Dridex: analysis of the traffic and automatic generation of IOCs

Lauren Rudman
Security and Networks Research Group
Department of Computer Science
Rhodes University
Grahamstown, South Africa
Email: g11r0252@campus.ru.ac.za

Barry Irwin
Security and Networks Research Group
Department of Computer Science
Rhodes University
Grahamstown, South Africa
Email: b.irwin@ru.ac.za

***Abstract*—In this paper we present a framework that generates network Indicators of Compromise (IOC) automatically from a malware sample after dynamic runtime analysis. The framework addresses the limitations of manual Indicator of Compromise generation and utilises sandbox environment to perform the malware analysis in. We focus on the generation of network based IOCs from captured traffic files (PCAPs) generated by the dynamic malware analysis. The Cuckoo Sandbox environment is used for the analysis and the setup is described in detail. Accordingly, we discuss the concept of IOCs and the popular formats used as there is currently no standard. As an example of how the proof-of-concept framework can be used, we chose 100 Dridex malware samples and evaluated the traffic and showed what can be used for the generation of network-based IOCs. Results of our system confirm that we can create IOCs from dynamic malware analysis and avoid the legitimate background traffic originating from the sandbox system. We also briefly discuss the sharing of, and application of the generated IOCs and the number of systems that can be used to share them. Lastly we discuss how they can be useful in combating cyber threats.**

***Index Terms*—network security; malware; dridex; indicators of compromise**

I. Introduction

Many security breaches or intrusions on computer systems are not reported, never made public or even detected [1]. This allows attackers to have free reign of victims' computers, which may have negative effects on organisations, if their employees' computers are compromised. When an organisation finds out about a compromised system or threat and responds accordingly, the information gathered may be valuable to others who experience a similar threat. This makes the sharing of information relating to the detection and identification of threats on an organisation network an important step in dealing with cyber-attacks [2]. The more that is known about a threat, the easier it is to understand, track and counter it.

An Indicator of Compromise (IOC) is defined by Harrington [3] as "a piece of information that can be used to identify a potentially compromised system. It could include suspicious IP addresses, domain names, email addresses, file hashes or a file mutex. This paper focuses on the automated generation of network related IOCs using samples of the Dridex malware [4] strain as test input. The paper will also discuss the process and analysis used to find the information used in the generation of indicators.

The described system, takes as input a malware sample and outputs IOCs, using a collection of 100 Dridex malware binaries. The systems goal is to focuses on the IOC artefacts which can be observed on a network connection – particularly DNS, HTTP, TCP, UDP, ICMP, FTP, SSH and target addresses. These indicators will be created from the PCAP file containing network traffic from automated dynamic malware analysis.

The remainder of the paper is structured as follows. Background information is presented in Section II, followed by an introduction to the common descriptive languages used for constructing IOCs in Section III. The generation system and data collection environment are described in Section IV. An overview of the observed network traffic , and the processing thereof is in Section V, followed, in Section VI, by the process of the generation of IOC's from the captured traffic. Section VII concludes the research, while proposed future enhancements are presented in Section VII.

II. Background

There are two types of malware analysis, static and dynamic. Static analysis entails analysing the source code of the malware, never executing it. Dynamic analysis, on the other hand, is all about executing the malware and observing its behaviour on a system. Dynamic analysis is usually performed using a sandbox environment instead of an everyday computer. This is in case the malware potentially deletes files, changes the registry or even steals information. A sandbox is a restricted execution environment, that is run on a system, which allows the safe execution of malware without effecting the host system [5]. There are quite a few online malware analysis sites, such as Anubis, Comodo and Malwr, but these do not scale as they sometimes limit submission speed and the time results are given. It was decided to use a sandbox that runs locally on a system, instead of online.

The current system utilises the Cuckoo Sandbox as the analysis environment. Cuckoo is an open source automated malware analysis system that provides fast and complete analysis results [6]. It takes inputs such as Windows executables, DLL files, PDF documents, Microsoft Office Documents, URLs

978-1-5090-2473-8/16/$31.00 ©2016 IEEE

and PHP scripts. Some malware does have anti-virtualization techniques and does not execute in a sandbox virtual machine environment [7]. The framework in this paper, will therefore not be able to successfully generate network IOCs from them as they will not generate network traffic. However after successful dynamic analysis of a sample that does execute, Cuckoo generates a PCAP file of captured packets and a report which includes screen shots, static analysis results, dropped files, DNS and HTTP requests and a behaviour summary.

No accepted standard format for the IOCs exists yet. There are however a few systems that have their own formats, such as OpenIOC[1], Cyber Observable Expression (CybOX)[2] and Structured Threat Information Expression (STIX)[3]. These are discussed in Section III. The current systems typically require the manual input or tagging of of information to generate IOCs, which is not scalable and would take a lot of time to generate many IOCs. One of the recent developments in the sharing of cyber threats is OASIS Cyber Threat Intelligence (CTI)[4]. The OASIS CTI Technical Committee, which includes the U.S Department of Homeland Security and other organisations have come together to develop standards to enable the analysis and sharing of threats and treat information. They are intending for the cyber threat information to be shared among trusted partners and communities [8]. It would be useful to have a standard format, so that IOCs can be easily shared without having to convert between formats. This would also allow for a greater distribution of IOCs and help security teams in tackling cyber threats.

There are other new solutions which allow for the uploading of IOCs in multiple formats, such as the AlienVault Open Threat Exchange (OTX)[5]. OTX is an online platform for sharing cyber threat information about malware or fraud campaigns and more. Another solution is the Malware Information Sharing Platform (MISP)[6], which is a platform for sharing IOCs of targeted attacks.

When conducting a search for tools that automatically generate IOCs, a few simple scripts such as IOC_Creator[7] and IOCAware[8] were found. IOC_Creator generates OpenIOC formatted IOCs from unstructured data, although it lacks detail and is not comprehensive interms of network Indicators. IOCAware uses a Cuckoo report generated after a file is analysed and only generates an Indicator for an IP address contacted and no other network IOCs.

Dridex is a type of malware, with the primary goal of infecting computers, stealing credentials, and obtaining money from victims bank accounts [4]. It was first observed in the wild in November 2014 [9]. When the malware is installed, the computer becomes part of a botnet [4], which can be used to send phishing emails. In mid October 2015 many command-and-control servers used by the Dridex botnet were taken down by the Federal Bureau of Investigation (FBI) with the help of the National Crime Agency (NCA) [10]. However in late October, security researchers found signs that the botnet might still be functioning [11]. According to [12], Dridex was barely seen from 24 December 2015, but resumed its operations again in early January 2016. In February of 2016, it was found that part of the Dridex botnet may have been hacked as part of its distribution channel was changed by replacing malicious links with an installer for the Avira antivirus [13]. In March 2016, the Dridex botnet started to send SPAM emails with JavaScript attachments that eventually install Locky ransomware [14]. According to [15], in May 2016, the botnet was compromised again to distribute a "dummy file" instead of the Dridex binary.

The actual Dridex malware is spread through multiple types of spam email attacks with a Microsoft Word or Excel document attached, which includes a payload that downloads the malware [9]. Macros must be enabled in Microsoft Word for the payload to work [9]. Once installed, Dridex uses HTML injections to retrieve banking details [9] and can even steal user credentials through keystroke logging, form grabbing, stealing cookies and screenshots [4] [10] [12]. It is able to steal banking details of nearly 300 financial institutions of generally English speaking countries [16].

Dridex was chosen as it is a topical malware strain and since the framework can take any malware binary as an input, we thought that Dridex would be a useful demonstrative family. For the purposes of this research, 100 binaries identified to contain variants of the Dridex strain were analysed to show how the system operates. These were dated within the last twelve months.

III. STIX

STIX stands for Standard Threat Information Expression [8] and is used to describe information about cyber threats. It is an XML based format and was created to have a language that allows threat information to be easily stored, analysed and shared in a consistent manner [17]. We chosen STIX as the IOC language of choice because is able to represent a wide number of network level indicators. The level of detail of a STIX object can vary from one single property of an object to multiple properties of an object and even the logical (AND/OR) combination of objects [18]. The allowance for multiple indicators to be logically combined allows for the creation of IOCs to be flexible and have a high level of detail when needed.

STIX allows for the creation of many types of cyber threats, such as observables, indicators, incidents, exploit targets and more. We will be focusing on the creation of STIX Indicators in this paper. A STIX Indicator is made up of CybOX objects, which contain a number of cyber observables. A STIX Indicator gives the CybOX objects context by adding a title and description. A set of related STIX Indicators is grouped by a STIX Report and and lastly the Indicators and Reports are grouped using a STIX Package. CybOX is a language used

[1] http://openioc.org/
[2] https://cybox.mitre.org/
[3] https://stixproject.github.io/about/
[4] https://www.oasis-open.org/
[5] https://otx.alienvault.com/
[6] http://www.misp-project.org/
[7] https://github.com/tklane/openiocscripts/blob/master/ioc_creator.py
[8] https://goo.gl/ipBjZL

to describe ”events of stateful properties that are observable in a cyber domain” [17]. CybOX's data model uses an XML schema as does STIX.

It was decided to create our own reporting module that uses a filtered PCAP file to generate detailed network-based STIX Indicators. STIX is also one of the formats chosen by the OASIS CTI team to be a standard in the future of cyber threat sharing [8].

A. CybOX Objects

There are a number of different CybOX objects that can be used to create network related IOCs. The objects we have used for our system are listed below:

- Address: can be used to store addresses which include e-mail, MAC and IP addresses.
- Port: stores a port value.
- URI: stores a URI.
- Socket Address: stores an IPv4 address and Port.
- DNS Query: stores properties of a DNS query.
- HTTP Session: can store properties of a HTTP request and the response.
- Network Connection object, which is used to store information regarding any type of network connection.

The Port, URI, Socket Address and DNS Query objects were used as described in the above list. The Address object was used to store an IPv4 address only and the HTTP Session was used to store properties of and HTTP GET/POST request, without the response. The Network Connection object was used to represent TCP, UDP, ICMP, SSH and FTP connections.

IV. Data Collection

A. Cuckoo Sandbox

We implemented the framework on top of the Cuckoo Sandbox [5] which we used to execute and perform a first pass analysis on each Dridex malware sample. Cuckoo was chosen as the analysis tool because it is written in Python and is fully customizable and extendable [6]. It takes a suspicious file as an input and performs dynamic malware analysis on it, then generates reports, screen shots and a PCAP file. Our framework will focus on taking these reports and PCAP file to generate network related IOCs.

The latest version of Cuckoo was downloaded and installed on a Debian 8 server. In order for the sandbox to function, it needs a VM, a snapshot of the VM and a few variables in the configuration files changed to suit the setup. Cuckoo was configured to allow the use of VirtualBox as its virtual machine manager and each sample was set to run for 30 minutes each. We also had to configure more specific settings for VirtualBox in Cuckoo, such as making sure Cuckoo ran the VM in headless mode, the IP address of the virtual machine and the name of the virtual machine together with its snapshot.

In order for Cuckoo to capture network traffic, the configuration file, `auxiliary.conf` had to be modified to enable the packet sniffer, give the path to the local installation of the `tcpdump` utility and the name of the network adapter to capture traffic from. If these are wrong Cuckoo will not be able to capture traffic, or will record from the wrong adapter.

B. Virtual Machine Setup

As previously stated, VirtualBox was chosen to create and manage the virtual machines. Windows 7 Ultimate SP1 was installed on a VM to replicate an everyday user's computer. Windows XP may be used to test the in future work, but may not be too relevant these days as it is outdated and no longer supported my Microsoft[9].

The VM was setup to have 2GB of RAM together with 20 GB of storage. A few outdated versions of programs were installed such as Mozilla Firefox, Adobe, Microsoft Office 2007, Java, Google Chrome, Flash player, Opera, Adobe air, iTunes and Mozilla Thunderbird. These were chosen because they are common everyday programs and Microsoft Office was chosen because it is one of the programs Dridex uses to run its payload. However, the samples may be secondary binaries that do not even utilise this. To ensure minimal security, the firewall, Windows defender, and Windows updates were turned off along with not installing an antivirus.

The virtual machine was allowed access to the internet through a bridged adapter. The IP address of the VM and other network settings were statically set because Cuckoo needs to know the IP of the VM.

C. Extracting useful information

A toolchain of Python scripts were used to extract and analyse information generated by Cuckoo. First a script was created to filter out (as best as possible) non malware related traffic from each PCAP file. A non-malicious image file was submitted to the Cuckoo Sandbox five times to observe traffic created by the VM's snapshot. By using the baseline PCAPs, all of the IP addresses that the VM communicated with were extracted (excluding the pre-configured DNS server and the VM's IP address). These addresses were added to a list of clean IP addresses to filter from the PCAPs after the malicious files are analysed by Cuckoo.

Next the DNS queries and responses were extracted, which allowed us to create a list of clean domain names and the resolved IP addresses (if the domain was resolved). Taking a look at the domain names, a second filter list of clean domains was created. This list included domains such as 'microsoft.com', 'google.com', 'bing.com', 'windowsupdate.com', 'apple.com', 'sun.com' and others. These domain names, of course depend on the specific operating system and program versions installed on the VM. The IP addresses that the domains resolved to were added to the first IP filter list, if they were not previously added.

These two filter lists were used to create a `tshark` filter that reads the Cuckoo generated PCAP and creates a new PCAP. The filtered PCAP has packets of the type TCP, UDP, DNS, HTTP and ICMP and does not have packets to or from the clean IP addresses and also does not have the DNS requests

[9]http://windows.microsoft.com/en-us/windows/end-support-help

and responses for domains in the domain filter list. Because of dynamic IP addresses these two filter lists would have to be regenerated, before running a number of samples through Cuckoo.

The `tshark` filter was saved to a bash script so it could be utilised by a custom Cuckoo processing module that was developed. A Cuckoo processing module [10] is a python script that lets you analyse the raw output from Cuckoo and append some information to a global container that can be used by the reporting modules. After Cuckoo has executed a sample in a Virtual Machine the processing modules are called with the reporting modules following. The processing module that was created calls the `tshark` filter script after Cuckoo has executed a sample for 30 minutes and has generated a PCAP. The custom processing module worked well for the system and managed to filter out most unwanted baseline traffic. However, some samples would very rarely be found contacting local university IP addresses, such as IT management servers and printers, so these IPs were added manually to the clean filter list.

The next step was to work with these filtered PCAPs assuming they only contain malicious traffic and gather information that may be useful in the creation of IOCs. After all the enabled Cuckoo processing modules are finished executing, the reporting modules are run. A custom reporting module was created for the purpose of retrieving information from the filtered PCAP and using that information for the creation of STIX Indicators. Secondary filtering had to be implemented in the reporting module because of dynamic IP addressing where some Microsoft domains would resolve to a different address as found in the baseline PCAP files.

V. Network Traffic Overview

From the 100 Dridex samples that ran for 30 minutes, Cuckoo was able to execute 100% of them, with 50 samples generating network traffic. The traffic only added up to 31,45 MB. We are unsure as to why some of the samples did not show network activity. According to Rossouw et al. [19], this may be because the samples were invalid, or only active when there is user activity or they detected the sandbox and stopped working. We suspect it may also be because the malware needed more time to run (longer than 30 minutes) in order to generate traffic. Another possibility is that some of the malware was designed to activate only within a certain time period (which had expired). The take down of much of the Dridex infrastructure in October may also have played a role, in the reduced volumes of observed traffic. The following subsections will discuss the results found when analysing the filtered PCAPs which we assume contain malicious traffic. We will show information about TCP connections, DNS requests and responses, HTTP requests and any hard-coded IP addresses.

[10]http://docs.cuckoosandbox.org/en/latest/customization/processing/

A. DNS

Of the 50 samples that generated traffic within 30 minutes of running, 40 of the samples used the DNS protocol (port 53 UDP). Of the 40 samples, all of them used the pre-configured DNS server. If some of the samples had used their own resolver, the information could have been used in the creation of an IOC. Some malware strains use their own iterative/recursive resolvers to avoid leaving traces in logs or caches of preconfigured resolvers on a victim network [19].

Table I shows the 14 domain names that were requested by some of the 40 samples. Ten of the domains were resolved, which could mean that the other four are no longer in use or they could be blocked by the preconfigured resolver set by the university.

Looking at the TTL values of the ten resolved domains, seen in Table I the most popular domain, icanhazip.com has a TTL of 5 minutes. There are three very small values seen such as 3, 10 and 20 seconds, which are generally related to Content Delivery Networks (CDNs) [20]. In this case the three domains are related to online certificates, which could explain why the values are low. A TTL of zero, which is not seen in these results, can indicate the use of fast flux domains which are used to "provide flexibility among the command and control infrastructure of bots" [21]. However, many domains using a TTL of zero could be included as part of an extended IOC in future work.

The most popular domain icanhazip.com was requests by 33 of the 100 samples and 66% of all samples that generated network traffic. This site is non malicious and is used to determine the IP address of the host that loaded the page. According to [22], [23] and [24] malware authors use this domain and similar sites to obtain the IP of an infected computer, as part of the environmental determination used prior to contacting the Command and Control (C&C) node(s). This domain is often suggested to be used as an IOC according to [24]. In this case, we think that using this domain as an IOC is a good idea as it appeared many times from the samples. The other domain, api.ipify.org, is also a non malicious site and is similarly used to check the IP address of a client computer.

As seen in Table I, three distinct samples were seen using the following domains: th.symcb.com[11], th.symcd.com[12] and ocsp.thawte.com[13] [25] and two of the three samples also queried crl.thawte.com[14]. These domains are non malicious in themselves, but have been identified to be requested by malware in some cases as referenced above on VirusTotal. These domains can be used together to create on IOC to represent the three samples.

The domains malwagroup.org, thedirtydelicious.com, nerdmeetsgirl.com, tanhadhidown.ru, herssofhaprigh.ru, nohissandbo.ru were requested by the same sample and according to [26], these domains are used to download the

[11]https://www.virustotal.com/en/domain/th.symcb.com/information/
[12]https://www.virustotal.com/en/domain/th.symcd.com/information/
[13]https://www.virustotal.com/en/domain/ocsp.thawte.com/information/
[14]https://www.virustotal.com/en/domain/crl.thawte.com/information/

TABLE I: Domain names requested from 40 of the samples

Domain name	Number of samples	IPs resolved	DNS TTL
icanhazip.com	33	64.182.208.185 64.182.208.184	300
th.symcb.com	3	23.42.5.163	20
th.symcd.com	3	23.42.11.27	3
ocsp.thawte.com	3	23.42.11.27	10
crl.thawte.com	2	23.42.5.163	900
ho7rcj6wucosa5bu.tor2web.org	1	194.150.168.70 38.229.70.4 65.112.221.20	3600
api.ipify.org	1	50.17.192.14 107.20.229.58 54.243.252.101	60
malwagroup.org	1	182.50.130.67	3600
thedirtydelicious.com	1	184.168.27.45	600
nerdmeetsgirl.com	1	184.168.47.225	600
tanhadhidown.ru	1		
herssofhaprigh.ru	1		
nohissandbo.ru	1		
mcreport.org	1		

Pony and Dyre banking malware. These six domains can be combined used to create one IOC as they are only seen in one sample.

One sample used the ho7rcj6wucosa5bu.tor2web.org and api.ipify.org domains. As stated above, api.ipify.org, is used to retrieve the IP address of an infected host and ho7rcj6wucosa5bu.tor2web.org is a known malicious IP address [27] [28] and is using the Tor2web network gateway[15]. These two domains can be used to to create two indicators or one combined indicator. The last domain, mcreport.org, was not resolved and is only mentioned as the result of the analysis of a malicious file on [29]. This domain seems to be out of use at the moment, but can still be used to create an IOC.

Other than the domains themselves, the IP addresses that were resolved can also be used in the generation of IOCs. The resolved IP addresses may change, so this may not be as effective as an IOC because it may not be relevant for long.

B. HTTP

Only three samples from the 50 that generated network traffic utilised the HTTP protocol. This is a significantly small amount of of samples and does not give too much to work with in terms of IOC generation. Between the three samples, there were nine HTTP requests and four ”200 OK” replies and four ”404 Not Found” replies and one did not receive a reply.

When looking at some of the HTTP header fields, it was found that six out of the seven User-Agent fields did not correspond to the programs and operating system. For example one of the requests specified the Opera browser (not installed on the VM) and another specified that it was running on Ubuntu. The User-Agents are shown in Table II along with the ID of the sample. Sample 4 used three different User-Agents and so did Sample 48. It is interesting that every HTTP request had a different User-Agent value. Sample 99 was the only sample to use a correct User-Agent field. In [19], which also found that malware forges their own User-Agents, it was suggested that one sample can use different User-Agent because of the modular nature of malware. Each different module has its own way of forging an HTTP request.

Sample 4, mentioned in Table II, HTTP GET requested three URIs. These were 70.127.18.124/online.htm, 178.137.58.176/main/htm and 94.139.196.46/home.htm. The first URI did not receive a reply and the last two received a 404 Not Found reply. Sample 4 did not send any DNS requests, so these addresses were not resolved from a domain name.

Sample 48, sent three unique HTTP GET requests to IP addresses that had not been resolved from DNS requests. The three requests were 79.119.76.125/online.htm , 79.119.76.12/welcome.htm and 122.118.192.8/index.htm, the last of which brings up a login page to a router. There is most likely a compromised computer behind the router that is part of the Dridex botnet. Since the previously mentioned IP addresses had not been resolved from DNS requests it makes them a good property to add to part of an indicator. Each of the HTTP requests can be made into an HTTP Session CybOX object, that can be wrapped by a STIX Indicator.

Sample 99 sent three HTTP GET requests seen in Figure 1. These headers are very similar and show that the sample was attempting to download a file called ’k1.exe’ from the three domains. The headers have identical accept-language, accept-encoding, accept, and user-agent values in the fields and two of the headers have the same GET parameter. The malware most likely sends three requests for the same file (assuming it is the same file) in case one or more of the domains is down. In the case of the sample run, the ‘nerdmeetsgirl.com’ request (shown in Figure 1a) received a HTTP/200 response. For HTTP requests for ‘thedirtydelicious.com’ and ‘malwagroup.org’, shown in Figures 1b and 1c, response codes were received indicating that the files were no-longer present for download.

From the four different HTTP requests all of them used the GET request method. Rossow et al [19], analysed the network

[15]https://tor2web.org/

TABLE II: HTTP User Agents and sample IDs

User Agent	Sample ID
Mozilla/5.0 (Windows NT 5.1; rv:21.0) Gecko/20130401 Firefox/21.0	4
Mozilla/4.0 (compatible; MSIE 10.0; Windows NT 6.1; Trident/5.0)	4
Mozilla/5.0 (compatible; MSIE 9.0; AOL 9.7; AOLBuild 4343.19; Windows NT 6.1; WOW64; Trident/5.0; FunWebProducts)	4
Mozilla/5.0 (Windows NT 6.2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/28.0.1464.0 Safari/537.36	48
Opera/9.80 (Windows NT 6.1; U; es-ES) Presto/2.9.181 Version/12.00	48
Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:21.0) Gecko/20130331 Firefox/21.0	48
Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E; GWX:QUALIFIED)	99

output of multiple types of malware samples, they also found that GET request was the most popular request method over POST.

```
GET /wp-content/plugins/cached_data/k1.exe HTTP/1.0
accept-language: en-US
accept-encoding: identity, *;q=0
host: nerdmeetsgirl.com
accept: */*
user-agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows
NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR
2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729;
Media Center PC 6.0; .NET4.0C; .NET4.0E; GWX:QUALIFIED)
connection: close
```

(a) HTTP request to 'nerdmeetsgirl.com'

```
GET /wp-includes/simplepie/net/k1.exe HTTP/1.0
accept-language: en-US
accept-encoding: identity, *;q=0
host: thedirtydelicious.com
accept: */*
user-agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows
NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR
2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729;
Media Center PC 6.0; .NET4.0C; .NET4.0E; GWX:QUALIFIED)
connection: close
```

(b) HTTP request to 'thedirtydelicious.com'

```
GET /wp-content/plugins/cached_data/k1.exe HTTP/1.0
accept-language: en-US
accept-encoding: identity, *;q=0
host: malwagroup.org
accept: */*
user-agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows
NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR
2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729;
Media Center PC 6.0; .NET4.0C; .NET4.0E; GWX:QUALIFIED)
connection: close
```

(c) HTTP request to 'malwagroup.org'

Fig. 1: Three HTTP GET requests generated by one of the samples

C. Other Protocols

In terms of TCP requests, 42 out of the 50 samples that generated TCP traffic with 108 connections established and 869 connections failing. The samples did not show any malicious ICMP, UDP, FTP or SSH traffic.

VI. IOC Generation and Results

In this section we discuss how the IOCs were generated and show an example of an IOC that was created. As stated above, we used STIX for the creation of indicators and we generated seven types of indicators, ICMP, TCP, UDP, SSH, FTP, DNS and HTTP. Figure 2 shows the resulting flow of the system, with IOCs and a final product and Intrusion Detection System or firewall rules as a potential final product. As seen in the image and stated previously, a malware sample is submitted to the Cuckoo sandbox, which dynamically analyses its behaviour and records the network traffic to a PCAP file. The

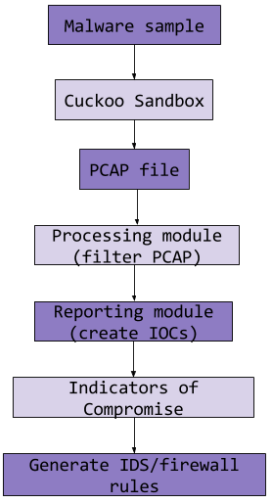

Fig. 2: IOC analysis and creation flow diagram

TABLE III: Cybox fields used for each IOC

IOC type	Network Properties
ICMP	IPv4 address, Type
TCP	IPv4 address, Port, TCP state
UDP	IPv4 address, Port
HTTP	Method, URI, Version, Host, Port, Accept, Accept language, Accept encoding, Authorization, Cache control, Connection, Cookie, Content length, Content type, Date, Proxy authorization
DNS	IPv4 address, Port, Domain name, Type
FTP	IPv4 address, Port, Request argument, Response argument
SSH	IPv4 address, Port, Public key

PCAP file is used by a custom processing module, which trims it, as described in Section IV-C. A custom reporting module then reads the new PCAP file and extracts useful values. Those values are then used to generates STIX Indicators for the specific protocols. The generation of the IOCs is discussed in more detail below.

A. IOC generation

In order to generate each IOC we need the correct values for each one. Table III shows the eight types of IOCs that were created and the network properties that were used to create them.

In order to generate the objects, we identified all the necessary packets for each malware sample and extracted the

```
−<cybox:Properties xsi:type="NetworkConnectionObj:NetworkConnectionObjectType">
    <NetworkConnectionObj:Layer3_Protocol>IPv4</NetworkConnectionObj:Layer3_Protocol>
    <NetworkConnectionObj:Layer4_Protocol>TCP</NetworkConnectionObj:Layer4_Protocol>
    <NetworkConnectionObj:Layer7_Protocol>HTTP</NetworkConnectionObj:Layer7_Protocol>
  −<NetworkConnectionObj:Layer7_Connections>
    −<NetworkConnectionObj:HTTP_Session xsi:type="HTTPSessionObj:HTTPSessionObjectType">
      −<HTTPSessionObj:HTTP_Request_Response>
        −<HTTPSessionObj:HTTP_Client_Request>
          −<HTTPSessionObj:HTTP_Request_Line>
              <HTTPSessionObj:HTTP_Method>GET</HTTPSessionObj:HTTP_Method>
              <HTTPSessionObj:Value>/wp-content/plugins/cached_data/k1.exe</HTTPSessionObj:Value>
              <HTTPSessionObj:Version>HTTP/1.0</HTTPSessionObj:Version>
            </HTTPSessionObj:HTTP_Request_Line>
          −<HTTPSessionObj:HTTP_Request_Header>
            −<HTTPSessionObj:Parsed_Header>
              <HTTPSessionObj:Accept>*/*</HTTPSessionObj:Accept>
              <HTTPSessionObj:Accept_Language>en-US</HTTPSessionObj:Accept_Language>
              <HTTPSessionObj:Accept_Encoding>identity, *;q=0</HTTPSessionObj:Accept_Encoding>
              <HTTPSessionObj:Connection>close</HTTPSessionObj:Connection>
            −<HTTPSessionObj:Host>
              −<HTTPSessionObj:Domain_Name xsi:type="URIObj:URIObjectType">
                  <URIObj:Value>nerdmeetsgirl.com</URIObj:Value>
                </HTTPSessionObj:Domain_Name>
              −<HTTPSessionObj:Port xsi:type="PortObj:PortObjectType">
                  <PortObj:Port_Value>80</PortObj:Port_Value>
```


Fig. 3: STIX Indicator for a HTTP GET Request

properties shown in Table III. Next the python-cybox[16] library was used to create CybOX objects out of the extracted values. These objects were then placed into STIX Indicators using the python-stix[17] library. The ID numbers of the STIX Indicators were placed into a STIX Report and the Indicators and the Report was finally wrapped by a STIX Package.

Part of a STIX Indicator that was created to represent a HTTP GET request is shown in Figure 3. A STIX Indicator can include a Title and Description which we used to describe the IOC (not shown in the Figure). The layout of a CybOX HTTP Session object is shown using the information from the before mentioned, 'nerdmeetsgirl.com' request shown in Figure 1a. The Host, URI, Port, Protocols and more are represented in Figure 3. CybOX objects have the useful trait of including the creation date of an IOC. This IOC layout is advantageous because it is simplistic as it does not contain too much information. It also contains meta data which is useful for sharing, so people can understand what the IOC is about.

Each malware sample ended up with an XML file containing all the indicators that were created, which is the final product of the system. These XML files can easily be shared manually using the AlienVault OTX. AlienVault requires a STIX file to be uploaded with the extension changed to '.ioc' from '.xml' before it is uploaded. MISP also allows for the upload and sharing of STIX data. MISP is also useful because it allows for the export of IOCs in different formats including Intrusion Detection Systems (IDS) rules, OpenIOC, plain text, Snort rules and Suricata rules.

[16]https://github.com/CybOXProject/python-cybox
[17]https://github.com/STIXProject/python-stix

VII. Conclusion

In this work, we presented a framework for the automatic generation of Indicators of Compromise from a malware sample. The Dridex malware strain was used as an example set of malware for analysis and the samples genearted PCAP files during dynamic analysis. An overview of the network traffic for DNS and HTTP protocols was shown, which resulted in some suspicious domain names, and HTTP request packets. The information gathered from these suspicious packets was used to generate the IOCs.

We can confirm that useful network-based IOCs can be generated from dynamic malware analysis while avoiding the legitimate background traffic originating from the sandbox system. An example of one of the IOCs can be seen in Section VI, and shows that the framework can create comprehensive STIX Indicators. Since the system can take any malware as an input, and uses PCAP files for the generation of Indicators, any malware that generates traffic during dynamic analysis in the sandbox used, will have a STIX file of IOCs generated.

The one downside of the system at the moment is that a baseline network traffic test has to be run every so often, but the method of filtering legitimate traffic from Cuckoo's PCAP file was very effective and lead us to create more accurate IOCs instead of creating IOCs from legitimate traffic, which would be troublesome. We believe that the framework, when expanded, will be a useful and scalable tool for the creation of all types of IOCs and could be used effectively in sharing cyber threats. This will help in combating cyber treats, by allowing the efficient generation of IOCs.

VIII. Future Work

Future research will be done with the aim of evaluating an optimal (and possibly flexible) means of sharing this IOC data in a way that it can be meaningfully utilised by others. This information will be used to expand the system to automatically share IOCs. Another useful expansion would be the creation of Intrusion Detection System and firewall rules from the STIX Indicators. This would allow for the data to be used as a defence mechanism against malware.

References

[1] D. W. Chris Johnson, Lee Badger, "Nist special publication 800-150 (draft) guide to cyber threat information sharing (draft)," October 2014. [Online]. Available: http://csrc.nist.gov/publications/drafts/800-150/sp800_150_draft.pdf

[2] J. A. L. Denise E. Zheng. (2015, March) Cyber threat information sharing recommendations for congress and the administration. CSIS. [Online]. Available: http://csis.org/files/publication/150310_cyberthreatinfosharing.pdf

[3] C. Harrington, "Sharing indicators of compromise: An overview of standards and formats," Conference Presentation, November 2013. [Online]. Available: https://www.rsaconference.com/writable/presentations/file_upload/dsp-w25a.pdf

[4] US-CERT. (2015, October) Alert (TA15-286A) Dridex P2P Malware. Online Article. US-CERT. [Accessed on: 23 October 2015]. [Online]. Available: https://www.us-cert.gov/ncas/alerts/TA15-286A

[5] D. Oktavianto and I. Muhardianto, *Cuckoo Malware Analysis*. Packt Publishing Ltd, 2013.

[6] A. Provataki and V. Katos, "Differential malware forensics," *Digital Investigation*, vol. 10, no. 4, pp. 311–322, 2013.

[7] X. Chen, J. Andersen, Z. M. Mao, M. Bailey, and J. Nazario, "Towards an understanding of anti-virtualization and anti-debugging behavior in modern malware," in *2008 IEEE International Conference on Dependable Systems and Networks With FTCS and DCC (DSN)*. IEEE, 2008, pp. 177–186.

[8] C. Geyer. (2015, July) Oasis advances automated cyber threat intelligence sharing with stix, taxii, cybox. Blog Post. OASIS. [Accessed on: 29 November 2015]. [Online]. Available: https://www.oasis-open.org/news/pr/oasis-advances-automated-cyber-threat-intelligence-sharing-with-stix-taxii-cybox

[9] M. Sanghavi. (2015, March) DRIDEX and how to overcome it. Blog Post. Symantec. [Accessed on: 23 October 2015]. [Online]. Available: http://www.symantec.com/connect/blogs/dridex-and-how-overcome-it

[10] Trend Micro. (2015, October) FBI, Security Vendors Partner for DRIDEX Takedown. Blog Post. Trend Micro. [Accessed on: 23 October 2015]. [Online]. Available: http://blog.trendmicro.com/trendlabs-security-intelligence/us-law-enforcement-takedown-dridex-botnet/

[11] D. Bisson. (2015, October) The Dridex botnet ain't done yet, say researchers. News Article. Graham Cluley. [Accessed on: 23 October 2015]. [Online]. Available: https://grahamcluley.com/2015/10/dridex-botnet-dead/

[12] D. OBrien, "Dridex: Tidal waves of spam pushing dangerous financial trojan," Symantec, White Paper, February 2016. [Online]. Available: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/dridex-financial-trojan.pdf

[13] L. Frink. (2016, February) Dridex botnet distributor now serves avira. Blog post. Avira. [Accessed on: 29 April 2016]. [Online]. Available: http://blog.avira.com/dridex_serves_avira/

[14] S. News. (2016, March) Dridex botnet spreading locky ransomware via javascript attachments. News Article. Security Week. [Accessed on: 29 April 2016]. [Online]. Available: http://www.securityweek.com/dridex-botnet-spreading-locky-ransomware-javascript-attachments

[15] Z. Zorz. (2016, May) Dridex botnet hacked, delivers dummy file. Online Article. Help Net Security. [Accessed on: 6 May 2016]. [Online]. Available: https://www.helpnetsecurity.com/2016/05/05/dridex-botnet-hacked/

[16] ——. (2016, February) Dridex botnet alive and well, now also spreading ransomware. Online Article. Help Net Security. [Accessed on: 29 April 2016]. [Online]. Available: https://www.helpnetsecurity.com/2016/02/17/dridex-botnet-alive-and-well-now-also-spreading-ransomware/

[17] MITRE. About STIX. The MITRE Corporation. [Online]. Available: http://stixproject.github.io/about/

[18] (2015) ObservableTypeCYBOX CORE SCHEMA. MITRE. [Acessed on: 1 November 2015]. [Online]. Available: http://stixproject.github.io/data-model/1.2/cybox/ObservableType/

[19] C. Rossow, C. J. Dietrich, H. Bos, L. Cavallaro, M. Van Steen, F. C. Freiling, and N. Pohlmann, "Sandnet: Network traffic analysis of malicious software," in *Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*. ACM, 2011, pp. 78–88.

[20] K. Fujiwara, A. Sato, and K. Yoshida, "Dns traffic analysiscdn and the world ipv6 launch," *Information and Media Technologies*, vol. 8, no. 3, pp. 833–842, 2013.

[21] T. Holz, C. Gorecki, K. Rieck, and F. C. Freiling, "Measuring and detecting fast-flux service networks." in *NDSS*, 2008.

[22] L. Teo, "Learning from the Dridex Malware - Adopting a Effective Strategy," SANS Institute, White Paper, October 2015. [Online]. Available: https://www.sans.org/reading-room/whitepapers/detection/learning-dridex-malware-adopting-effective-strategy-36397

[23] AplusWebMaster. (2015, July) 'Changed Identification Numbers', 'Hilton Hotel' SPAM, Zombie 'Orkut' Phish ... Forum Post. Spybot. [Accessed on: 1 Novemeber 2015]. [Online]. Available: https://forums.spybot.info/showthread.php?23632-SPAM-frauds-fakes-and-other-MALWARE-deliveries/page75

[24] B. Duncan. (2015) Upatre/Dyre - the daily grind of botnet-based malspam. Forum Post. SANS ISC InfoSec. [Accessed on: 1 November 2015]. [Online]. Available: https://isc.sans.edu/forums/diary/UpatreDyre%20the%20daily%20grind%20of%20botnetbased%20malspam/19657/

[25] [Online]. Available: https://www.virustotal.com/en/domain/ocsp.thawte.com/information/

[26] [Online]. Available: https://www.virustotal.com/en/file/facc9a5f02e8d18c9cbac9ee760ffa38b2854e5d5c89a529e368be8857bc55a9/analysis/

[27] B. Duncan. (2015, February) 2015-02-02 - malspam run pushes chanitor - subject: Logmein promo code - get 50MALWARE-TRAFFIC-ANALYSIS.NET. [Accessed on: 1 Novemeber 2015]. [Online]. Available: http://www.malware-traffic-analysis.net/2015/02/02/index.html

[28] [Online]. Available: https://www.virustotal.com/en/domain/ho7rcj6wucosa5bu.tor2web.org/information/

[29] [Online]. Available: https://malwr.com/analysis/ZjJkMmJkNTk3YmUyNDliZWFkMDNiZmQ3MmQ1YjJkZGU/