Towards Automated Threat Intelligence Fusion

Ajay Modi, Zhibo Sun, Anupam Panwar, Tejas Khairnar, Ziming Zhao, Adam Doupé, Gail-Joon Ahn
Arizona State University
{aamodi, zsun41, apanwar4, tkhairna, zmzhao, doupe, gahn}@asu.edu

Paul Black
AllState
Paul.Black@allstate.com

***Abstract*—The volume and frequency of new cyber attacks have exploded in recent years. Such events have very complicated workflows and involve multiple criminal actors and organizations. However, current practices for threat analysis and intelligence discovery are still performed piecemeal in an ad-hoc manner. For example, a modern malware analysis system can dissect a piece of malicious code by itself. But, it cannot automatically identify the criminals who developed it or relate other cyber attack events with it. Consequently, it is imperative to automatically assemble the jigsaw puzzles of cybercrime events by performing threat intelligence fusion on data collected from heterogeneous sources, such as malware, underground social networks, cryptocurrency transaction records, etc. In this paper, we propose an Automated Threat Intelligence fuSion framework (ATIS) that is able to take all sorts of threat sources into account and discover new intelligence by connecting the dots of apparently isolated cyber events. To this end, ATIS consists of 5 planes, namely analysis, collection, controller, data and application planes. We discuss the design choices we made in the function of each plane and the interfaces between two adjacent planes. In addition, we develop two applications on top of ATIS to demonstrate its effectiveness.**

I. Introduction

The volume and frequency of new cyber threats and variants targeting the private sectors have exploded and become critical concerns. In the meantime, government and other public sectors are also facing unprecedented cyber attacks, which may potentially undermine national security and critical infrastructure [19]. It is estimated that the likely annual cost to the global economy from cybercrime is more than $400 billion in 2014 [14]. And, the number is projected to reach $2 trillion in 2019 [15].

The staggering number of cyber crimes are able to evade existing security measures, because they have complicated workflows and involve multiple criminals and organizations. For example, *Try2DDos* is a tool to perform distributed denial of service attack (DDos). It was first released on a French forum *Underground konnekt* in June, 2005. More than one year later, the first public variant of this tool in Spanish appeared on an Argentina hacker forum. From 2005 to 2008, this tool and its variants spread to China, Russia, Guatemala, and Argentina, and many have used it to damage a large number of networked systems [11]. Obviously, to fully understand how *Try2DDos* evolved over the years and who had been distributing this tool requires the combination of binary code analysis, underground social analysis, etc.

However, current practices for threat analysis and intelligence discovery are still performed piecemeal in an ad-hoc manner. Even though a modern malware analysis system can dissect a piece of malicious code [8], it cannot automatically identify the criminals who developed/released it or relate other cyber attack events with it. Consequently, it is imperative to automatically assemble the jigsaw puzzles of cybercrime events by performing threat intelligence fusion on data collected from heterogeneous sources, such as malware, underground social networks, cryptocurrency transaction records, etc.

In this paper, we design an automated threat intelligence fusion framework (ATIS) that can automatically extract all threat intelligence from heterogeneous sources and correlate them. To this end, ATIS consists of 5 planes from bottom to top: i) collection plane, ii) analysis plane, iii) control plane, iv) data plane and v) application plane.

The collection plane is the home of data crawlers. Analysis plane is composed of different analysis modules that only analyze certain types of data. ATIS abstracts each analysis model as a set of $\langle input, output, relationship \rangle$ 3-tuple. A logically centralized controller, which is responsible for automatic data collection and intelligence discovery by orchestrating collection and analysis modules and storing discovered intelligence to data plane, is the brain of ATIS. The data plane is used to store global knowledge base of the collected and generated intelligence. Analysts can define their own business logic and develop corresponding tools that run in the application plane to perform threat analytics based on their own needs.

The main contributions of this paper are:

- We design an automated threat intelligence framework ATIS that consists of 5 planes, namely analysis, collection, controller, data and application planes. The brain of ATIS is control plane that orchestrates modules in other planes to automatically discover new intelligence;

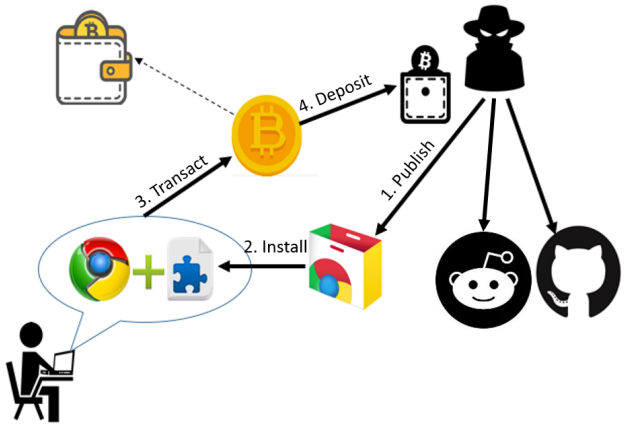

Figure 1. A motivating Example

- We implement a prototype ATIS system that integrates various open source and in-house analysis platforms including the Cuckoo malware analysis platform, an in-house Bitcoin blockchain parser, etc.;
- To demonstrate the effectiveness of ATIS, we develop two applications that rely on ATIS APIs for intelligence discovery.

The rest of this paper is organized as follows. Section II presents a motivating example to demonstrate why holistic analysis that considers heterogeneous data is important to understand cybercrime event. Section III illustrates the system design of ATIS. Section IV describes implementation of ATIS. Section V demonstrates the effectiveness of ATIS by presenting two applications developed on top of ATIS. Section VI describes related work inventing parallel to our ATIS. Section VII concludes this paper.

II. A Motivating Example

In this section we present a motivating example to show the complexity in modern cybercrime process, which clearly shows the necessity of correlating apparently isolated intelligence from heterogeneous sources for the understanding of cyberattack events.

Recently, there were cases of siphoning off bitcoin by changing bitcoin address while pasting at another location [17]. Figure 1 shows how criminals change users' bitcoin deposit address to steal their money. The flow of carried out attack looks as follows: 1. An attacker publishes malicious chrome extension in chrome web store. Recently, BitcoinWisdom Ads Remover extension was tampered and loaded with malicious javascript and published to the chrome web store. 2. A victim downloads and installs add-on from the web store. 3. A victim performs a transaction to transfer bitcoin to a desired genuine bitcoin address. 4. Chrome extension, such as BitcoinWisdom Ads Remover replaces the bitcoin address while loading the DOM or while copying the bitcoin address in browser through javascript code. After successful transaction, bitcoin gets deposited into adversary's account. A careful analysis gives us information about the author who published the extension into chrome web store and revealed information about social profile of the attacker on Reddit and Github. However, there is a possibility that the author's account got hacked and malicious person published the tampered version of extension. In this scenario, heterogeneous sources such as malicious chrome extension, bitcoin addresses and social profiles of threat actor correlate together to complete the story of how an attack was carried out.

III. System Design of ATIS

In this section, we present the design goals that ATIS strives to meet. Then, we overview the high level architecture of ATIS and illustrate the functionality of each plane and the design choices we make.

A. Design Goals

Reuse Existing Analysis Systems. Existing analysis tools are good at analyzing certain types of data. For example, Cuckoo sandbox is the leading open source malware analysis system, which executes malware samples in a simulated environment, monitors system calls and automatically generates detailed static and dynamic analysis reports [10]. BitIodine is a tool for analyzing and profiling to extract intelligence from the Bitcoin transaction records [20]. By reusing existing analysis systems, ATIS will be able to interlink relevant information from heterogeneous sources and discover new intelligence.

Automated Intelligence Fusion. Given the ever increasing volume of threat intelligence, the manual process of extracting key attributes and linking it with relevant data is impossible for human analysts in timely manner. Therefore, automatic processes are desperately needed to help analysts utilize their time for value-added analysis.

Interoperability. Many existing threat intelligence analytics tools represent knowledge in proprietary format, which is cumbersome to share. It is important to integrate standard advanced threat sharing language to support interoperability in sharing of new discovered intelligence. ATIS should follow and support standard formats of representation such as STIX [2], TAXII [7], OpenIOC [16] and YARA [6] to share information.

B. Architecture Overview

Figure 2 shows an overview of the architecture of the ATIS framework, which consists of 5 planes from bottom to top: i) collection plane ii) analysis plane, iii) control plane, iv) data plane and v) application plane. Collection plane consists of autonomous crawlers and their corresponding wrapper/parser. For example, a crawler for malware will crawl from various sources to collect malware samples and

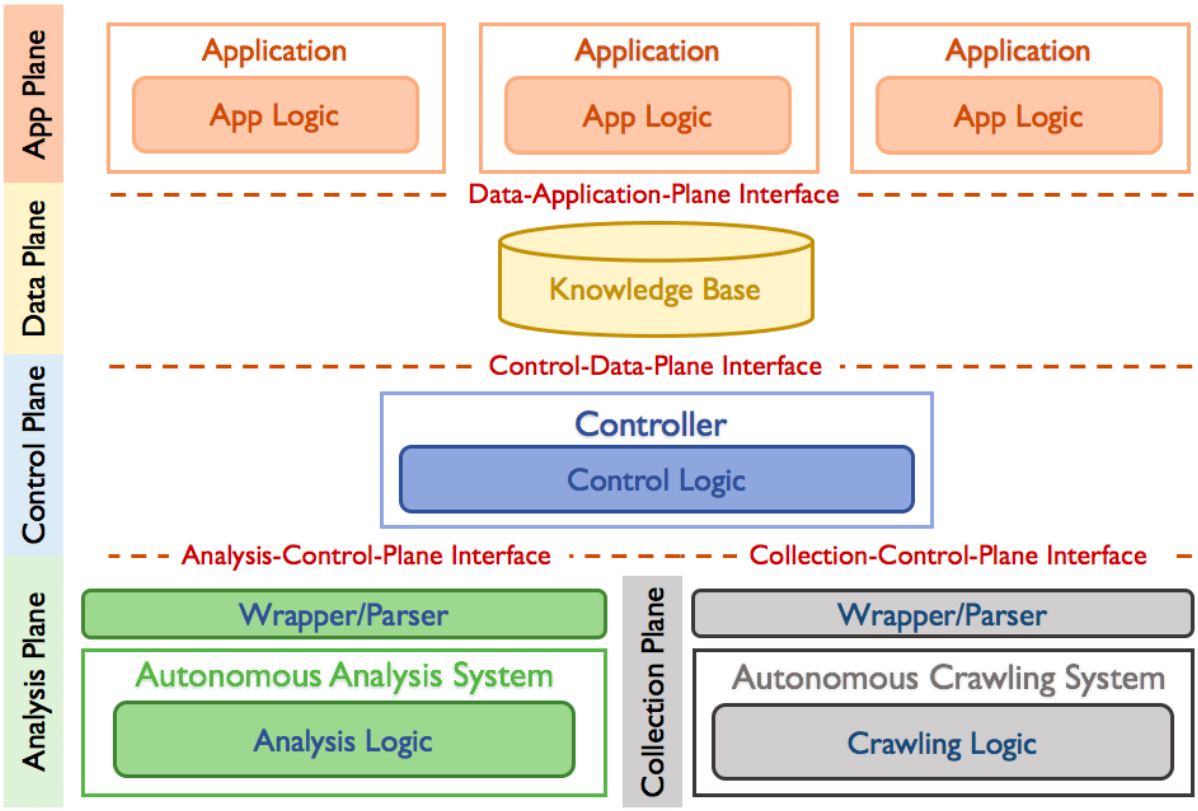

Figure 2. ATIS Architecture Overview

can store them in its own database. Analysis plane includes all sorts of analysis modules and their corresponding parsers. For example, Cuckoo, which is an automated malware analysis system, resides in the analysis plane. ATIS abstracts each analysis module as a set of $\langle input, output, relationship \rangle$ 3-tuple. A logically centralized controller that resides in the control plane is the brain of ATIS. The controller is responsible for automatic intelligence discovery by orchestrating all the analysis modules and storing discovered intelligence to the data plane. Data plane stores a global knowledge base of the collected and generated intelligence information. Knowledge in ATIS is represented as a graph. The application plane includes ATIS applications that utilize the generated knowledge base to perform holistic threat analytics.

Besides the 5 planes, ATIS also defines the communication interface between each adjacent pair of planes to enable a reusable, scalable and flexible design.

C. Collection Plane

Collection plane consists of autonomous data crawlers, which only care about crawling relevant threat feeds following the controller's commands. A typical cycle for a data collector can be broken into 2 phases: i) collector determines which data feeds provide relevant and valuable data and collect the raw contextual data through autonomous crawlers; and ii) collector stores the collected data in structured data format.

D. Analysis Plane

Each analysis module is autonomous and it only cares about analyzing certain types of data. ATIS abstracts each analysis module as a set of $\langle i, o, r \rangle$ 3-tuple, where i stands for the type of input, o is the type of output, and r is the type of the relationship between input and output. In ATIS, types are not predefined. They are simply case-sensitive strings that are provided by analysis module developers. This design can make adding new types be a very simple task. We denote the set of all possible input types as I, output types as O, all known types in the system as the union of them $T = I \cup O$, and all relationships as R.

Take the Cuckoo sandbox, an automated malware analysis system, as an example. Cuckoo takes an file or a URL as input and performs static and dynamic analysis on it. ATIS abstracts Cuckoo as a set of $\langle i, o, r \rangle$ 3-tuples:

$$\{\langle \texttt{"windows-exe"}, \texttt{"ip"}, \texttt{"connect-to"} \rangle, \quad (1)$$
$$\langle \texttt{"windows-exe"}, \texttt{"string"}, \texttt{"has-string"} \rangle, \quad (2)$$
$$\langle \texttt{"windows-exe"}, \texttt{"md5"}, \texttt{"md5-is"} \rangle, \quad (3)$$
$$\langle \texttt{"pdf"}, \texttt{"md5"}, \texttt{"md5-is"} \rangle\} \quad (4)$$

Line (1) says Cuckoo can take a Windows executable file as an input, and output what IP addresses the executable file tries to connect. The relationship between the input and output is "connect-to". Line (2) says Cuckoo can also report what strings the executable file has. Additionally, the outputs of an analysis module can also be represented as a set of 3-tuple with the actual analysis values. Line (3) and (4) show that Cuckoo relates an analysis value, MD5, with the original input, which is an executable file or a pdf file. Obviously, a comprehensive analysis system, such as Cuckoo, will be abstracted as a set of hundreds or even thousands of 3-tuples like this.

Not only an analysis system's ability can be abstracted as a set of 3-tuples, it also reports to the controller in a similar format by replacing the input and output types with actual values as follows:

$$\{\langle \texttt{"18c323..."}, \texttt{"8.8.8.8"}, \texttt{"connect-to"} \rangle, \quad (1)$$
$$\langle \texttt{"18c323..."}, \texttt{"Hello!"}, \texttt{"has-string"} \rangle\} \quad (2)$$

In these two examples, Cuckoo reports the analysis results that executable whose MD5 is `"18c323..."` tries to `"connect-to"` the IP address `"8.8.8.8"`, and it also has a string `"Hello!"`.

E. Analysis-Control-Plane Interface

Since most existing analysis systems do not report their results in this way, an analysis module-specific shim is needed to translate the results to the required format and forward to the controller. In addition, the shim is also responsible for wrapping the existing analysis systems so that they provide a consistent interface for the controller to call. The interface consists of the following high level operations:

i) *CreateTask* is used to create a new task. When calling this function, the controller should send the to-be analyzed data along. The analysis module returns a uniquely identifiable attribute, MD5, SHA1 for file, email for a thread of a forum, etc., using which the controller can retrieve the results, or an error message if it cannot perform the task.

ii) *ListTask* is used to retrieve the list of tasks. The analysis module returns objects of tasks as list, or an error message if it cannot perform the task. Considering the number of objects in the list can be huge, this query provides only uniquely identifiable attribute of the object which can be utilized to obtain detailed object.

iii) *RetrieveTask* is used to retrieve the result of a task. The analysis module returns detailed result of a task by filtering result with matching parameter, or an error message if it cannot perform the task.

iv) *ExistsTask* is used to perform validation on the existence of a task. The analysis module returns a boolean value based on filtering result of matching arguments, or a error message if it cannot perform the task. It returns yes, if record exists and no if it gets an empty set.

v) *UpdateTask* is used to update an existing task. The analysis module returns a uniquely identifiable attribute, MD5, SHA1 for file, email for a thread of a forum, etc., using which the controller can retrieve the results, or an error message if it cannot perform the task or no record exists.

vi) *DeleteTask* is used to delete an existing task. The analysis module returns a uniquely identifiable attribute, MD5, SHA1 for file, email for a thread of a forum, etc., with the proper message to understand the task was deleted in a successful manner, or an error message if it cannot perform the task or no record exists.

vii) *Status* is used to check status of an analysis module. The analysis module returns status of the module, i.e. status = active, disabled, or an error message if it cannot perform the query.

F. Control Plane

A logically centralized controller resides in the control plane. To orchestrate all the analysis systems for automated intelligence discovery, the controller needs to know 1) what analysis systems are available? and 2) what are their abilities? To this end, it maintains all the analysis system names, such as Cuckoo or WHOIS, and their addresses. It also maintains the abilities of each analysis system, which is represented as the set of all analysis system's input-output-relation 3-tuples.

To best utilize all the analysis system, the controller also maintains a tree of all input types. Figure 3 shows a part of the tree that is used in our implementation. Each child in the tree is a subtype of its parent. For example, `"windows-exe"` is a subtype of `"executable"`.

Whenever there is a new piece of data, the controller will send it to the analysis systems that can take this data type as input by consulting the type tree. When an output is generated, the controller stores it into the data plane. If the output data can be further analyzed by other analysis systems, the controller will automatically send the data to them as well. It is this cascading effect that enables automatic intelligence discovery.

In addition, whenever a new analysis system is plugged into the system. The controller will check if there is any data in the data plane that can be analyzed by this system.

G. Data Plane

The data plane stores a global knowledge base of the collected and discovered intelligence. Given the heterogeneousness of the intelligence type and the systems that generate them, the data plane needs to store and represent knowledge in a suitable manner so that a holistic picture of heterogeneous intelligence can be painted.

At a high level, the knowledge base can be viewed as a graph, where each vertex is a data point that includes some information about it, such as the type $t \in T$ of the data

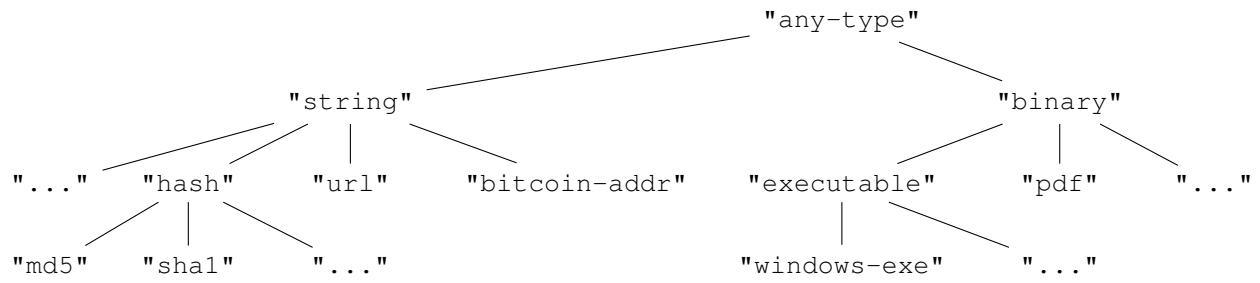

Figure 3. A Part of the Tree of Input Types I in our Implementation

point and the actual value of it. Each edge in this graph is labeled with a relationship $r \in R$ and the name and version of the analysis system that discovered the relationship, such as `"Cuckoo 2.0-RC1"`. By abstracting the analysis plane outputs, the controller can easily obtain all the information needed to form the graphical knowledge base at the data plane.

H. Application Plane

Intelligence analysts can develop applications to perform analytics on the knowledge base. Each application defines its own application logic and communicates with the data plane via the data-application-plane interface.

For instance, a 'Searcher' application can take analysts' input and search related information in the knowledge base and visualize the output. Such a 'Searcher' can search based on the attributes of both vertex and edge.

The connected intelligence enables applications on new topics and new methods on old topics as well. For example, a 'Malware Clustering' application may not only take the attributes of a malware into account but also considers its relationships with other data points.

IV. Implementation of ATIS

We implemented a prototype ATIS framework that integrates various open source and in-house analysis platforms including the Cuckoo malware analysis platform, an in-house Bitcoin blockchain parser, an in-house social analysis system [21], etc.

The autonomous crawling and analysis systems were implemented in different programming languages. For example, social webscrapper and analysis was built in JAVA, HTML parser, Neo4j graph database, Apache Lucene library, Jackson2 library. However, malware crawler and analysis system was implemented Python, Flask, MySQL, MongoDB etc. All modules are exposed as REST web services to receive commands from the controller and send data to the controller. The multithreaded controller was developed using Python, Flask, RabbitMQ [4].

All discovered intelligence is stored in a Neo4j graph database instead of relational database systems like MySQL, since it has been found that graph databases work well on highly connected data. A web-based management interface is developed to show the potential of the system through various applications by using Flask framework, Python, Cytoscape.js which queries to Data-Application-Plane interface to provide required filtered data.

We also deployed ATIS on our Openstack. Each ATIS module, including each analysis system, controller, knowledge graph database and each application runs on a dedicated Openstack instance with 8GB RAM.

V. Example Applications

In this section we present two applications we developed on top of ATIS to demonstrate the effectiveness of ATIS. Taking advantage of the features offered by ATIS the first application, a searcher, considers all the nodes and relationships in the knowledge graph and returns search results on the graph based on human analysts' inputs, whereas the second application, namely SocialSEAL, only considers the relationships discovered by our in-house social analysis module and generates results of social dynamics. At the time of writing, the controller has orchestrated the analysis plane to discover new intelligence and inserted more than 89,000 nodes from heterogeneous data sources and more than 230,000 relationships between nodes into the knowledge graph of ATIS.

A. A Searcher Application

The searcher application provides a platform for human analysts to carry out research queries, visualize relationships between heterogeneous data nodes and understand how they are correlated to get the insight of data, which is generally not eminent by just looking at the data. Human analysts can filter the data based on various different node types, node attributions and node relationship hops.

For example, if human analysts want to find out how the email address of `cralu@gecad.ro` is connected with varoius cybersecurity events, they can simply input the email address in the search bar. In Figure 4, a blue node represents an email address while a green node represents a file with its MD5 value. As

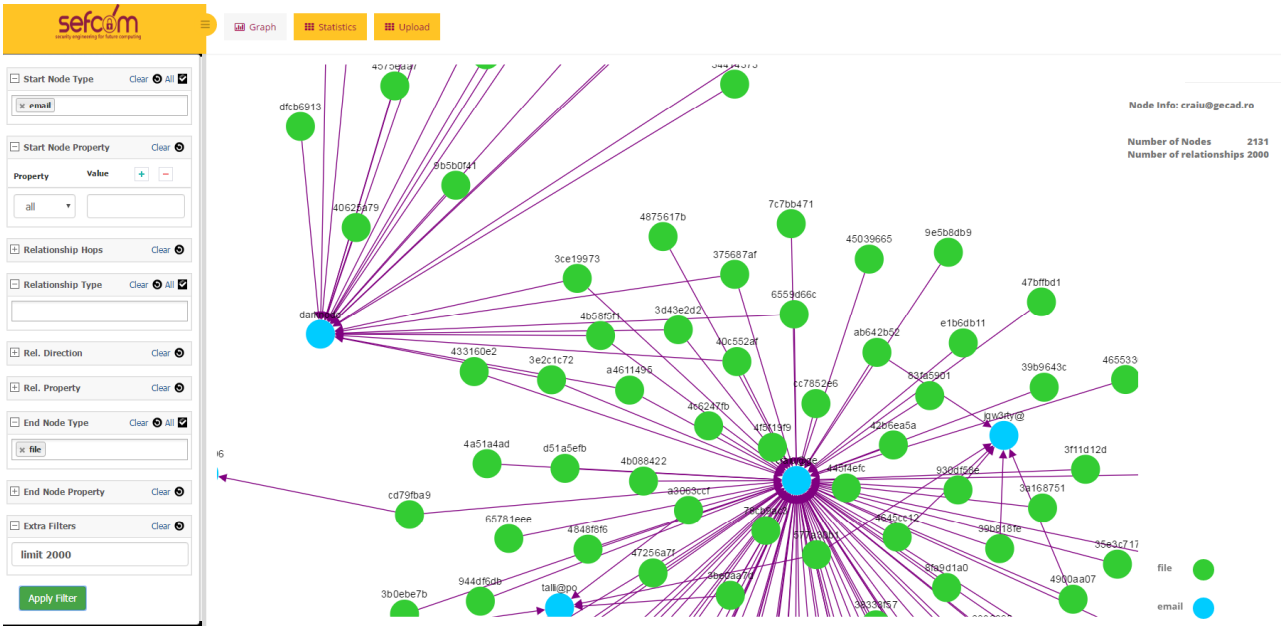

Figure 4. A Searcher Application: Interconnection between cralu@gecad.ro and and darknode@oninet.es through malwares.

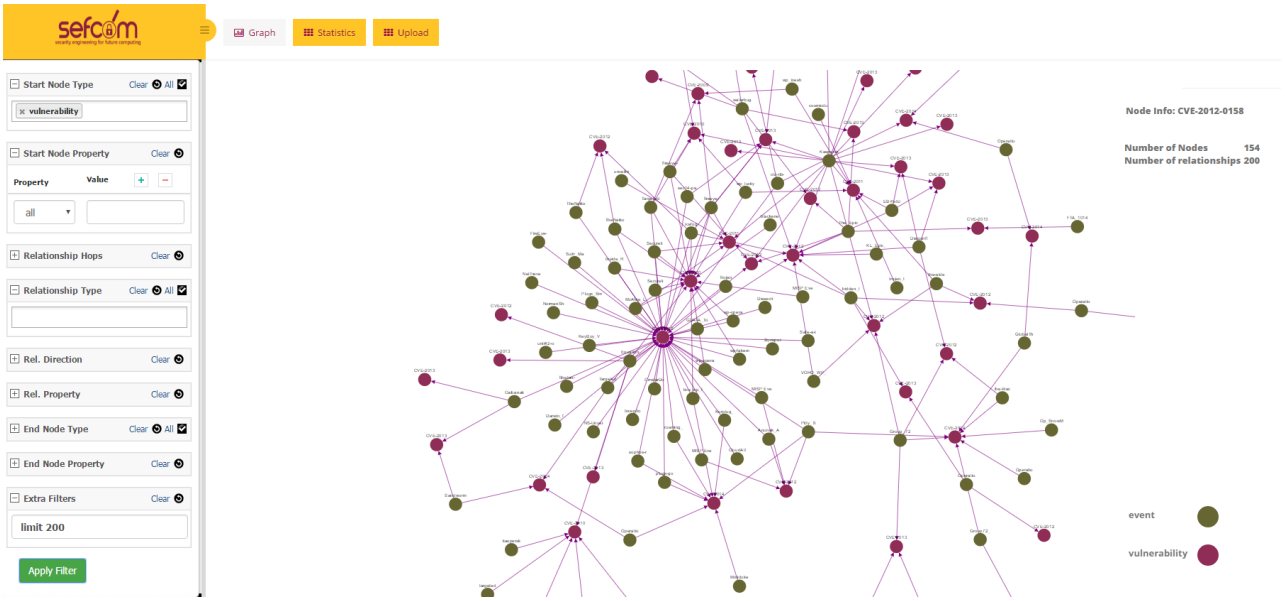

Figure 5. A Searcher Application: Events sharing Common Vulnerabilities and Exposures.

shown in Figure 4, ATIS has automatically identified that many malwares has a connection to `cralu@gecad.ro` and `darknode@oninet.es`. The searcher visualizes that the email address of `cralu@gecad.ro` has connections with more than 50 malicious files, while the email address of `darknode@oninet.es` has connections with more than 15 malicious files in our knowledge graph. In addition, a human analyst can easily identify from Figure 4 that several malwares (e.g. `375687afa577c769de9b89f1e1449dc4` and `4b58f5f1622517c381ebc9544a380273`) have connections to both email addresses, suggestting that the two emails are connected.

In another example, if human analysts want to find out which cyber attack events have exploited the vulnerability identified by `CVE-2012-0158`, they can simply input the CVE number in the search bar. Figure 5, in which a green node represents an event and a red node represents a vulnerability, shows that attack events 'Dissecting

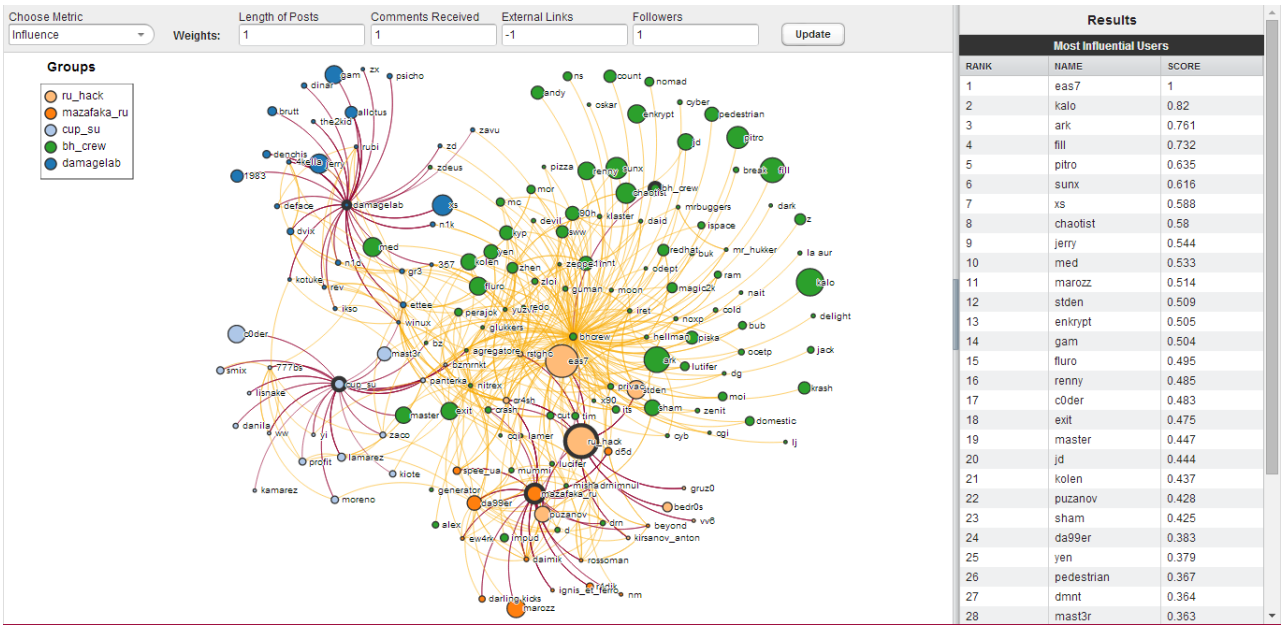

Figure 6. SocialSEAL application: Visualization of underground social network where nodes are ranked by influence.

the Kraken' [18] and 'NormanShark-MaudiOperation' [9] exploited `CVE-2012-0158`. And, 'Securelist RedOctober' [12] has a connection with `CVE-2012-0158`, `CVE-2010-3333` and `CVE-2009-3129`. Similarly, 'WP Operation Tropic Trooper' [1] event has a connection with `CVE-2012-0158` and `CVE-2010-3333` suggesting that threat actors are likely to exploit multiple vulnerabilities at the same time to increase their success rate.

B. SocialSEAL Application

SocialSEAL is an application developed on top of ATIS that only cares about the relationships discovered by our in-house social analysis system. SocialSEAL is a great example to show that even though the ATIS knowledge graph maintains a global and comprehensive view of all intelligence discovered by all sorts of autonomous systems from the analysis plane, the application developers can define their own business logic to further analyze the intelligence.

SocialSEAL defines a suite of metrics to rank users and groups based on user activeness, user influence, group activeness, and group influence. Figure 6 shows the nodes in the knowledge graph ranked by their Influence. The top filters provide the analysts with the controls to change the metrics or weights of the parameters used in the influence and activeness computation. The variation in the User/Group node size gives an intuitive idea of the qualified influence value of a user or group. The left-hand side legend suggests the color of different groups presented in the graph. The nodes are also color-coded to indicate which group they belong to. The yellow links represent User-User relationships and maroon links represent User-Group relationships. On the sidebar, both the search results and the users/groups in the knowledge graph ranked by their influence scores are shown.

On clicking any user or group node in the graph, SocialSEAL takes the analyst to a page which shows just the social circle of a threat actor and information from the profile such as photo, location, interests etc. Figure 7 shows the profile of the threat actor in the dataset which displays security-related terms such as zombie, crack, hack, rootkit, spam, exploit, attack etc. being used frequently by this actor. It also provides the count of following and followed users to understand the influence of the actor in the community.

VI. Related Work

The community interest in threat intelligence analysis and sharing platform has continuously growing throughout the years. Magee et al. presented a collective of threat intelligence gathering system [13]. Their system can report threats to network administrators in a plurality of threat feeds, including for example malicious domains, malicious IP addresses, malicious e-mail addresses, malicious URLs and malicious software files.

Beaver et al. proposed a generic threat assessment approach that provides a computational means for merging multi-modal data for the purpose of assessing the presence of a threat and negated the need for relying heavily on human analysis to both combine any available data and draw conclusions about the probability of a threat [3].

MITRE also presented a similar system called Collaborative Research Into Threats, which combines an analytic engine with a cyber threat database that not only serves as

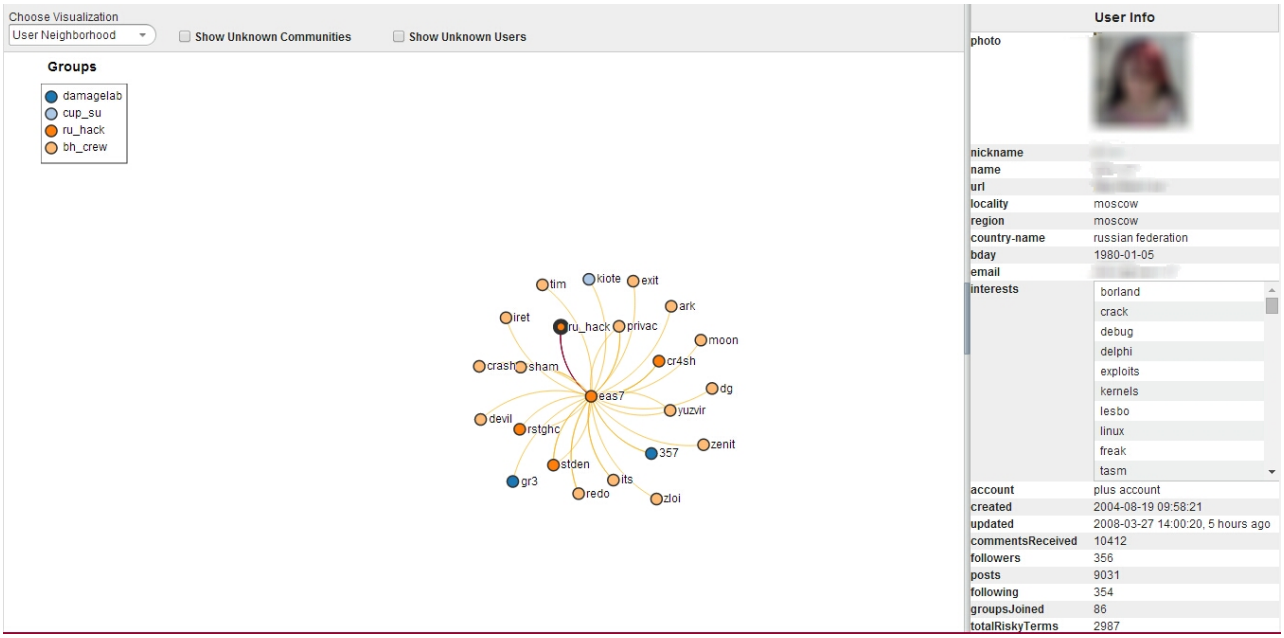

Figure 7. SocialSEAL application: Profile detail of a user

a repository for attack data and malware, but also provides analysts with a powerful platform for conducting malware analyses, correlating malware, and for targeting data. The hierarchical structure provided by the system gives analysts the power to 'pivot' on metadata to discover previously unknown related content [5].

VII. Conclusions

In this paper, we presented the design and implementation of an automated threat intelligence fusion framework ATIS that is able to take heterogeneous data into consideration and discover new intelligence from apparently isolated cyberattack events. To this end, ATIS consists of 5 planes, namely analysis, collection, controller, data and application planes. We discussed the design choices we made in the function of each plane and the interfaces between two adjacent planes. In addition, we developed two applications on top of ATIS to demonstrate its effectiveness.

Acknowledgment

This work was supported in part by grants from the AllState Corporation, the U.S. Army Research Laboratory and the Center for Cybersecurity and Digital Forensics at Arizona State University. The information reported here does not reflect the position or the policy of the funding agency or project sponsor.

References

[1] K. Alintanahin. Operation tropic trooper: Relying on tried-and-tested flaws to infiltrate secret keepers. http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-operation-tropic-trooper.pdf, 2015.

[2] S. Barnum. Standardizing cyber threat intelligence information with the Structured Threat Information eXpression (STIX). The MITRE Corporation, 2012.

[3] J. M. Beaver, R. A. Kerekes, and J. N. Treadwell. An Information Fusion Framework for Threat Assessment. In *Proc. 12th International Conference on Information Fusion*, pages 1903–1910, 2009.

[4] S. Boschi and G. Santomaggio. *RabbitMQ Cookbook*. Packt Publishing Ltd, 2013.

[5] C. Community. Collaborative Research Into Threats. https://github.com/crits/crits, 2016.

[6] Y. Community. YARAs documentation. http://yara.readthedocs.io/en/v3.4.0/, 2016.

[7] J. Connolly, M. Davidson, and C. Schmidt. The Trusted Automated eXchange of Indicator Information (TAXII). The MITRE Corporation, 2012.

[8] M. Egele, T. Scholte, E. Kirda, and C. Kruegel. A survey on automated dynamic malware-analysis techniques and tools. *ACM Computing Surveys*, 44(2):6, 2012.

[9] S. Fagerland. The chinese malware complexes: The maudi surveillance operation. Norman Shark Technology, 2013.

[10] C. Guarnieri, A. Tanasi, J. Bremer, and M. Schloesser. The cuckoo sandbox. In *Black Hat USA*, 2013.

[11] T. J. Holt. Social networks in the computer underground. In *Congresso de Seguridad en Computo*, 2008.

[12] Kaspersky. The "red october" campaign - an advanced cyber espionage network targeting diplomatic and government agencies. https://securelist.com/blog/incidents/57647/the-red-october-campaign/, 2013.

[13] J. Magee, A. Andrews, M. Nicholson, J. James, H. Li, C. Stevenson, and J. Lathrop. Collective threat intelligence gathering system, Jan. 2 2014. US Patent App. 13/538,831.

[14] McAfee. Net losses: Estimating the global cost of cybercrime. economic impact of cybercrime. http://www.mcafee.com/us/resources/reports/rp-economic-impact-cybercrime2.pdf, 2014.

[15] S. Morgan. Cyber crime costs projected to reach $2 trillion by 2019. http://www.forbes.com/sites/stevemorgan/2016/01/17/cyber-crime-costs-projected-to-reach-2-trillion-by-2019/, 2016.

[16] L. Obrst, P. Chase, and R. Markeloff. Developing an ontology of the cyber security domain. In *Semantic Technologies for Intelligence, Defense, and Security*, pages 49–56, 2012.

[17] T. O'Ham. Chrome add-on steals bitcoin with social engineering, qr codes vulnerable. http://bitcoinist.net/chrome-add-on-steals-bitcoin-using-social-engineering-all-qr-codes-vulnerable/, 2016.

[18] P. Rascagnres. Dissecting the "kraken": Analysis of the kraken malware that was used for a targeted attack in uae. https://blog.gdatasoftware.com/2015/05/24280-dissecting-the-kraken, 2015.

[19] C. W. Robert P. Hartwig. Cyber risks: The growing threat. Insurance Information Institute, 2014.

[20] M. Spagnuolo, F. Maggi, and S. Zanero. Bitiodine: Extracting intelligence from the bitcoin network. In *International Conference on Financial Cryptography and Data Security*, pages 457–468. Springer, 2014.

[21] Z. Zhao, G.-J. Ahn, H. Hu, and D. Mahi. Socialimpact: systematic analysis of underground social dynamics. In *European Symposium on Research in Computer Security*. Springer, 2012.