* (standing for Forensics Dropbox) proposes an architecture for a social network about threats that is based on the XMPP [4] and ActivityStreams [5] standards and allows participants to publish information about malware detected in their networks. However, this is especially restricted to non-sensitive information, so that it does not have to deal with trust and privacy issues.

On the other hand, standardization organizations have also started working on sharing threat data, in order to define exchange formats that can be used by all participants. In the IETF, there has been work on the IODEF [6] format for incident description. This standard only defines the exchange format, and thus does not specify how various entities can actually exchange information. At the ITU-T, work is still ongoing on the CYBEX [7] X.1500 standard, that will include the definition of both the exchange format and overall architecture.

III. The Tamias Distributed Storage

Tamias is a distributed storage system built on top of Tahoe-LAFS [8]. It adds identity and access capabilities management, fine-grained sharing, access capability scoping, and identity-related services. Tahoe-LAFS consists of a peer-to-peer network of storage nodes. Each storage node hosts indexed buckets of encrypted data. Prior to storage, objects

are encrypted on the local client with the key being based on the hashed contents. In this system, knowing the storage index allows to retrieve encrypted contents, and knowing the key allows to decrypt the object. In this Tahoe-LAFS system, access is granted by sharing a capability consisting of the storage index and encryption key. Thus the name of Least Authority File System, because knowing the access capability grants the right to share it further automatically.

In order to scope those access capabilities, Tamias introduces an identity for each storage user. This identity is made of a private/public keypair. The public key is shared with other parties and allows to authenticate messages received, as long as a proper introduction was made beforehand. Then, this public-key is associated with each bucket that the storage client uses. Furthermore, the storage server will not serve a block to anyone else than the genuine client, or a client that can show an access authorization signed by the actual owner of the bucket.

This access authorization is made of several pieces of information, such as the storage index to which the authorization relates, an expiration date, and a target identity. The whole is protected by a signature, precluding forgery. By using these access authorizations, it is possible to share different levels of information to different groups, at various trust levels. For example, partners bound by a NDA might receive access authorizations for raw data, whereas occasional partners only get anonymized data, and another lower level of trust would only gain access to aggregate results.

Finally, in order to ease the identity bootstrapping process, Tamias provides a globally writable object known as the phonebook. While everyone can publish to the phonebook, each entry is signed by the user public-key. It is thus possible for the entity to publish information about itself, such as identity details, website, and so on. Actually, any kind of information could be posted to the phonebook, since the phonebook leverages RDF [9] to provide semantics.

The phonebook is an example of service that leverages identities. There are other that are directly integrated with the Tamias client. The inbox is another such service. For example, if user Arthur and user Brutus trust each other, Arthur can create an inbox dedicated to Brutus, and provide him the access authorization to this object. Brutus can then write to Arthur about new access authorizations, or any other RDF-based information. The last integrated mechanism leveraging identities is the public inbox. This is an inbox that is published by Arthur in the phonebook and is writable by anyone. It allows users that Arthur does not trust to write to him, for example in order to self-introduce themselves to Arthur or solicit access to specific data.

Relying on Tamias' scoped sharing, identity properties and identity services, we propose to build a collaborative repository of cybersecurity data. Using access authorizations, entities can have a very fine-grained control of what is being shared, and to whom. It is also possible to use the phonebook mechanism to publish general and public information about ongoing threats and trigger collaboration with interested parties.

A. Identity Principles

We have explained in the previous subsection, that in the Tamias storage system, each participant is defined by his identity. For the purpose of building trust specifically for sharing threat data information, offline exchange seems to be the most trustworthy way. For example, if two entities already have a trusted means of communication, they can use it for this exchange of public keys.

On the other hand, in a sort of *friend of a friend* fashion, participants can introduce their trusted peers to each other. This is an integrated feature of Tamias (see Section III). Finally, we also propose to extend the phonebook for reputation building. With additional RDF grammar elements, it becomes possible for an entity to publish a recommendation about another one. By summing all those recommendations, an interested entity can evaluate the reputation of a new partner. This provides another metric for entities when choosing which partners to trust.

B. Storage principles

In the Tamias storage system, it is possible to store objects of arbitrary size. Once inside the storage, these objects can not be easily leaked to the outside. Indeed, stealing the access authorization of another entity is not sufficient to gain access. Knowing the storage index, or even the encryption key, does not grant access to the buckets.

In order to be able to fetch a block from a storage server, a Tamias client must show an access authorization that was signed by the actual owner of the file. The owner of the file is the one whose public key has been recorded with the block when it was first stored. Also, the access authorization itself has a time limit, so that any authorization eventually expires. Finally, block owners can revoke access authorizations directly at the storage server level, if they want to stop an access authorization before it expires.

In addition, since the storage network is distributed, access to the data can be much faster, provided that the distribution is consistent with respect to locality. Indeed, an increased number of storage servers brings an increased number of resources, thus making the system faster as it grows.

C. Federation principles

The global Tamias storage system in itself has been designed to work as a federation of small Tamias storage networks for scalability purposes. Using a federation of Tamias networks allows each entity to host the datasets that it collected and wants to be able to share. This way, it prevents the entity from suffering from quota limitations in the storage network.

In addition to this, entities can now trust whole networks. By doing this, each entity builds its own view of a global repository where the information sources are limited to the partners that they trust. Then, as for identities of separate entities, it is possible to recommend networks to each other, thus controlling the scope of information and the coverage of the datasets.

IV. Preparing Tamias for threat data

In this section, we introduce our proposal to build a collaborative repository based on Tamias to share threat data.

A. Data Semantics

One advantage of the Tamias storage system, is the fine-grained sharing mechanism that allows to specify which user can access which data in a very detailed way. As explained in section III, access authorizations are sent to intended recipients through their shared inbox using RDF tuples.

In addition to sharing messages, we propose to define the following messages that describe datasets. They can be sent to the intended user at the same time as the access authorization itself.

Properties
: This allows to describe the properties of the dataset itself. For example, what kind of token can be found inside: IP addresses, malware signatures, packets, URLs, time range, and more. But it could also be a reference to file types, e.g. pcap, sflow, netflow, rfc822.

Analysis results
: By attaching analysis results to the dataset, an entity can tag its datasets for easier collaboration. This way, a partner can try to look for datasets that have specific results associated, such as traces of NTP attacks, Zeus Botnet activity, etc...

Standards
: An entity will use this kind of message to specify that an object is described by an information exchange standard. It could be IODEF [6], CYBEX [7], or the n6 [10] JSON format among others.

Alternate view
: This type of message allows to offer another view of the dataset. Depending on the level of trust (see subsection IV-B) between the sharing parties, an alternate view might be the only view available. For example, it might refer to an anonymized version of the dataset.

Besides writing to intended recipients, users can also choose to publish dataset-related information directly to the public repository (the phonebook), or to a message box shared by a task-force group.

B. Trust levels

For the purpose of easy and safe sharing of threat data within our repository, we propose to define several trust levels. The granularity of those levels is defined by the user, because it depends on the kind of data and the relationships he maintains with the other users of the system. For example, let's consider a dataset of packets coming from the sampling of a link traffic. We can define the following levels of trust, from the highest to lowest:

High
: Sharing at this level grants access to the raw dataset to the recipient. However, it is not exactly similar to providing a copy of the dataset because the access authorization will eventually expire.

Moderate
: Sharing at this level grants access to an anonymized version of the dataset. All details that can identify a specific victim or person of interest in the dataset have been altered.

Low
: Sharing at this level only provides information about a summary of the dataset. For example, it could be the output of the various anomaly detectors that have been run on the dataset.

Least
: Sharing at this level will provide only information about specific threat information. For example, the IP source of an NTP attack, or a list of malware signatures that have been spotted in the dataset. This might allow to advertise the dataset to unknown parties before starting the identity exchange process.

C. Collaboration

In order to foster collaboration, it is important to go beyond the usual partners of an entity and allow entities to discover each other in the context of a specific threat.

For that purpose, we propose the creation of a distributed journal of recent information. It is a message box similar to the phonebook, but hosting information with a short lifetime. Each entity can then publish, to this journal, any information that is related to its current situation. It could be details of an ongoing threat, or request for specific information. Other entities can then look at this journal and find out if they can relate to this information.

Another important feature that would enhance collaboration is the creation of short-lived groups. These groups can be created quickly and announced to the journal described above. It allows to share directly with all the members at a given trust level. Some examples of groups might be "DNS reflection attack victims club", but also "Botnet XYZ take-down coordination center". These groups eventually disappear when the related activity is concluded.

Then, pursuing the publish/subscribe concept, we propose a journal subscription application. This application runs locally within the Tamias client and parses the journal updates to look for any kind of information about which the user has specified interest. The user can describe his subscriptions with full-text search in new messages, or using keywords in conjunction with the semantic attributes that underlie the journal itself.

D. Sharing lifetime

Finally, an important property of the Tamias storage system is that access authorizations always expire. This helps to bring a sense of safety to all the players because they can feel confident that the data can not be misused at a later date. In addition to this, we propose to extend the client revocation mechanism with triggers. Whenever an access authorization is created, the user has the opportunity to attach it to a trigger. If this trigger is activated, the client will automatically revoke access to all the authorizations involved. For example, Arthur

might choose to share its raw traffic sampling traces in the context of a task force investigating NTP-amplification attacks. Upon sharing, he decides to create an associated trigger that he calls *end of the NTP threat*. At a later date, when the NTP problem is dealt with, he just has to enable this trigger, at which point the client revokes access to the raw data everywhere it was shared with this trigger.

E. Examples

Various trust levels and notifications

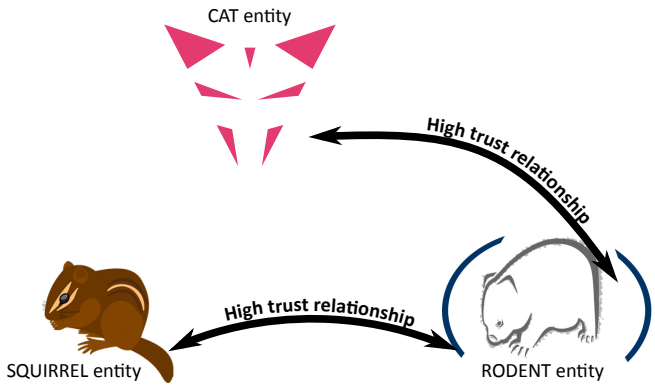

Fig. 1. Trust relationships as they stand in the first example scenario.

All the principles we have detailed here are illustrated in Figures 1–3. In this example, we have three entities participating in the repository. As seen in Figure 1, the entity on top is the CAT entity, while we have SQUIRREL on the bottom left and RODENT on the bottom right. RODENT trusts both SQUIRREL and CAT, but SQUIRREL and CAT do not know each other.

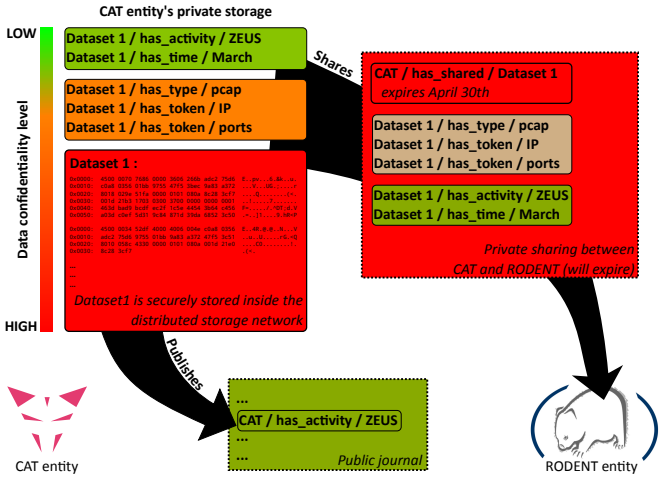

Fig. 2. The CAT entity stores Zeus-related activity in its private storage space, publishes summarized related information to the phonebook and shares it entirely with RODENT.

Now, we can see in Figure 2 that CAT owns a dataset that it shared with RODENT. Since they trust each other, CAT provided a large description about the dataset. It included the type, time coverage, a list of tokens and the results of an analysis for Zeus activity. Also, CAT decided that in order to fight the Zeus botnet, it would publish summary information informing all participants that it has traces of Zeus activity, without any details about the dataset though.

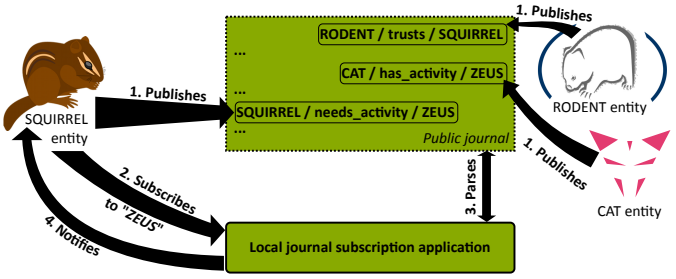

Fig. 3. The SQUIRREL entity publishes its needs, it also subscribes to Zeus-related updates and gets a notification.

Meanwhile, in Figure 3, SQUIRREL has published information about its needs. Namely, people from the SQUIRREL entity are looking at Zeus activity and are thus eager to find more information about it. They publish this requirement to the public journal. From then on, there are several possibilities. By subscribing to Zeus information, SQUIRREL gets a notification about the message that CAT has published. Otherwise, SQUIRREL might also look at the journal and realize that CAT has interesting data. Although they do not know each other, RODENT actually trusts CAT and could serve as third-party for introduction. Conversely, if CAT proactively looks for more Zeus information, it might notice SQUIRREL request and propose to establish a trust relationship, by seeing that RODENT trusts SQUIRREL.

From this example, it appears that there are many ways to take advantage of this system, but also, that the system is not fully automated. This is an important point if entities want to be sure that nothing happens outside of their control. Even though CAT and SQUIRREL have a common friend, no sharing happens before CAT decide to do so. If CAT chooses not to trust SQUIRREL, notwithstanding that RODENT trusts it, it can opt to not trust SQUIRREL nor share data with it.

Anonymization and fine-grained sharing

In Figure 4 we show how our system can be used to provide different views of the dataset to entities that have different trust relationships. In this example, we build upon the previous example and assume that the CAT entity has decided to mildly trust SQUIRREL, after it made sure that RODENT was actually trusting SQUIRREL as well. In this situation, we see that the same dataset is shared to both entities, however SQUIRREL receives much less information compared to what RODENT already had. This is because the anonymized dataset has jumbled IP addresses and ports. For this reason, the present tokens specified by CAT are not present in the sharing message with SQUIRREL.

In this situation, we assumed that CAT already had an anonymized version of the dataset, where IP addresses and ports have been transformed to protect the actual values. In the future, it might be possible to provide application plugins that would be able to anonymize data prior to sharing with targets

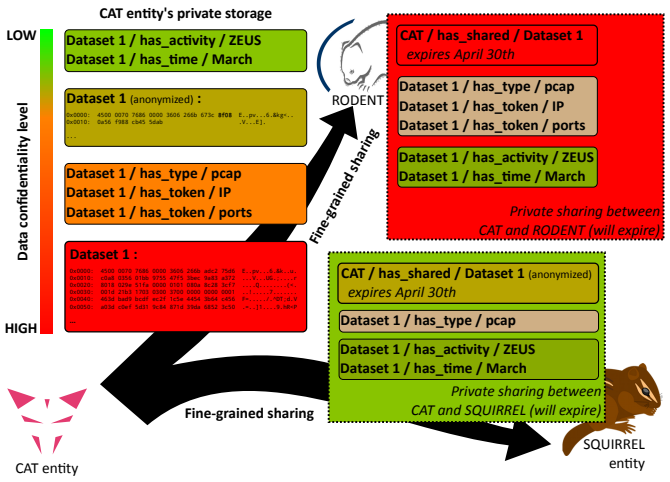

Fig. 4. The CAT entity decides it can trust SQUIRREL thanks to BADGERS endorsement, however it only grants partial access to the data by providing the anonymized version of the dataset.

such as SQUIRREL that do not satisfy the required trust level for full access to the data. This kind of plugin would have to be prepared once for each kind of threat data standard though.

V. Conclusion

We have proposed a system based on the Tamias distributed storage to efficiently share large amounts of threat data. Our solution enables fine-grained sharing of threat data with a per-destination control of the amount of information provided. Access policy is based on the identity of each destination and its trust level as perceived by the owner of the threat data.

While this is a work in progress, we can already anticipate that it will address some very important problems such as provisioning trust into the threat data exchange system, limiting the scope of sharing in both time and space, helping the discovery of related datasets, and providing basic collaboration tools for information sharing.

Our future work will of course include a prototype open-source implementation based on the existing Tamias code, which will lead to more detailed work on various aspects of this specification. Nevertheless, we expect to reach interesting results thanks to the unique and powerful nature of the underlying Tamias storage system.

Acknowledgments

This research has been supported by the Strategic International Collaborative R&D Promotion Project of the Ministry of Internal Affairs and Communication, Japan, and by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 608533 (NECOMA).

References

[1] J. Lorchat, C. Pelsser, R. Bush, K. Shima, H. S. (IIJ), and L. J. (SUNET), "TAMIAS: a distributed storage built on privacy and identity," in *The 28th Trans European Research and Education Networking Conference, 21 - 24 May, 2012, Reykjavik, Iceland*, May 2012.

[2] ENISA - European Union Agency for Network and Information Security, "Detect, SHARE, Protect - Solutions for Improving Threat Data Exchange among CERTs," Nov 2013, https://www.enisa.europa.eu/activities/cert/support/data-sharing.

[3] J. Berggren, "Social CERT," The 28th Trans European Research and Education Networking Conference, 21 - 24 May, 2012, Reykjavik, Iceland, May 2012.

[4] XMPP Standards Foundation, "The Extensible Messaging and Presence Protocol (XMPP)," http://xmpp.org.

[5] Activity Streams Community, "A format for syndicating social activities around the web," http://activitystrea.ms/.

[6] R. Danyliw and J. Meijer and Y. Demchenko, "RFC5070 - The Incident Object Description Exchange Format," http://www.ietf.org/rfc/rfc5070.txt.

[7] A. Rutkowski, Y. Kadobayashi, I. Furey, D. Rajnovic, R. Martin, T. Takahashi, C. Schultz, G. Reid, G. Schudel, M. Hird, and S. Adegbite, "Cybex: The cybersecurity information exchange framework (x.1500)," *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 5, pp. 59–64, Oct. 2010. [Online]. Available: http://doi.acm.org/10.1145/1880153.1880163

[8] Z. Wilcox-O'Hearn and B. Warner, "Tahoe: the Least-Authority FileSystem," in *Proceedings of the 4th ACM international workshop on Storage security and survivability*. ACM, 2008, pp. 21–26.

[9] W3C/RDF Working Group, "The Resource Description Framework (RDF)," http://www.w3.org/standards/techs/rdf.

[10] CERT Polska, "n6 - network security incident exchange," http://www.cert.pl/projekty/langswitch_lang/en.