Malware investigation using semantic technologies

Rodrigo Carvalho[1,2], Michael Goldsmith[1], Sadie Creese[1]

[1] Cybersecurity Group, University of Oxford, Oxford, UK,
rodrigo.carvalho@cs.ox.ac.uk,
[2] Brazilian Federal Police, Brasília, Brazil

Abstract. Malware investigation is a major issue in fighting cybercrime. Because most of the research in this area comes from commercial companies, there is a bigger emphasis on detection rather than attribution (i.e., finding the criminal). Regarding this challenge, we believe that semantic technologies could help the human analyst to reveal relationships among the rich dataset of artefacts within the malware ecosystem. Aiming at a better balance between human reasoning skills and computer processing capabilities, we reproduce the rationale described in a relevant malware investigation report from 2015 [8]. Differently from most published reports, its transparent methodology and analysis of competing hypotheses regarding a sample of 52 distinct exploit files made it possible to create the ItCo ontology as a case study. By leveraging domain-specific entities and incremental exporting of axioms, it was possible to reach the same conclusions as the authors.Further on, by adding more samples from another data source to the knowledge base, we were able to find new relationships in a semi-automated, scalable fashion.

Keywords: malware, investigation, incremental reasoning, entity linkage

1 Introduction

Cyber-attacks are so frequent nowadays that few people still get surprised to know that an important company has been hacked. In a situation in which even hospitals are being targeted [5], the common saying "It's not a matter of *if* a cyber-attack will target you, but *when*" is becoming more popular outside the cybersecurity domain.

Back in 2013, an United Nations study on cybercrime stated that "At the global level, law enforcement respondents ... perceive increasing levels of cybercrime, as both individuals and organized criminal groups exploit new criminal opportunities, driven by profit and personal gain ...with cybercrime black markets established on a cycle of malware creation, computer infection, botnet[3] management... Cybercrime perpetrators no longer require complex skills or techniques".[17].

[3] A botnet is an interconnected network of computers infected with malware without the user's knowledge and controlled by cybercriminals. They're typically used

This last sentence, which is one of the topics of this paper, is endorsed by the fact that "Places in the Internet where cybercriminals converge to sell and buy different products and services exist. Instead of creating their own attack tools from scratch, they can instead purchase what they need from peers who offer competitive prices. Like any other market, the laws of supply and demand dictate prices and feature offerings. " [12]

In addition to malicious software, other resources like stolen data and services (e.g. botnet rental) are regularly traded in such "black markets". In addition to attracting more people to cybercrime, the wide availability of a broad range of easy-to-use malicious resources has definitely changed the malware ecosystem, in a way that "...today there is no disorganized digital crime. Because of the way criminals have organized, the threat landscape is ever evolving and more importantly, ever growing.' [7]. Such a trend was confirmed in a report by Anti-Phishing Working Group(APWG) [3]:

- Computers around the world continue to be infected with malware at a high rate. The global infection rate was 36.51% in Q1, 32.21% in Q2, and 32.12% in Q3 of 2015;
- The number of unique phishing[4] reports submitted to APWG from Q1 through Q3 was 1,033,698. APWG is now receiving twice as many reports as it did in 2014.

At the same time such alarming statistics might indicate that, in fact, cybercriminals are more organized nowadays, they also bring to attention that there is at the same time more related data available. However, malware investigation is still a recent development, with a large part of the security community focusing on detection rather than actually identifying the criminal. In this paper, we discuss the potential of semantic technologies to tackle the latter issue. In the following sections we will explain the reasons for the case study chosen and discuss how our semantic investigation approach could aid the malware investigator's "insight-generation" process when analysing a malware dataset.

1.1 ItCo report

A significant cyber-security event[16] happened in 2015, in which data from the company Hacking Team[5] was leaked to the internet. Among the exposed files,

to send spam emails, transmit viruses and engage in other acts of cybercrime. *Source:* https://usa.kaspersky.com/internet-security-center/threats/botnet-attacks, *accessed in* 27/04/2016.

[4] Phishing is the attempt to acquire sensitive information such as usernames, passwords, and credit card details (and sometimes, indirectly, money), often for malicious reasons, by masquerading as a trustworthy entity in an electronic communication. *Source:* https://en.wikipedia.org/wiki/Phishing, *accessed in* 26/04/2016.

[5] Hacking Team is a Milan-based information technology company that sells offensive intrusion and surveillance capabilities to governments, law enforcement agencies and corporations. *Source:* http://en.wikipedia.org/wiki/Hacking_Team, *accessed in* 24/04/2016.

there were some 0-day vulnerabilities[6] which were promptly leveraged by hackers. Following this breach, researchers at Shadow Server Foundation[7] published a paper entitled *The Italian Connection: An analysis of exploit supply chains and digital quartermasters*, whose main objective is to " ... to reveal relationships between groups that appear to be independent based on analysis of standard technical artifacts such as dropped payload[8] and command and control infrastructure." [8].

Differently from most reports describing malware campaigns, this report clearly states the methodology used (which and why each investigation step was taken), explains the rationale behind each assertion they make (e.g. "We define independent operators as actors that maintain distinct infrastructure without any technical overlaps such as ip history.") and conclude by presenting competing hypotheses regarding whether different exploits might have a unique origin. Its distinctiveness was also asserted by at least one more cyber-security analyst[6].

Therefore, we consider this report (which will be simply referred as *ItCo report* for the rest of the paper) an excellent opportunity to reproduce a cybercrime investigation using semantic technologies. Although this idea has been discussed before in a position paper by the authors[9], it was difficult to implement due to the lack of specific domain knowledge (i.e., the rationale used by malware investigators) and difficulty to obtain a suitable dataset for the case study. For the rest of the paper, the Shadow Server's report will be simply referred as *ItCo report*.

1.2 Related Work

Applying semantic technologies to fight cybercrime is starting to gain attention from some authors and organisations. The European Union is currently funding the European Informatics Data Exchange Framework for Courts and Evidence project (Evidence project) [10], which aims to develop best practices and procedures for digital evidence gathering and sharing across its members. They are engineering ontologies to describe, for instance, the concept *InternetLog* metadata (which is a subclass of *AutomaticGeneratedEvidence*), and the relationship *ElectronicEvidence isContainedIn DigitalSource*.

[6] A zero day vulnerability refers to a hole in software that is unknown to the vendor. This security hole is then exploited by hackers before the vendor becomes aware and hurries to fix it. *Source:* http://www.pctools.com/security-news/zero-day-vulnerability/, acessed in 24/04/2016

[7] Established in 2004, The Shadowserver Foundation gathers intelligence on the darker side of the internet. We are comprised of volunteer security professionals from around the world. Our mission is to understand and help put a stop to high stakes cybercrime in the information age. *Source:* https://www.shadowserver.org/wiki/, *accessed in* 27/04/2016.

[8] In computer security, payload refers to the part of malware which performs a malicious action. Source https://en.wikipedia.org/wiki/Payload_(computing), *accessed in* 25/04/2016.

More specific to malware analysis, researchers from the Computer Emergency Response Team (CERT) at Carnegie Mellon University published a paper[14] discussing an ontology dedicated to malware analysis based on established vocabularies and taxonomies [13]. Their goal is to provide a more scientific approach to malware research, and they hope other experts will adopt such ontology, thus starting to speak the same language. Although sharing some goals, our approach focuses more on the relationship between distinct samples of malware and other IOCs[9] rather than the analysis of the malware itself.

Finally, there are already some initiatives [2, 1, 11] aiming to provide a standard structure for describing IOCs. The last two also describe relationships between entities, which approximate them to ontologies. However, their wide scope and slow adoption by the cybersecurity community dissuaded us from adopting them in our prototype, at least for now.

2 Objective

Aiming at a better balance between human reasoning skills and computer processing capabilities, the main objective of this paper is to demonstrate that semantic technologies can help the cybercrime investigator when analysing digital evidence. In addition to enabling the integration of distinct data sources (which is a native feature of semantic technologies), we believe that our approach could facilitate the insight-generation process of the analyst by allowing him to test different investigation hypotheses by assessing the resulting states of the dataset.

Therefore, we will demonstrate that the same conclusions of the authors from the *ItCo report* could be achieved using our approach, which could be described as an iterative process of *analysing the dataset* → *creating and applying new investigation rules* → *analysing the new state of the dataset*. The main features of the prototype implementing this approach are:

- Easy roll-back to a previous state of the dataset (for instance, in the case that the current transformation is deemed irrelevant);
- On-the-fly definition of novel classes and relationships resulting from the intelligence gathered in the current transformation, which would facilitate organizing the entities for the next transformations;
- The ability to reproduce the steps towards a specific hypothesis, allowing other analysts to assess it.

3 Investigation process

The main idea of the prototype is to allow the analyst to apply a bespoke set of rules to an existing knowledge base to check whether his assumptions are

[9] Indicator of compromise (IOC) in computer forensics is an artifact observed on a network or in an operating system that with high confidence indicates a computer intrusion. Typical IOCs are virus signatures and IP addresses, MD5 hashes of malware files or URLs or domain names of botnet command and control servers. *Source:* https://en.wikipedia.org/wiki/Indicator_of_compromise, *accessed in* 27/04/2016.

correct. Once confirmed, the intelligence gained may inspire him to define new rules to be applied to the new state of the knowledge base and so on, until he reaches a final conclusion about the hypothesis being tested.

As a preparation for this investigative process, there are two tasks to perform:

- *Define the initial state of the ontology*, reflecting the data available and basic relationships between them. It is not necessary to specify all the entities at this stage, as the ontology will be refined during the process;
- *Obtain relevant, structured datasets and convert them to a semantic format.* It is important to verify if the converted data still reflects the source data (e.g., check if all object properties are actually connecting instances, not only entities).

To perform both of this tasks, we rely on the well-known and intuitive tools Protege[10] and Karma[11], respectively. For the reasoning task we will make use of RDFox[12] [15], which is called by a script implementing three general functions:

1. Format the input data in a way expected by the reasoner;
2. Invoke the reasoner and export the inferred axioms to a new file;
3. Create a new folder to store the current transformation.

In addition to automating the processes described above, the python script also allows us to effortlessly delete the newly created folder and files. This could be performed in the case an error occurs or if the analyst is not happy with the outcomes of the inferences. Minimizing the burden of such trivial but time-consuming tasks is important to avoid losing the focus on the mental reasoning process the analyst is conducting. The script expects three mandatory arguments:

- *Action*, whether to infer new axioms or the delete the last iteration;
- *Version*, or state (folder) to which the action will be performed.;
- *BaseDir*, which will store the different versions folders.

The script consumes a turtle [4] file. *Source:* file containing the ontology and the instances, and also multiple text files containing the rules to be applied. The output is a new turtle file containing all the data from the previous state and the inferred axioms. For the basic "versioning control", the folders adopt the naming convention"dd" (e.g. *82*), in which the least significant digit refers to the iterations from the specific hypothesis identified by the most significant digit.

[10] A free, open-source ontology editor and framework for building intelligent systems. *Source:* http://protege.stanford.edu/, *accessed in* 27/04/2016.

[11] Karma is an information integration tool that enables users to quickly and easily integrate data from a variety of data sources. *Source:* http://usc-isi-i2.github.io/karma/#about, *accessed in* 27/04/2016.

[12] RDFox is a highly scalable in-memory RDF triple store that supports shared memory parallel datalog reasoning. *Source:* https://www.cs.ox.ac.uk/isg/tools/RDFox/, *accessed in* 27/04/2016

4 Case Study

In order to demonstrate the viability of our approach, we will reproduce the investigation described in section 1.1 using our prototype. The case study was reproduced in a MacBook Pro with *16 GB* RAM, *2.2 GHz Intel Core i7* processor and *Journaled HFS+* file system formatted in a *ssd* storage. The versions of the software used are: OS Yosemite 10.10.5, Protege 5.0.0, Karma 2.045, RDFox 2510 and Python 3.4.0.

4.1 Preparation

This case study starts with a careful reading of the report to determine which concepts will be included in the ontology, and their types (whether classes, object properties or data properties). Although it is not the aim of this paper to discuss why each of them were chosen, some brief explanation of the most important of them will be given in due course.

The dataset used by the authors comprises a total of 52 exploit samples obtained by crawling "...specific websites that have been previously used to deliver exploits and malware" [8], and also searching online databases such as VirusTotal[13] and ShadowServer using Yara[14] rules as the "search term". Such rules were designed to identify exploit files targeting either CVE[15]-2015-5119 or CVE-2015-5122, which are the vulnerabilities related to Hacking Team's 0-day exploits leaks.

For each sample, the authors of the *ItCo report* collected the features described in Tables 1 and 2, which can be extracted by using static and dynamic sandbox[16] analysis. The authors published such features, also known as Indicators of Compromise (IOCs), as an Excel spreadsheet containing 6 worksheets. Some of the column headers of the main worksheet, "SWF IOCs", are described in Table 1 together with the assigned semantic entity and a sample value.

The other 5 worksheets are pretty similar among themselves. They contain the md5 hash digests of different *ActionScript*[17] classes embedded in the `ExploitFiles` analysed, as depicted in Table 2.

[13] VirusTotal is a free service that analyzes suspicious files and URLs and facilitates the quick detection of viruses, worms, trojans, and all kinds of malware. *Source:* https://www.virustotal.com/, *accessed in* 27/04/2016.

[14] YARA is a tool aimed at (but not limited to) helping malware researchers to identify and classify malware samples. *Source:* https://plusvic.github.io/yara/, *accessed in* 27/04/2016.

[15] CVE is a dictionary of publicly known information security vulnerabilities and exposures. *Source:* https://cve.mitre.org/, *accessed in* 27/04/206.

[16] A sandbox is a security mechanism for separating running programs. It is often used to execute untested code, or untrusted programs. *Source:* https://en.wikipedia.org/wiki/Sandbox_(computer_security), *accessed in* 27/04/2016.

[17] ActionScript 3 is the programming language for the Adobe Flash Player and Adobe AIR runtime environments. *Source:* http://www.adobe.com/devnet/actionscript/learning.html, *accessed in* 27/04/2016.

Table 1. Some IOCs from the report and the defined semantic entities.

Column header	**Semantic Entity**	**Sample value**
last-modified of swf/html	lastModifiedDate	7/14/15
exploit site	WebServer	cosmetech.co.jp/css/movie.swf
swf md5	File	e33cf5b9f3991a8ee4e71f4380dd7eb1
swf cve	cve	CVE-2015-5122
swf compression	compression	lzma
payload in swf	embeddedPayload	yes
payload url	WebServer	103.230.109.222/flashupdate.exe
payload md5	File	5a22e5aee4da2fe363b77f1351265a00
payload family	family	PlugX
C2	WebServer	amxil.opmuert.org

Table 2. Extract of the worksheet HT_Exploit ActionScript

Column header	Semantic Entity	Sample value
swf md5	File	75dc1e22e16c39e3532673f75fd41b93
HT_exploit.as	InternalClass	55bc2ac6bfcaaf9364a67cbd837aa66e
MyClass.as	InternalClass	3652a267b318b13c99c1a817416406ee
MyClass1.as	InternalClass	4b705980ed1b07becd76f47e007b5b3a

The large quantity of *ActionScript* classes contained in some `ExploitFiles` (often exceeding 30 instances) makes it infeasible to compare every pair of them using a graph-approach. Preliminary tests were conducted with 39 unique files, embedding between 6 and 8 `InternalClasses` each. Comparing every two of them for 5 shared `InternalClasses` took the *RDFox* reasoner $\tilde{4}$s, and increasing by 1 the number of compared *InternalClasses* exponentially increased the reasoning time to $\tilde{1}0$ minutes. Even though other reasoners could have been tested, the naivety of the approach in this specific task suggested that external processing should be sought to increase efficiency.

As we intend to determine `ExploitFiles` similarity based on the number of identical (i.e. same hash value) *ActionScript* classes shared by two different `ExploitFiles`, the Jaccard Similarity Score[18] seems a good fit.

Therefore, the binary files from all the 52 malware samples mentioned in the *ItCo report* were downloaded, and the embedded *ActionScript* classes extracted using the software *FFdec Decompiler*[19], which was also used by the authors.. Then, a python script was implemented to produce the MD5[20] digest for each `ExploitFile` and internal *ActionScript* classes and calculate the Jaccard Sim-

[18] The Jaccard coefficient measures similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets. *Source:* https://en.wikipedia.org/wiki/Jaccard_index, *accessed in* 06/07/2016

[19] Opensource flash SWF decompiler and editor. *Source:* https://www.free-decompiler.com/flash/, *accessed in* 06/07/2016

[20] Although the MD5 algorithm is susceptible to collisions, it was chosen in order to keep compatibility with the dataset from the *ItCo report*

ilarity Score. Finally, the output from each tool was exported to a semantic format (*turtle*) using either *Karma* or *RDFLib*[21], and the resulting graphs were merged using the latter one.

The initial ontology, depicted in Figure 1, considered common malware domain knowledge, such as `Webserver` *hosts* `URL`. We will not consider some specializations like *Domain resolvesTo IP* within `WebServer` in this case for the sake of demonstration simplicity and also because we have not included DNS data yet. It is also worth noticing that, although one `File` can embed any other `File`, knowledge from the malware domain suggests that, normally, `ExploitFile` *embeds* `PayloadFile`, and not the opposite.

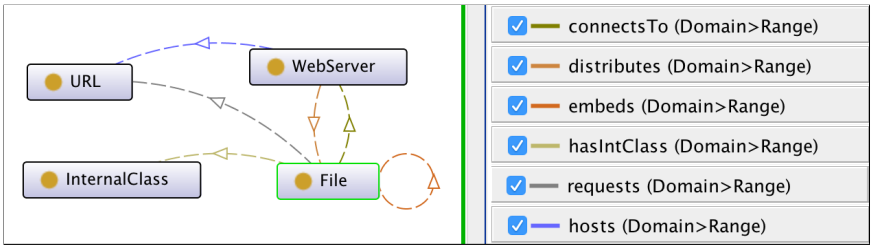

Fig. 1. Initial ontology.

4.2 Investigation

The authors of the *ItCo report* designed a process to cluster the collected samples, from which they could identify five distinct set of exploits. These clusters were named after the *ActionScript* class used to invoke the exploit: *HT_Exploit cluster, flash_exploit_002 cluster, exp1_fla cluster, exp2_fla cluster and movie_fla cluster*. One of their main arguments is that the exploits of two of these clusters were created by two different generator tools, which were shared with multiple operators. Such generator tools, or exploit kits, enable "...an operator to quickly and easy bind a payload or remote download url to shellcode in the flash exploit file via a handful of mouse clicks or a simple command." [8].

Different `ExploitFiles` created by the same generator tool necessarily share features among themselves. Therefore, the authors started by analysing which features could describe an unique generator. For instance, they argue that all `ExploitFiles` in the *HT_Exploit* cluster share the following characteristics:

- Created on the same date of 7/7/2015;
- Targeted the same vulnerability of CVE-2015-5119;
- Compressed via the LZMA algorithm;

[21] RDFLib is a pure Python package work working with RDF. *Source:* https://rdflib.readthedocs.io/en/stable/, *accessed in* 06/07/2016

– Contained an embedded payload;
– Had identical *ActionScript* classes.

It is not known how they come up with these features, or how they compared the values between different `ExploitFiles`. In any case, our approach could help assessing the results based on testing different combinations of features. We will use some of these features to start establishing semantic relationships between different `Files`.

Initial assessment of the dataset As a start point, our aim was to check if there were any `ExploitFiles` sharing a similarity score of 1 (i.e. all the `InternalClasses` are identical) with each other. For that, the *turtle* file containing the initial state of the knowledge base was loaded into a Fuseki[22], and the following *SPARQL* query was ran:

```
SELECT DISTINCT ?s ?ac ?co ?cve ?pl     WHERE {
 ?s vti:asCount ?ac .                   ?o vti:asCount ?ac .
  ?s vti:compression ?co .              ?o vti:compression ?co .
  ?s vti:cve ?cve .                     ?o vti:cve ?cve .
  ?s vti:embeddedPayload ?pl.           ?o vti:embeddedPayload ?pl .
  ?s vti:hasJSS-1.0-with ?o .           } ORDERBY(?ac)
```

26 triples were retrieved, among which it was easy to notice three clusters described in Table 3:

Table 3. Exploit files from the same generator.

Name	**EF**[a]	**JSS**[a]	**Compression**	**AS**[a]	**CVE**	**EB**[a]	**ItCo equivalent**
cluster-1-8	12	1	LZMA	8	2015-5122	yes	flash_exploit_002[b]
cluster-1-7	9	1	LZMA	7	2015-5119	yes	HT_Exploit[c]
cluster-1-50	5	1	ZLIB	50	2015-5119	no	MOVIE[d]

[a] EF: *number of files*, JSS: *Jaccard score*, AS: *number of ActionScript classes*, EB: *embedded Payload*
[b] contains one less file, not mentioned in the written ItCo but present in the digital IoCs. Al the other files match with Table 4 from *ItCo report*
[c] contains two more files. Due to an error the preparation phase, these files did not have their ActionScripts extracted, and thus were not identified as a member of this cluster. Al the other files match with Table 1 from *ItCo report*
[d] All files match with Tables 13 and 14.

Although *SPARQL* queries are currently being used to explore the data, we plan to implement, as a next step in the project, a GUI providing interaction for

[22] Fuseki is a SPARQL server. It provides REST-style SPARQL HTTP Update, SPARQL Query, and SPARQL Update using the SPARQL protocol over HTTP. *Source*: https://jena.apache.org/documentation/serving_data/, *accessed in* 06/07/2016

both the visual analysis of the results of each iteration and also the definition of new rules.

Iteration 1 In order to move on with the investigation, the prototype allows for the recently discovered knowledge to be materialized into the dataaset. In our present case, to group the identified files into specific clusters for further assessment. For this, a rule similar to the following one was created (one for each cluster) and fed to the reasoner. Although simple to define, the rule still have to be coded manually. We expect to significantly reduce this burden once the interactive GUI is implemented.

Rule 1.1. Class *cluster*

```
vtinv_ip:cluster-1-7(?ef1),          vtinv_ip:cluster-1-7(?ef2) :-
vtinv_ip:File(?ef1),                 vtinv_ip:File(?ef2),
vtinv_ip:compression(?ef1,?co),      vtinv_ip:compression(?ef2,?co),
vtinv_ip:asCount(?ef1,?as),          vtinv_ip:asCount(?ef2,?as),
FILTER(?co = "lzma"),                FILTER(?as = "7") ,
vtinv_ip:hasJSS-1.0-with(?ef1,?ef2) .
```

The 3 clusters stated in Table 3 are now part of our knowledge base. Their members were compressed using the same algorithm, exploit the same vulnerability, embed a payload file and share all their *ActionScript* classes. Finally, it was possible to map them to the clusters from the *ItCo report* by matching their MD5 digest values.

These facts validate, respectively, the following conclusions of the *ItCo report* authors: "*The common attributes strongly suggests these Flash exploits were created by a single exploit generator.*"(page 5 and Tables 1 and 2) and "*These common attributes indicate that these malicious Flash files were created by a single exploit generator*"(page 8, Tables 4 and 5).

Iteration 2 The next iteration starts by assessing the new state of the dataset. Having grouped the files with same properties and similarity score of 1 , our intuition would be to check if there are files also sharing the same properties, but with Jaccard score between 0.9 and 1. This would indicate, according to the authors of the *ItCo report*, files that are very similar but could not have been created by the same generator. Therefore, the following *SPARQL* query was run:

```
SELECT DISTINCT ?s ?ac ?co ?cve          WHERE {
 ?s vti:asCount ?ac .                    ?o vti:asCount ?ac .
 ?s vti:compression ?co .                ?o vti:compression ?co .
 ?s vti:cve ?cve .                       ?o vti:cve ?cve .
 ?s vti:hasJSS-0.9-with ?o .
  MINUS {?s rdf:type vti:cluster-1-50} .
  MINUS {?s rdf:type vti:cluster-1-8} .
  MINUS {?s rdf:type vti:cluster-1-7} .
} ORDERBY(?ac)
```

Eight results were returned this time, apparently forming two clusters, as show in Table 4. Note that we excluded `Files` from the previously identified clusters for the sake of demonstration simplicity. However, further exploration of the dataset with the following query:

```
SELECT DISTINCT ?s ?ot WHERE {  ?s rdf:type vti:cluster-1-50 .
  ?s vti:hasJSS-0.9-with ?o .   ?o rdf:type ?ot }
```

would have returned 5 `Files` that are part from both *cluster-1-50*(MOVIE) and *cluster-0.9-52*(EXP1), in accordance with page 13 of the *ItCo report*: " *Table 14 demonstrates the DNSCalc/APT12 operator used code from the exp1_fla cluster to develop the exploits in the movie_fla cluster.*" Finally, rule 1.1 was used to define the clusters described by Table 4 in the knowledge base.

Table 4. Exploit files sharing source code

Name	**EF**[a]	**JSS**[a]	**Compression**	**AS**[a]	**CVE**	**EB**[a]	**ItCo equivalent**
cluster-0.9-52	6	0.9	ZLIB	52	2015-5119	no	EXP1_fla [b]
cluster-0.9-57	2	0.9	ZLIB	57	2015-5122	no	EXP2[c]

[a] EF: *number of files*, JSS: *Jaccard score*, AS: *number of ActionScript classes*, EB: *embedded Payload*

[b] contains one less file. This was expected, since it has compression value *none*. Al the other files match with Tables 7 and 14 from *ItCo report*.

[c] Both files match with Table 10 from *ItCo report*.

That brings us to the same conclusion of the authors, stated in page 11: "*It is therefore unlikely that the files seen in the exp1_fla cluster were created via a shared exploit generator. A single generator would be unlikely to produce the differences seen in the underlying ActionScript. However, it is also doubtful that the above four underlying ActionScript classes seen in Table 9 would be identical unless the different operators were sharing code.*"

Iteration3 The last iteration from our case study aims to identify different actors (or operators) sharing source-code or generator tools. According to page 2 of the *ItCo report*, the authors "*... define independent operators as actors that maintain distinct infrastructure without any technical overlaps such as ip history*".

Common knowledge from the malware domain states that "*actors* create `ExploitFiles`, which embed `PayloadFiles`, which connect to `WebServers`, described by either their *IP* or *domain name*". So, we could define rule 1.2 to test the following hypothesis: "Distinct exploit files embedding payloads connecting to the same web servers come from the same actor".

Rule 1.2. Object property *sameActorAs*

```
vtinv_ip:sameActorAs(?ef1,?ef2):-
vtinv_ip:File(?ef1), vtinv_ip:File(?ef2),
```

```
vtinv_ip:embeds(?ef1,?pl1), vtinv_ip:embeds(?ef2,?pl2),
vtinv_ip:connectsTo(?pl1,?ws), vtinv_ip:connectsTo(?pl2,?ws),
FILTER(?ef1 != ?ef2) .
```

Table 5 lists the results from the previous reasoning step, linking different `ExploitFiles` to the same *Actor*. These relationships are compatible with the links between tables 3, 6, 8, 12 and 15 from the *ItCo report*.

Such results also reveal another interesting fact: some *operators* leveraged `ExploitFiles` targeting *distinct* `cves`. For instance, APT 18 and APT 20, which is consistent with page 14 of the *ItCo report* "*...the operators known as Webby/APT18 and APT20 quickly deployed both CVE20155119 and CVE20155122 exploits.*"

Table 5. Actors and ExploitFiles from different clusters

Actor	**Table**	**ExploitFile**	**Cluster**	**WebServer**
APT18	3	079a440bee0f86d8a59ebc5c4b523a07	HT	223.25.233.248
	6	726bd0bd6cca8d481cf6165c95528caa	002	
UNK1	6	b65076f4cb6e74429dd02fcacda0bec3	002	nfitsub.com
	6	8a8e9bbf1ca2a926f0a5d06217eeea55	002	
	3	f46019f795bd721262dc69988d7e53bc	HT	
APT20	8	c101d289d36558c6fbe388d32bd32ab4	EXP1	win7.myz.info
	12	195bdc84f114c282e61f206dc88cd26d	EXP2	
DNSCalc/APT12	15	edcd313791506c623d8a2a88b9b0e84c	MOVIE	213.186.164.211
	15	83388058055d325a2fa5288182a41e89	MOVIE	202.183.129.155
UNK10	6	451c52652ddb28e9071078f214a327a7	002	amxil.opmuert.org
	6	e33cf5b9f3991a8ee4e71f4380dd7eb1	002	

5 Validation

In the previous Section, our aim was to demonstrate that the investigation rationale of the authors of the *ItCo report* could be translated into semantic classes, properties and rules from a semantic prototype. Therefore, feeding the prototype with the same dataset would generate equivalent results, as it did.

For validation purposes, though, it is necessary to analyse new malware samples with the same approach and rules, to find out if novel relationships could be uncovered. So we searched *VirusTotal* for malware binaries exploiting either *CVE-2015-5119* or *CVE-2015-5122* vulnerabilities. It was possible to download 152 novel samples (i.e. whose MD5 hash values did not match any of the files analysed in the *ItCo report*.

Such files went through the same process of extracting *ActionScript* classes and `PayloadFiles`, calculating Jaccard similarity scores and converting to a semantic format. The only extra step was running the extracted PayloadFiles

in the sandbox engine *Cuckoo*[23], in order to capture connections attempts to remote `WebServers`. Finally, they were added to the existing knowledge base.

5.1 Results

After applying Iterations 1 and 2 to the expanded dataset, the following results were obtained:

Table 6. New binary samples fitting in existing clusters.

IT[a]	Name	EF[a]	JSS[a]	CO[a]	AS[a]	CVE	ItCo equivalent
1	cluster-1-8	12	1	LZMA	8	2015-5122	flash_exploit_002
1	**cluster-1-7**	**31**	1	LZMA	7	2015-5119	HT_Exploit
1	cluster-1-50	5	1	ZLIB	50	2015-5119	MOVIE
2	**cluster-0.9-52**	**10**	0.9	ZLIB	52	2015-5119	EXP1
2	cluster-0.9-57	2	0.9	ZLIB	57	2015-5122	EXP2

[a] IT: *Iteration*, EF: *number of files*, JSS: *Jaccard score*, CO: *Compression*, AS: *number of ActionScript classes*

There were 22 new samples in cluster-1-7 (which means they were produced from the same generator as the other cluster members), and 4 new samples in cluster-0.9-52 (meaning they share most part of their source code with the other cluster members).

Relating to Iteration 1, there were also discovered 5 new clusters, sharing 2, 3, 12, 31 and 51 identical classes. However, due to both their small number of members (4 of them had only 2 members, and 1 had 4) and also the lack of identified connection attempts with remote `WebServers`, they were not considered for analysis. For similar reasons, 2 clusters containing 28 and 30 classes respectively were discarded in Iteration 2.

Finally, Table 7 lists the supplementary results of Iteration 3. In addition to 4 "new" actors (i.e. not identified by the authors of the *ItCo report*) employing `ExploitFiles` from the same generator (as they belong to *cluster-1-7*), one novel relationship is worth noticing: based on the hypothesis regarding *independent actors*, it seems that UNK5 and UNK11 actually correspond to the same author. Moreover, this actor would be producing `ExploitFiles` targeting both vulnerabilities 2015-5119 and 2015-5122 and, in either case, leveraging generators rather than sharing source-code.

6 Conclusion

In this paper, we have demonstrated an interesting use of semantic technologies: aiding malware investigation. We reproduced the steps taken by two researchers in analysing different pieces of malware in order to find relationships which could reveal a common origin or indicate different criminals employing them.

[23] Cuckoo Sandbox is an automated dynamic malware analysis system. *Source:* https://github.com/cuckoosandbox/cuckoo, *accessed in* 27/04/2016.

Table 7. New and existing actors.

Actor	Exploit File	Cluster	WebServer
NEW1	95d6c542a56e854562a3d09ca09df629	cluster-1-7	115.231.12.242
	ed4bb4f0a1c0e2a215e90023d2b34582	cluster-1-7	
NEW2	671fde3dcbc20506df37b60d2d1fde3e	cluster-1-7	119.29.25.169
	9581b0bd22b51f0fd40c4664b4e1bed0	cluster-1-7	
	acfc57b47338665dfd36ba1b693556d9	cluster-1-7	
	db3d8ec31054ff2e8844317d07eeee55	cluster-1-7	
NEW3	196280ad5f3e00d88e2d11a52415baf5	cluster-1-7	120.24.183.176
	70e614ad3264f66bd053f7f72e488a60	cluster-1-7	
	8669d94a06d26c938f1f2a0d01217346	cluster-1-7	
	c2fd972f68d2dea4b0d3c350e08aaf2b	cluster-1-7	
NEW4	95d6c542a56e854562a3d09ca09df629	cluster-1-7	yun.160mama.com
	ed4bb4f0a1c0e2a215e90023d2b34582	cluster-1-7	yun.cnhpyx.com
UNK5	00591821f328911380277272164d08cd	**cluster-1-7**	211.226.71.4
UNK11	b1238ccbb10af3e81110d3afacd98161	**cluster-1-8**	

By materialising the domain knowledge and thinking process of the authors into semantic entities and rules and applying them iteratively to a initial dataset, it was possible to arrive to the same conclusions, with the benefit of seamlessly scaling such "insight-building process" to a bigger dataset of malware. We have demonstrated this by loading new samples to the knowledge base.

Although only 26 out of 152 files were classified within the existing clusters, the main purpose of our paper is not check whether the rules suggested by the authors of the *ItCo report* are the most effective ones for identifying related malware. Rather, we defend that the task of defining and assessing different investigation hypotheses could be made more efficient using semantic technologies.

For instance, simply removing the requirement "files within a cluster must share the exact number of *ActionScript* classes" in Iteration 2 would have returned 19 more samples, some among which might indeed have common actor. Additionally, other concepts (e.g. class `URL`), object property `distributes` and data property `createDate`) could have been explored as well.

Although in its first version, we believe that the prototype ability to easily define new semantic concepts and apply bespoke rules to the knowledge base, in addition to rolling-back and assessing different hypotheses results, could increase the efficiency of any investigation task involving digital evidence, not only malware. We expect to have demonstrated the potential of our approach with this paper, and we will keep refining the prototype capabilities, as described in Section 6.1.

6.1 Future Work

This paper presented the first version of a prototype aimed at improving malware investigation tasks using semantic technologies. Some necessary follow-ups are: to gain a better understanding of the full potential of the RDFox reasoner, which

could turn both the investigation process and the prototype development more flexible; to automate exporting of the entities created in each iteration, in order to enable to "fast-forward" multiple states at once; finally, to keep testing the prototype with bigger datasets from different sources.

In terms of long-term objectives, we could enumerate:

1. Implement a GUI to facilitate the creation of new rules and entities and, more importantly, to provide a visual representation of the results of each iteration;
2. Interview malware experts in order to define the most relevant rules to be applied and also some basic ones to serve as demonstration;
3. Run lab experiments with different investigators to assess the tool usability and effectiveness.

Acknowledgements

We would like to thank Dr.Robert Piro and Dr. Yavor Nenov for their help on technical details regarding the RDFox reasoner. The first author's DPhil programme is funded by CAPES-CSF and supported by the Brazilian Federal Police.

References

1. ICSG Malware Metadata Exchange Format Working Group, http://grouper.ieee.org/groups/malware/malwg/Schema1.2/
2. The OpenIOC Framework, http://www.openioc.org/
3. Phishing Activity Trends Report - 1st-3rd Quarters 2015, https://docs.apwg.org/reports/apwg_trends_report_q1-q3_2015.pdf
4. RDF 1.1 Turtle, https://www.w3.org/TR/turtle/
5. Three US hospitals hit by ransomware, http://www.bbc.co.uk/news/technology-35880610
6. What Analysts Can Learn From Shadowservers Italian Connection Report CYINT Analysis, http://www.cyintanalysis.com/what-analysts-can-learn-from-shadowservers-italian-connection-report/
7. Armerding, T.: Cybercrime: Much more organized (Jun 2015), http://www.csoonline.com/article/2938529/cyber-attacks-espionage/cybercrime-much-more-organized.html
8. Ben Koehl, Ned Moran: The Italian Connection: An analysis of exploit supply chains and digital quartermasters, https://drive.google.com/file/d/0Bw35r_AUUldgMEZUdXUyWUR0T3M/view?usp=sharing
9. Carvalho, R., Goldsmith, M., Creese, S.: Applying Semantic Technologies to Fight Online Banking Fraud. pp. 61–68. IEEE (Sep 2015), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7379724
10. Italian National Research Council: EVIDENCE Semantic Structure. Tech. rep., European Informatics Data Exchange Framework for Courts and Evidence (Jan 2015), http://s.evidenceproject.eu/p/e/v/evidence-ga-608185-d02-1-final-31072014-136.pdf

11. Kirillov, I., Beck, D., Chase, P., Martin, R.: Malware attribute enumeration and characterization. The MITRE Corporation, Tech. Rep (2010), https://maec.mitre.org/about/docs/Introduction_to_MAEC_white_paper.pdf
12. Mercs, F.: The Brazilian Underground Market. Tech. rep., http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-the-brazilian-underground-market.pdf
13. Mundie, D.A., McIntire, D.M.: The MAL: A Malware Analysis Lexicon. Tech. rep. (2013), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6226344
14. Mundie, D.A., Mcintire, D.M.: An Ontology for Malware Analysis. pp. 556–558. IEEE (Sep 2013), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6657289
15. Nenov, Y., Piro, R., Motik, B., Horrocks, I., Wu, Z., Banerjee, J.: RDFox: A Highly-Scalable RDF Store. In: The Semantic Web-ISWC 2015, pp. 3–20. Springer (2015), http://link.springer.com/chapter/10.1007/978-3-319-25010-6_1
16. Ragan, S.: Hacking Team hacked, attackers claim 400gb in dumped data (Jul 2015), http://www.csoonline.com/article/2943968/data-breach/hacking-team-hacked-attackers-claim-400gb-in-dumped-data.html
17. Steven Malby, Robyn Mace: Comprehensive Study on Cybercrime. Tech. rep., United Nations Office on Drugs and Crime (2013), http://www.unodc.org/documents/organized-crime/UNODC_CCPCJ_EG.4_2013/CYBERCRIME_STUDY_210213.pdf