Investigating Malware Campaigns with Semantic Technologies

Rodrigo Carvalho*†, Michael Goldsmith*, Sadie Creese*
*Department of Computer Science
University of Oxford, Oxford, UK
{rodrigo.carvalho, michael.goldsmith, sadie.creese}@cs.ox.ac.uk
†Brazilian Federal Police, Brasília, Brazil
carvalho.rac@dpf.gov.br

Abstract—Malware-campaign investigation is a major factor in fighting cybercrime. Most of the research in this area comes from commercial companies, so potentially there is a greater emphasis on detection rather than malware attribution. Aiming at a better balance between human reasoning skills and computer processing capabilities, our hypothesis is that semantic technologies could greatly enhance the investigation of the malware ecosystem.

To demonstrate this, we have reproduced using our prototype the investigation rationale described in a report from 2015 . The transparent methodology regarding the analysis of 52 files allowed us to translate the domain knowledge of the authors of the report into entities, relationships and rules. As an extension to the case study, we have further enriched our knowledge base with 155 new files. This led to the prompt identification of relationships and patterns among the new dataset entities in a semi-automated, scalable fashion, confirming our hypothesis.

1 Malware cybercrime investigation

Cybercrime is a major problem which is directly associated with the current high level of connectivity and data ubiquity within our society. Surveys from organizations such as Anti-Phishing Working Group(APWG) [1] inform about its alarming rates:

- "The total number of unique phishing[1] sites observed in the second quarter of 2016 was 466,065. This was an all-time high."
- "APWG member PandaLabs found 18 million new malware samples in Q2 , an average of more than 200,000 a day. This is 10 percent lower than in the previous quarter, when 20 million new samples were found."

One of the proposed reasons for such widespread cybercrime is that there are many places on the Internet in which cybercriminals trade malware and infrastructure services, such as botnet[2] rental, at competitive prices, see [2].

AV vendors, who form a large part of the security community, tend to tackle this problem by detecting and black-listing the malicious files and web servers distributing them. This makes sense since their main commercial focus is securing the computers of their clients, and not investigating the associated malware campaigns.

Determining the attribution of a malware campaign is typically the responsibility of Law Enforcement Agencies (LEAs) **because it is a criminal act. However, possibly due to the** difficulty of such task, most agencies focus on disrupting the computer infrastructure used to spread malware. Operation Avalanche [3] is a one of the most recent documented cases.

Although demanding a major cooperative effort spanning several jurisdictions, disruption is an effective strategy to stop malware distribution at large scale, ultimately leading to increasing the costs of cybercrime activity. However, criminals on the loose can always hire different infrastructure and acquire upgraded malware, in a continuous "cat and mouse" game. **We believe that more productive investigations could not only bring them to justice but also act as a deterrent.**

One of the most challenging issues in investigating a malware campaign is producing comprehensive evidence against its perpetrators. Often, probative and court-admissible evidence can only be found after seizing the devices owned by the criminal organizations. Obtaining such devices is a complex task, even if they are within the same jurisdiction of the LEA: investigators must hypothesize about a high volume of heterogeneous, supposedly

1. 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* 26/11/2016.

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

unrelated Indicators of Compromise[3] (IOCs) for patterns and relationships which could eventually support a search warrant.

More than ever, "Todays tools must be re-imagined to facilitate investigation and exploration" [4]. We address this particular challenge with a prototype that leverages semantic technologies to integrate such horizontally-spread data and facilitate the insight-building process of the analysts. **The purpose of our research being to explore and demonstrate the utility of semantic technologies in this domain.**

2 Semantic Technologies

It is probably easier to explain what semantic technologies are if we present the idea behind the Semantic Web as defined by Tim Berners-Lee: "The Semantic Web is not a separate Web but an extension of the current one in which information is given well-defined meaning, better enabling computers and people to work in cooperation" [5].

Such well-defined meaning is given to information through an ontology, which aims to explicitily define meanings shared by a community [6]. The basic components of an ontology are:

- **Classes**: describe concepts, or the type of things within a domain. In our case study, `WebServer` and `File` are two classes, and `PayloadFile` is a subclass of the latter, inheriting all its properties.
- **Properties**: relationships between members of different classes (*object properties*) or between members of a class and a literal (*data properties*). Some examples would be *PayloadFile* `connectsTo` *WebServer* and *File* `name` *string*, respectively.
- **Individuals**: the instances of a class (or the objects described by it). For instance, *evil.org* could be one instance of the class `WebServer`. Often, the Individuals are not considered part of the ontology *per se*, but together with it form what is called a *knowledge base*.

Classes, properties, individuals and all the other components of a knowledge base are logically stored as triples (*subject predicate object*), describing one statement each. Triples are represented using the Resource Description Framework[4](RDF). RDF uses web-based URIs to name the relationship between things as well the things themselves, allowing structured and semi-structured data to be integrated and shared across different applications.

Listing 1 presents the components of a sample ontology as triples in *Turtle* syntax. *Turtle* is a RDF serialization which is compact, easy to read by humans and also a W3C recommendation:

Listing 1: Ontology definition in Turtle syntax.

```
# Prefixes for standard namespaces and bespoke ontology
@prefix owl: <http://www.w3.org/2002/07/owl/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema/> .
@prefix onto: <http://ontology.com/> .

# Classes and subclasses
onto:WebServer rdf:type owl:Class .
onto:File rdf:type owl:Class .
onto:PayloadFile rdf:type owl:Class ;
   rdfs:subClassOf onto:File .

# Object property
onto:connectsTo rdf:type owl:ObjectProperty ;
   rdfs:domain onto:PayloadFile;
   rdfs:range onto:WebServer .

# Datatype property
onto:name rdf:type owl:DatatypeProperty ;
   rdfs:domain onto:File ;
   rdfs:range xsd:string .

# Individuals
<http://ssreport.com/1a2b3c4d5e6f> rdf:type onto:PayloadFile .
<http://ssreport.com/1a2b3c4d5e6f> onto:connectsTo
<http://ssreport.com/evil.org> .
<http://ssreport.com/1a2b3c4d5e6f> onto:name "malware.exe"
```

As Listing 1 illustrates (through the PREFIX clauses), one of the key characteristics of the Semantic Web is that classes and properties are defined by Uniform Resource Identifiers (URIs). This ensures that they have an unique definition accessible by anyone on the Web, avoiding concept misunderstanding across different contexts [6].

For instance, one web-master could mark-up the universities described in her site using the definition present in `https://schema.org/Organization`. This would enable any software agent aware of *schema.org* (e.g. a search engine) to recognize "University of Oxford" as an organization, which would then be a "thing" instead of a string. Consequently, this "thing" could now hold the relationship `https://schema.org/employee` with multiple individuals marked-up as `https://schema.org/Person`.

Of course, "The computer doesn't truly 'understand' any of this information, but it can now manipulate the terms much more effectively in ways that are useful and meaningful to the human user." [5]. A very good example is linking disparate resources describing the same "thing", which could be very useful for data enrichment while keeping its provenance. This will be further discussed in Section 5.4 below.

3 The Investigation of Malware Campaigns

Our objective is not to suggest an ontology for malware investigation **in the terms of the aforementioned *schema.org*. There are already comprehensives taxonomies whose concepts could be used to define one, such as STIX[5], OpenIoc [6] and MISP[7]. Their development is a consequence of the**

3. 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. *Source:* https://en.wikipedia.org/wiki/Indicator_of_compromise, *accessed in* 07/05/2017.

4. RDF is a standard model for data interchange on the Web. RDF has features that facilitate data merging even if the underlying schemas differ, and it specifically supports the evolution of schemas over time without requiring all the data consumers to be changed. *Source:* https://www.w3.org/RDF , *accessed in* 07/05/2017

5. Structured Threat Information Expression. https://oasis-open.github.io/cti-documentation

6. https://github.com/mandiant/OpenIOC_1.1

7. Malware Information Sharing Platform – https://github.com/MISP/misp-objects

modern understanding that cyber-threat intelligence cannot be effectively represented as an uncontextualized flat list of IOCs, as it describes complex and richly connected entities [7].

However, it is debatable whether current threat-intelligence platforms, whether leveraging such taxonomies or processing raw IOCs, are actually producing intelligence. Most of them still focus on data collection rather than analysis, which is often limited to searching, browsing and attribute-filtering [8] as opposed to knowledge and intelligence creation.

Thus, we consider our approach a step towards completing the intelligence cycle, as it aims to facilitate the insight-building process of the investigator during the analysis stage. For that, we believe semantic technologies could excel in the following tasks:

- Easy data integration across disparate data sources: because it might be infeasible for all stakeholders to adopt a particular schema, a straightforward way to define mappings between them is recommended;
- Facet querying: **in addition to interactive filtering, it helps understanding the relevant dimensions of a particular domain [9];**
- Provenance, or efficiently tracing back any piece of information to the original data source.

Szekely et al describe how such tasks have improved the investigation of human trafficking in a law-enforcement setting [10]. We agree but would go further as we consider that other native capabilities could further enhance relationship-searching and hypothesis-testing within the malware-campaign investigation domain, as detailed in Table 1.

TABLE 1: Mapping needs of an investigation to compatible semantic capabilities.

	Common investigation needs	Compatible semantic technologies capabilities
1	Clustering entities	Defining classes with property restrictions
2	Establishing links	Creating object properties
3	Inserting tags or comments	Updating datatype properties
4	Merging data about entities	Materialising object property "owl:sameAs"
5	Reproducing investigation steps	Reapplying saved queries and rules
6	Rolling back in case of dead end	Loading previous version of KB into the Triple Store

Features 1 to 4 are mostly used for building knowledge bases before the users of linked-data exploratory systems can browse and search them. This holds true for most systems described in [9] which, despite sharing some similar objectives with our prototype, do so using different techniques. For instance, using in-session memory to memorize the user browsing sequences and suggesting queries to inspire exploration.

The novelty of our approach lies in actually allowing the user to manipulate these features in order to shape knowledge which would inform his investigation hypotheses. In addition, we believe Feature 5 could be useful for automation and reproducibility purposes, and Feature 6 to avoid repeating prior investigative steps. In order to implement these semantic capabilities, a Triple Store is necessary.

Triple Stores and RDFox

Triple Stores (or RDF Stores) are a kind of graph database with the added benefits of knowledge-inferencing and materialisation using rules and set processing. Graph databases are more suited to store and query highly-connected data than relational database systems, facilitating analysis tasks in which relationships and connection patterns between different items is important, see [11] for discussion.

Although native graph databases outperform Triple Stores in data-processing performance, the latter are better suited for information-modelling and sharing. Moreover, their inherent capability of integrating datasets is relevant, as current threat-intelligence platforms provide limited automated data-integration support, see [8] for explanation.

We consider that leveraging a graph technology which favours data integration contributes to investigation pivoting, described by [7] as the fundamental analytic task of *hypothesis-testing* which relies on the ability of analysts to *understand the relationship between elements from distinct data sources*. Therefore, we have adopted Triple Stores.

In order to assess the benefits and the feasibility of the ideas discussed in this section, we have implemented the capabilities from Table 1 in a prototype that leverages RDFox, which is a main-memory Triple Store coupled with a parallel datalog reasoner[8] , from the University of Oxford. In addition to supporting *sparql* (a sql-like query language for RDF, based in graph pattern matching and endorsed by W3C) and handling the "owl:sameAs" object property efficiently, RDFox is very fast at computing and updating data materialisations [12]. **The importance of these features to our approach will become clear in Section 5, which reproduces the investigation presented in the next section.**

4 The *Italian Connection* Report

A significant cyber-security event happened in 2015, in which data from the company Hacking Team was leaked to the internet. Among the exposed files, there were some zero-day vulnerabilities which were promptly harnessed by other hackers.

Following this breach, researchers at Shadow Server Foundation published a report [13] aiming to revealing relationships between supposedly independent groups based on similarities across IOCs such as payload[9] and command and control infrastructure. This report will henceforth be referred as *ItCo*, in reference to its original name "The Italian Connection".

Differently from most reports describing malware campaigns, the *ItCo* report clearly states the methodology used (which investigative step was taken at each stage, and why), explains the rationale behind each assertion they make (e.g.

8. **A semantic reasoner, reasoning engine, rules engine, or simply a reasoner, is a piece of software able to infer logical consequences from a set of asserted facts or axioms. The notion of a semantic reasoner generalizes that of an inference engine, by providing a richer set of mechanisms to work with. The inference rules are commonly specified by means of an ontology language, and often a description logic language. *Source:* https://en.wikipedia.org/wiki/Semantic_reasoner, *accessed in* 07/05/2017**.

9. 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.

"We define independent operators as actors that maintain distinct infrastructure without any technical overlaps such as ip history.") and concludes by presenting competing hypotheses regarding whether different exploits might have a unique origin.

The dataset used by the authors of the *ItCo* report comprises a total of 52 Adobe Flash files exploiting either CVE[10]-2015-5119 or CVE-2015-5122, which are the vulnerabilities related to Hacking Team leaks. These samples were obtained by crawling websites known for distributing malware and also searching online repositories such as *VirusTotal* and *Shadow Server*. For each sample, IOCs such as domains, ips, hashes and compression method were extracted by static and dynamic sandbox[11] analysis.

The first step of the investigation was to distinguish the exploit files produced by a common *generator tool*[12] from the ones obtained via *source-code sharing*. According to the authors' knowledge, exploit files produced by the same generator tool share specific features. For instance, they used the following values to define one of the clusters:

- Created on the same date of 7/7/2015;
- Targeted the same vulnerability of CVE-2015-5119;
- Compressed via the LZMA algorithm;
- Contained an embedded payload;
- Had identical *ActionScript* classes.

All exploits produced by one *generator tool* would necessarily have all embedded *Action Script* classes identical. However, some other exploits, despite sharing most features with each other, would have one or more differing classes. In these cases, the authors of the *ItCo* report would classify such exploits as *source-code sharing*.

The report does not mention if an automated method was used to compare the classes from each file. Our approach was to first automatically extract all the classes from each file and produce their MD5 hashes. Then, we calculated the Jaccard Similarity Score (JSS) for every two files, using their internal classes MD5 hashes as the set of features. The pairs with a score of 1 were attributed the relationship `hasJSS-1.0-with`, and those with a score between 0.9 and 0.99, `hasJSS-0.9-with`.

By assessing the different clusters created (two *same-generator* clusters and three *shared source-code* ones) and which of their members would potentially have a unique origin (i.e. the same actor), **the authors established competing hypotheses about the exploits supply-chain. These were mostly related to the malware-development skills of distinct actors, which groups would be "collaborating" among themselves and the speed with which their exploits were deployed into the wild.**

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

11. 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.

12. 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." [13]

We consider the *ItCo* report an excellent opportunity to validate our approach, presented before in a position paper [14]. At that time it was difficult to implement due to the lack of specific domain knowledge (i.e., the rationale used by malware investigators) and the difficulty of obtaining suitable datasets for the case study (now made available by the authors of the report as a spreadsheet containing all the IOCs).

The following sections will demonstrate how we have fully reproduced the *ItCo* report by harnessing the capabilities mentioned in Table 1 in our prototype.

5 Semantic Investigation

First, it is necessary to define our ontology, which will be the basis for converting data to a linked format. We can either extend the concepts from an existing ontology or create a new one.

We decided to create a new ontology for two reasons:

- **There is not a widely accepted ontology for malware investigation, but only taxonomies which do not follow semantic-technology standards;**
- **To demonstrate that designing a basic ontology with a few classes and properties, using an editor such as *Protege*[13], is not a difficult task.**

The *datatype properties* of our ontology derived from the column headers of the IOCs spreadsheet are shown in Table 2. The first 4 rows also represent *classes*, as we have chosen `md5` and `domain` to compose the URI of the entities `File` and `Webserver`, respectively.

TABLE 2: **Mapping IOCs to semantic concepts.**

Column headers	Classes and datatype properties	Sample values
swf md5	File (Exploit File) and md5	e33cf5b9f...71f4380dd7eb1
payload md5	File (Payload File) and md5	5a22e5aee...77f1351265a00
exploit site	WebServer (ExploitServer) and domain	news.turkceil.tk/movie.swf
C2	WebServer (C2Server) and domain	amxil.opmuert.org
payload in swf	embeddedPayload	yes
create date of swf	createDate	7/14/15
swf cve	cve	CVE-2015-5122
swf compression	compression	lzma
payload family	family	PlugX

Figure 1 illustrates the resulting ontology in *vowl*[14] notation: circles are *classes* and green rectangles are *datatype properties*. The blue rectangles represent *object properties*, or bespoke relationships between *classes* reflecting the knowledge disclosed by the authors in the report. Note there are no individuals yet.

Once the ontology is ready, it is necessary to map the data (in our case, the information from the IOCs spreadsheet) onto it. This was accomplished using the tool *Karma*[15], which provides an intuitive interface to create the mapping model and later allows for batch processing.

The resulting linked data resembles the individuals depicted in Listing 1: *ssreport.com* is the bespoke namespace we have defined for our data, and *vtinv* is the ontology comprising, among others, the property `connectsTo`. *1a2b3c4d5e6f* and *evil.org* are the identifiers for one `File` and one `Webserver` within our namespace, respectively.

13. http://protege.stanford.edu
14. http://vowl.visualdataweb.org
15. http://usc-isi-i2.github.io/karma

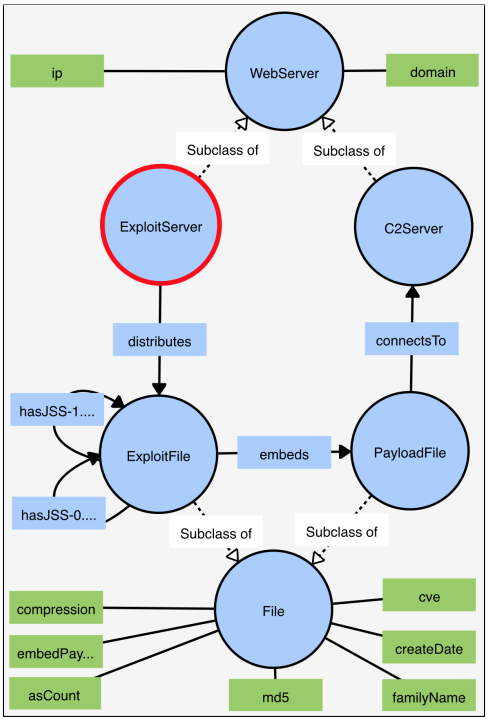

Fig. 1: The vtinv ontology.

Finally, the knowledge base (comprising both the ontology and the data) was loaded into our prototype. The initial number of individuals is shown in Table 3, column 0.

TABLE 3: Evolution of the number of individuals. **Bold denotes a change in value, and '-' means the concept is not present in that iteration.**

Iteration	0	1	2	3	4	5	6	7
File	74	**43**	43	**70**	**43**	**198**	**161**	**137**
new_HT	-	**11**	11	11	11	11	11	**31**
new_002	-	**12**	12	12	12	12	12	12
new_exp1	-	**6**	6	6	6	6	6	**10**
new_exp2	-	**2**	2	2	2	2	2	2
WebServer	65	65	65	**81**	**71**	**109**	**103**	103
hasJSS-1.0	230	230	230	230	230	**1174**	**970**	970
hasJSS-0.9	150	150	150	150	150	150	150	150
connects	42	42	42	**62**	**50**	**96**	**91**	91
distributes	36	36	36	36	36	36	36	36
embeds	37	37	37	37	37	**73**	73	73
sameC2as	-	-	**14**	14	14	14	14	**38**
sameC2ESas	-	-	**2**	**-**	-	-	-	-

The number of *Files* in each iteration includes both `ExploitFiles` and their embedded `PayloadFiles` (obtained from the columns *swf md5* and *payload md5* of the IOCs spreadsheet). Therefore, there is no conflict with the initial count of exploit files given in Section 4 (52).

5.1 Semantic Facet querying

As mentioned before, the initial step of the authors of the *ItCo* report was to cluster files potentially created by the generator. After exploring the data, they come to the cluster definition given in Section 4.

Although the report does not mention their exploratory process, we could simulate it using facet querying: we firstly queried for `Files` which hold the relationship `hasJSS-1.0-with` with another `File`. Analysing the results it was clear to notice that `Compression` and `CVE` would be good features for clustering.

This is a basic use of the facet querying, and the fact that we are analysing previously processed data certainly made it easier to spot good cluster features. Hence, a more complex query will be demonstrated below in Subsection 5.3, and Subsection 5.4 will demonstrate our approach within a larger dataset.

Our prototype allows for semantic facet queries to be built in a systematic fashion. Figure 2 illustrates the query builder interface, in which the content of the drop-down boxes is updated according to the results of the previous run. It is possible to incrementally append relationships between instances (first column) and restrictions to *datatype properties* using either existing literals (second column) or logical and numerical operators (third column).

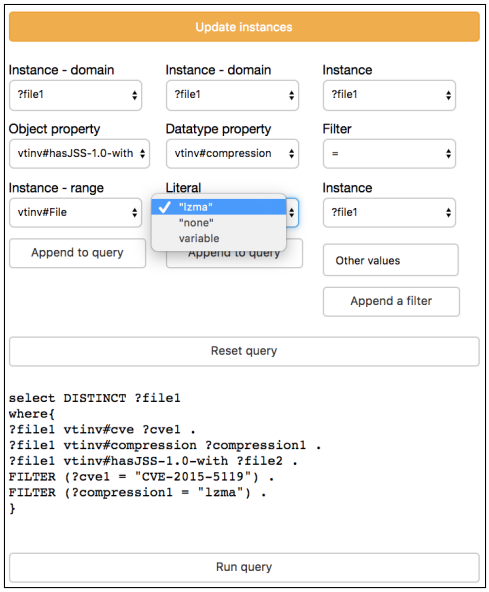

Fig. 2: The query builder. Items with "?" represent placeholders for distinct class instances, and items in quotes are literals.

Running the query from Figure 2 on the knowledge base yielded the same results as Table 1 from the *ItCo* report (which enumerates the members of the *HT_exploit* cluster). If deemed relevant for the investigation, the individuals returned by the query can easily be grouped into a new class.

5.2 Clustering or tagging entities

The similar syntax of *sparql* queries and the *datalog* rules expected by RDFox makes it easy to transform the previous query into a class definition. The rule depicted in Listing 2 is automatically generated by our prototype, and could be applied any time new data is loaded into the knowledge base.

Listing 2: Rule derived from the *sparql* query in Figure 2.

```
vtinv:new_HT(?file1) :-
```

```
vtinv:compression(?file1, ?compression1),
vtinv:hasJSS-1.0-with(?file1, ?file2),
vtinv:cve(?file1, ?cve1),
FILTER (?cve1 = "CVE-2015-5119"),
FILTER (?compression1 = "lzma")
```

Thereafter, it is necessary to materialize the new facts resulting from this rule into our knowledge base. This is accomplished by invoking the reasoner, and would result in a new class `vtinv:new_HT`, subclass of `vtinv:File`, containing all individuals matching the rule.

Likewise, we could simply tag the individuals returned by the previous search. Suppose we want to attribute a confidence level regarding their origin. Then, instead of creating a new class, we could define the datatype property `confidence_generator` with the value *high*. To accomplish that, it would be enough to update the first line of Listing 2 to:

Listing 3: Defining or updating a datatype property.

```
vtinv:confidence_generator(?file1, 'high') :-
```

We believe that this feature, in addition to the native property `rdfs:comments`, could also be useful for easily documenting and retrieving non-technical information, such as the *Tactics, Techniques, and Procedures* (TTPs) of the cybercriminals.

Going back to the exploration process mentioned in Section 5.1, we could identify one more *same generator* cluster, comprising files exploiting *CVE-2015-5122* and compressed with the *lzma* algorithm. These coincide with cluster flash_exploit_002[16], described in Table 4 of the *ItCo* report.

In a similar fashion, two classes of *shared-code* exploits were created: *new_exp1*[17] (Table 7 from *ItCo* report) and *new_exp2* (Table 10 from *ItCo* report). The updated number of individuals at this point in the investigation is given in Table 3, column (iteration) 1.

5.3 Establishing links

The authors of the *ItCo* report define independent actors as the ones who "...maintain distinct infrastructure without any technical overlaps such as ip history".

Thus, we could hypothesize that two `ExploitFiles` would potentially share the same actor if (1) they are being distributed by the same `ExploitSite` or (2) their embedded `PayloadFiles` connect to the same `C2Server`[18].

In order to find out about (1), a simple and direct query would be enough : *Find all `ExploitFiles` that are `distributed` by the same*[19] `ExploitServer`.

On the other hand, finding about (2) would require a "*join*-like" operation which is better handled by graph-based technologies. After all, it would involve traversing two relationships (`ExploitFile` *`embeds`* `PayloadFile` and `PayloadFile` *`connectsTo`* `C2Server` – see Figure 1) for all `ExploitFiles` in the knowledge base to find out which of them share the same `C2Server`.

The native linking structure of the triples makes it easier to define this *sparql* query as depicted in Listing 4:

Listing 4: **Querying for different ExploitFiles (?file1 and ?file3) with matching C2Servers (?webserver1).**

```
select DISTINCT ?file1 where
?file1 vtinv:embeds ?file2 .
?file2 vtinv:connectsTo ?webserver1 .
?file3 vtinv:embeds ?file4 .
?file4 vtinv:connectsTo ?webserver1 .
FILTER (?file1 != ?file3) .
```

This search returned 11 different `ExploitFiles`. If we consider the results relevant, we can transform this query into a rule for defining the new object property `sameC2as`, **similarly to what we did in Listing 2. Feeding this rule to the reasoner resulted in the materialized knowledge depicted in Figure 3. The yellow edges represent the recently created instances of the *sameC2as* property, connecting the `ExploitFiles` returned in the original query. These files belong to different clusters (new_HT, new_002, new_exp2 and new_exp1, derived from the materializations described in Section 5.2). In addition to listing their *md5* hashes, Table 4 also informs that our conclusions regarding `ExploitFiles` belonging to the same actor agree with ones from the *ItCo* report: column *ItCo table* indicates the original tables comprising these files and their matching `C2Server`.1**

Rolling back

Creating bespoke relationships allows us to promptly reuse them as necessary. For instance, we could restrict the definition of *same actor* **by considering (1) *and* (2) instead of (1) *or* (2). The new `sameC2ESas` rule, building upon `sameC2as`, would then be:**

Listing 5: Defining a more restrict definition of *same actor*.

```
vtinv:sameC2ESas(?file1, ?file3) :-
vtinv:distributes(?webserver0, ?file1),
vtinv:distributes(?webserver0, ?file3),
vtinv:sameC2as(?file1, ?file3) .
```

Once materialised, this last rule added two instances of the object property `sameC2ESas` to the knowledge base, linking two distinct `ExploitFiles` which happen to embed the same `PayloadFile`. The updated counts of entities at this point in the investigation is shown in Table 3, column (iteration) 2.

Suppose, however, that the analyst decides that the recently created relationship does not add value to the investigation. In that case, it would only be necessary to load the version prior to this materialization, and apply new queries or rules from there on.

Next, we will extend the *ItCo* report by enriching the original dataset, which will conveniently demonstrate the other capabilities mentioned in Table 1.

5.4 Data enrichment

Eventually it may be necessary to add data from other sources during the course of an investigation. There are

16. Contains one less file, not mentioned in the written report but present in the spreadsheet.

17. Contains one less file. This was expected, since it has compression value none.

18. Command-and-control (C&C) servers are used to remotely send often malicious commands to a botnet, or a compromised network of computers.*Source:https://www.trendmicro.com/vinfo/us/security/definition/command-and-control-(c-c)-server, accessed in 12/04/2017*

19. We will not consider domain resolution as it is not present in the original data. However, a more precise *Webserver* definition, considering its assigned IP on a particular day, could also be modelled.

TABLE 4: Correspondences between the *ItCo* report tables and our results in Figure 3, with regard to `ExploitFiles` supposedly from the same actor.

Actor[20]	*ItCo table*	ExploitFile	*ItCo cluster*	WebServer
APT18	3 6	079a440bee0f86d8a59ebc5c4b523a07 726bd0bd6cca8d481cf6165c95528caa	HT 002	223.25.233.248
UNK1	6 6 3	b65076f4cb6e74429dd02fcacda0bec3 8a8e9bbf1ca2a926f0a5d06217eeea55 f46019f795bd721262dc69988d7e53bc	002 002 HT	nfitsub.com
APT20	8 12	c101d289d36558c6fbe388d32bd32ab4 195bdc84f114c282e61f206dc88cd26d	EXP1 EXP2	win7.myz.info
DNSCalc/APT12	15 15	edcd313791506c623d8a2a88b9b0e84c 83388058055d325a2fa5288182a41e89	MOVIE MOVIE	213.186.164.211 202.183.129.155
UNK10	6 6	451c52652ddb28e9071078f214a327a7 e33cf5b9f3991a8ee4e71f4380dd7eb1	002 002	amxil.opmuert.org

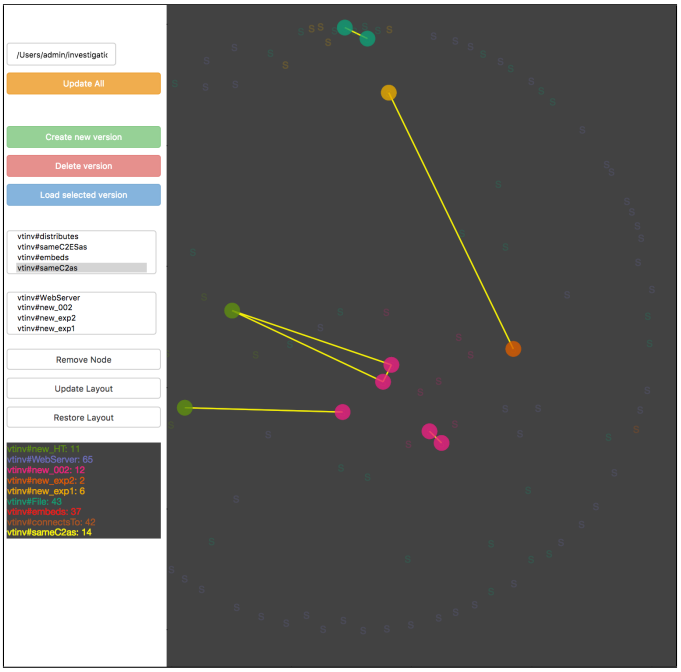

Fig. 3: Members from different clusters linked by the bespoke relationship `sameC2as`.

Fig. 4: Before owl:sameas: two resources (in green) referring to the same *file thing*.

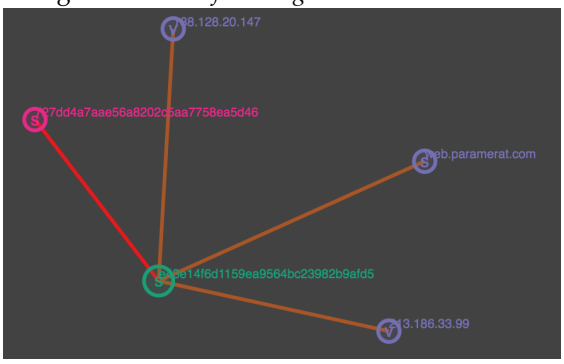

Fig. 5: After owl:sameas: integrated datatytpe properties.

two possibilities, which are not mutually exclusive: enriching data about existing entities or adding new entities

Regarding the former, we wanted to enrich the existing `PayloadFiles` with network data from *VirusTotal*. **After searching this repository using their md5 hashes, it was possible to download a total of 27 file reports.**

In the same way as described before, we must firstly define a mapping between our ontology and the *VirusTotal* network reports. For demonstration purposes, we have only mapped the object property `connectsTo`, and the datatype properties `ip`, `domain` and `md5`.

The resulting linked-data version comprised 20 `Files` (out of the initial 27) holding the relationship `connectsTo` with 16 `WebServers`. After defining the namespace "http://vt.com/", we merged it to our knowledge base and loaded it to our prototype.

As expected, the counts of `File`, `WebServer` and `connectsTo` have increased, as shown in Table 3, column (iteration) 3. Even though all of these "new" `File` instances refer to pre-existing `PayloadFiles`, they are still distinct resources (e.g. *http://ssreport.com/e43e14...b9afd5* and *http://vt.com/file-e43e14...b9afd5*), which might hold identical or complimentary information about the same *thing*: the file with md5 hash *e43e14...b9afd5*.

This is where the native object property `owl:sameAs` can be most useful: to link distinct resources referring to the same thing. Similarly to Section 5.3, we can use it to connect any two files with matching md5 hash values (Figure 4)[21]. The semantic consequence is that both resources will now share the conjunction of their respective object and datatype properties (Figure 5).

In our prototype, any two individuals linked by the `owl:sameas` property are graphically represented as one bigger circle. This feature was implemented to reduce

21. Please refer to Figure 3 for the color legend.

the visualization clutter, since both individuals still exist within their own namespaces (*http://ssreport.com* and *http://vt.com*, indicated in Figures 4 and 5 by the letters 's' and 'v').

Column (iteration) 4 of Table 3 shows that, after the materialization of the `owl:sameas` property, the quantity of files decreased to the same number as iteration 2 (i.e. before the addition of the network reports in iteration 3). This was expected, since we have only downloaded reports for files pre-existing in the knowledge base.

For demonstration purposes, we have also established the `owl:sameas` property for any two `WebServers` with matching `domain` or `ip`. Differently from `Files`, the difference between versions 2 and 4 indicates that there are six new `WebServers` in the knowledge base, out of the sixteen added in iteration 3.

5.5 Validation

So far, we have demonstrated how the semantic capabilities mentioned in Table 1 could facilitate data analysis and hypothesis-testing during the course of a malware campaign investigation.

Because all the steps taken were defined and saved as rules, it is possible to submit them together with the initial dataset to a third party for auditing purposes (e.g. to check whether the results of the current investigation are sound). In addition, it also allows other analysts to extend or tune them, in order to get a different view of the same dataset. We believe that such capabilities are compatible with the reported need for "producing and sharing intelligence instead of raw IOCs".

Further, following the procedure described in Section 4, we have searched *VirusTotal* for the tags *CVE-2015-5119* and *CVE-2015-5122* in April 2016. It was possible to download 155 new files, comprising both exploits and payloads.

After extracting the *ActionScript* classes from the Adobe Flash exploit files and calculating their JSS, the data was converted to linked format and merged into the knowledge base. Column (iteration) 5 of Table 3 gives the new individuals count, and column (iteration) 6 reflects the counts resulting after applying the `owl:sameas` property to both `Files` and `WebServers`.

Then, we quickly reapplied the rules defining the four clusters and the relationship `sameC2as`. After the last materialisation (column 7 of Table 3), cluster `new_HT` was automatically populated with twenty new members and cluster `new_EXP1` with four new ones.

Moreover, it was easy to spot three new sets of `ExploitFiles` holding the property `sameC2as` among themselves, as indicated in Figure 6. They are all members of cluster `new_HT`, endorsing the hypothesis in page 17 of the ItCo report: "The model of a single quartermaster developing and sharing generators would explain the identical nature of the malicious ActionScript classes in the HT_Exploit and flash_exploit_002 clusters."

6 Conclusion

We have demonstrated that our approach could indeed facilitate data exploration, making it flexible and fast enough so that analysts can focus on the analysis of the results, avoiding much of the time-consuming, error-prone work in manipulating and correlating data.

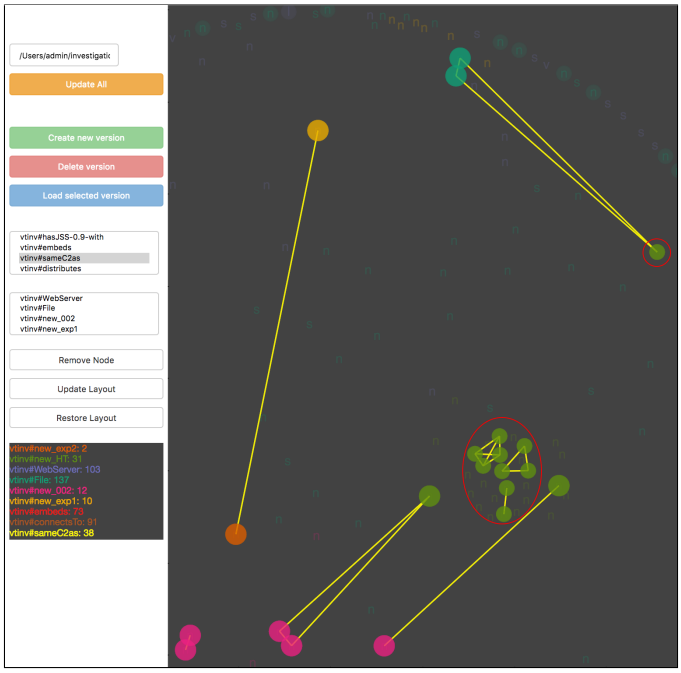

Fig. 6: The result of iteration 7 revealed ten new `ExploifFiles` holding the relationship `sameC2as`.

Our semantic prototype allows quick and semi-automated generation of new knowledge based on expressive rules defined by the analyst during the course of an investigation. This is our contribution towards insight-generating tools which improve the balance between human reasoning (able to perform deep analyses on few items at once) and computer processing (which can evaluate large amounts of superficial information, but struggles to identify complex patterns without prior training).

It is worth noticing that not a single line of code was necessary to perform the transformations described in the previous sections. Instead, we relied solely on native semantic technology capabilities.

Moreover, the fact that information about provenance of the individuals and the meaning of all classes and relationships are explicitly defined within the data (instead of hard-wired in the source code) facilitates the integration and exploration of distinct datasets.

Limitations and future work

We acknowledge that there is a considerable preparation step in our approach, mainly because we first need to convert the original dataset to a linked format. However, once we have produced a mapping file for a specific data source, the rest of the process can be automated.

Interestingly enough, the increasing adoption of JSON-LD[22] (JSON for Linked Data) within the web community (search engines such as Google are recommending it for

22. https://www.w3.org/TR/json-ld, http://json-ld.org

information mark-up [15]) could motivate online data providers to publish their JSON data in JSON-LD. After all, the compatibility between both formats means that only small modifications would be necessary to convert the former into the latter.

We believe that this fact, in addition to the active research regarding the scalability of current reasoners and benchmarks for linked data processing [11] could foster the development of novel semantic approaches to the exploration of the available data sources on the web.

Finally, we foresee as future work:

- **Improving the prototype capabilities for malware campaign investigation by modelling domain resolution and handling dates;**
- **Enhancing the user interface according to *visual analytics* principles. The current visualisation serves solely to demonstrate the features of our semantic approach;**
- **Testing the prototype with expert users, to assess trade-offs such as expressiveness against semantic integrity.**

7 Acknowledgements

We would like to thank VirusTotal for providing access to the binaries. This project is being developed during the first author's DPhil programme, which is funded by CAPES-CSF (award 9084/13-4) and supported by the Brazilian Federal Police.

References

[1] APWG, "Phishing activity trends report – 2nd quarter 2016." [Online]. Available: http://docs.apwg.org/reports/apwg_trends_report_q2_2016.pdf

[2] F. Merces, "The brazilian underground market." [Online]. Available: https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp-the-brazilian-underground-market.pdf

[3] Europol. Avalanche network dismantled in international cyber operation. [Online]. Available: https://www.europol.europa.eu/publications-documents/operation-avalanche-infographic

[4] S. L. Garfinkel, "Digital forensics research: The next 10 years," vol. 7, pp. S64–S73. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1742287610000368

[5] T. Berners-Lee, J. Hendler, O. Lassila, and others, "The semantic web," vol. 284, no. 5, pp. 28–37. [Online]. Available: http://ldc.usb.ve/~yudith/docencia/UCV/ScientificAmerican_FeatureArticle_TheSemanticWeb_May2001.pdf

[6] N. Shadbolt, T. Berners-Lee, and W. Hall, "The semantic web revisited," vol. 21, no. 3, pp. 96–101. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1637364

[7] S. Caltagirone, A. Pendergast, and C. Betz, "The diamond model of intrusion analysis." [Online]. Available: http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA586960

[8] C. Sauerwein, C. Sillaber, A. Mussmann, and R. Breu, "Threat intelligence sharing platforms: An exploratory study of software vendors and research perspectives," in *Towards Thought Leadership in Digital Transformation: 13. Internationale Tagung Wirtschaftsinformatik, WI 2017, St.Gallen, Switzerland, February 12-15, 2017*. [Online]. Available: https://wi2017.blob.core.windows.net/website/download/papers/WI2017-0188.pdf

[9] N. Marie and F. Gandon, "Survey of linked data based exploration systems," in *Proceedings of the 3rd International Conference on Intelligent Exploration of Semantic Data-Volume 1279*. CEUR-WS. org, pp. 66–77. [Online]. Available: http://dl.acm.org/citation.cfm?id=2877806

[10] P. Szekely, C. A. Knoblock, J. Slepicka, A. Philpot, A. Singh, C. Yin, D. Kapoor, P. Natarajan, D. Marcu, K. Knight, and others, "Building and using a knowledge graph to combat human trafficking," in *International Semantic Web Conference*. Springer, pp. 205–221. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-319-25010-6_12

[11] LDBC. Graph data management. [Online]. Available: http://ldbcouncil.org/public/why-graph

[12] Y. Nenov, R. Piro, B. Motik, I. Horrocks, Z. Wu, and J. Banerjee, "RDFox: A highly-scalable RDF store." [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-319-25010-6_1

[13] Ben Koehl and Ned Moran. The italian connection: An analysis of exploit supply chains and digital quartermasters. [Online]. Available: http://blog.shadowserver.org/2015/08/10/the-italian-connection-an-analysis-of-exploit-supply-chains-and-digital-quartermasters/

[14] R. Carvalho, M. Goldsmith, and S. Creese, "Applying semantic technologies to fight online banking fraud." IEEE, pp. 61–68. [Online]. Available: http://ieeexplore.ieee.org/document/7379724/

[15] Google. Introduction to structured data. [Online]. Available: https://developers.google.com/search/docs/guides/intro-structured-data