

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/327622961>

A sharing platform for Indicators of Compromise

Conference Paper · September 2016

CITATIONS
0

READS
3,066

2 authors, including:



Barry Irwin

Noroff University College

183 PUBLICATIONS 877 CITATIONS

SEE PROFILE

A sharing platform for Indicators of Compromise

Lauren Rudman

Security and Networks Research Group
Department of Computer Science
Rhodes University
Email: g11r0252@campus.ru.ac.za

Barry Irwin

Security and Networks Research Group
Department of Computer Science
Rhodes University
Email: b.irwin@ru.ac.za

Abstract—In this paper, we will describe the functionality of a proof of concept sharing platform for sharing cyber threat information. Information is shared in the Structured Threat Information eXpression (STIX) language displayed in HTML. We focus on the sharing of network Indicators of Compromise generated by malware samples. Our work is motivated by the need to provide a platform for exchanging comprehensive network level Indicators. Accordingly we demonstrate the functionality of our proof of concept project. We will discuss how to use some functions of the platform, such as sharing STIX Indicators, navigating around and downloading defense mechanisms. It will be shown how threat information can be converted into different formats to allow them to be used in firewall and Intrusion Detection System (IDS) rules. This is an extension to the sharing platform and makes the creation of network level defense mechanisms efficient. Two API functions of the platform will be successfully tested and are useful because this can allow for the bulk sharing and of threat information.

Keywords—network security; sharing platform; 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 [3].

This paper discusses a proof of concept sharing platform that was created to share network Indicators of Compromise (IOC) generated automatically from dynamic malware analysis by a previously created framework [3]. The information intended for sharing is in the STIX format and automatically generated after dynamic malware analysis has been performed.

Dynamic malware analysis is performed by executing malware samples and observing the behaviour on a system at runtime [4]. The system that generates Indicators utilises the Cuckoo Sandbox as the analysis environment. Cuckoo is an open source automated malware analysis system that provides fast and complete analysis results [5]. After analysis, Cuckoo generates a PCAP file of captured packets and a report which

includes screen shots, static analysis results, dropped files and more. The PCAP file is analysed to identify and generate IOCs in the STIX format.

This paper is structured as follows. In Section II, we will explore all the background information needed for this paper, which includes IOCs, STIX and cyber threat sharing. Section III will discuss related sharing platforms and Section IV following with the system design. The functionality of the system is tested in Section V and the conclusions are stated in Section VI. We will conclude the paper with Section VI that discusses future work that can be done to improve the framework.

II. BACKGROUND

An Indicator of Compromise is defined by Harrington [6] as “a piece of information that can be used to identify a potentially compromised system. It could include a suspicious IP address, an entire network packet, domain name, email address, file hash or a file mutex. There is currently no standard format for IOCs yet, however, a few that have developed their own formats, such as IODEF^{1,2} OpenIOC³, Cyber Observable Expression (CybOX)⁴ and Structured Threat Information Expression (STIX)⁵.

The OASIS CTI Technical Committee⁶, 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 threat information. They intend for cyber threat information to be shared among trusted partners and communities [7]. A standard format would be beneficial 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.

A. STIX

The Standard Threat Information Expression (STIX) is used to describe information about cyber threats. The goal of STIX is to have a language that allows information to be easily stored, analysed and shared in a consistent manner [8]. It

¹<https://www.ietf.org/rfc/rfc5070.txt>

²<https://tools.ietf.org/html/rfc5901>

³<http://openioc.org/>

⁴<https://cybox.mitre.org/>

⁵<https://stixproject.github.io/about/>

⁶<https://www.oasis-open.org/>

was developed by MITRE⁷ and was released in 2012, but has since been updated and improved. STIX has been adopted by many industry leaders and threat intelligence communities [9]. CybOX, STIX and TAXII are the most promising standards for threat intelligence sharing [9]. Finding a standard format to share information and having an online sharing platform are critical steps towards improving cyber defenses.

STIX not only allows for the creation of Indicators but also other contextual information such as observables, incidents, exploit targets, courses of action, campaigns and threat actors. STIX can be used in the analysis of cyber threats, specifying indicator patterns, managing cyber threat response patterns and sharing cyber threat information [10].

Since STIX uses CybOX objects, it also uses an XML schema. CybOX stands for Cyber Observable Expression and is now a community-driven effort, originally developed by MITRE, but is now in the hands of OASIS. It is designed for capture, specification, characterization and communication of cyber observables and is intended to be flexible enough to give a solution to all cyber security use cases [11] [12]. A cyber observable is a stateful property and a cyber indicator is made up of observables with relevant contextual information.

Each STIX Indicator is made up of CybOX objects, and a set of STIX Indicators can be grouped in a STIX Package [8]. A STIX Package can contain a combination of observables, indicators, exploit targets and more. STIX can define the relationship between constructs due to its structured nature [13].

B. Sharing Cyber Threats

The sharing of cyber threat information has been done previously by sharing CSV or PDF documents or with small groups of people sharing information amongst themselves [14]. Sharing information can build relationships between organizations and support the intelligence-driven security model [6]. Also, crowd sourcing is a good way of gathering information because no vendor has all the answers [15]. The reduction of duplication of effort would also be an advantage, as wasting time performing already performed analysis is not efficient [2]. Cyber defenses are enhanced by organizations by taking advantage of the knowledge, capabilities, and experience of a broader community, which allows for greater agility when defending against evolving threats [2]. Accurate shared information can help minimize the damage from an attack or even avoid major breaches [16].

III. RELATED WORK

There are a number of new sharing frameworks being developed for the purpose of sharing cyber threat information. These include TAXII, AlienVault Open Threat Exchange and MISP, which will be discussed below.

A. TAXII

The Trusted Automated Exchange of Indicator Information (TAXII) [13] was created by MITRE in 2012 and defines

⁷<http://www.mitre.org/>

services and message exchanges for sharing information [17]. It is the preferred exchange mechanism for STIX formatted threat information. The users of TAXII can choose what information they share and with who they share it with, which makes it flexible. TAXII supports different sharing models, like hub and spoke, source/subscriber and peer to peer.

Hub and spoke is a sharing model in which there is a main data hub where information is coordinated and shared through. The source/subscriber model has a single source and connected nodes which retrieve information from the source. They are not able to contribute and share information with the source node, just receive information. The peer to peer model allows two or more parties to share information directly with each other. There is no coordination of information exchange so one party may end up having much less shared information than another [18].

B. AlienVault Open Threat Exchange

AlienVault Open Threat Exchange (OTX)⁸ [16] allows for the uploading of IOCs in multiple formats. OTX is an online platform for sharing cyber threat information about malware, fraud campaigns and more. It uses crowdsourcing to find out about new malware, malicious IPs, vulnerabilities and exploits [15]. Many big security firms have not embraced the crowdsourcing method and like to keep their information in closed groups [15]. However, HP and Intel Security and others have partnered with AlienVault OTX.

OTX allows for the generation of threat information from the community, can handle collaborative research and provides automation of updating a company's threat data. In order to share an IOC on AlienVault, a pulse must be created. A pulse is a summary of a threat with the software it targets and the corresponding IOCs. The kinds of IOCs supported are IP address, domain name, hostname (subdomain), email, URL, URI, file hashes, Classless Inter-Domain Routing (CIDR) rules, file paths, MUTEX names and Common Vulnerability and Exposure (CVE) number.

After signing up for the OTX you have a profile and people can follow you, or you can follow other people and get notified if they create a pulse. You can also subscribe to pulses and get notified when they change [19]. A pulse can be created manually or automatically using their API.

Information on each indicator for the pulse can either be manually inputted or extracted from a source including a STIX report, OpenIOC file, blog post link, PDF threat reports or even text files. This makes AlienVault flexible as it takes many inputs. When viewing a pulse, the IOCs can be downloaded in four different formats: CSV, OpenIOC 1.0, OpenIOC 1.1 and STIX [19].

C. MISP

The Malware Information Sharing Platform (MISP)⁹ is a platform for sharing IOCs of targeted attacks. It can only be joined by cyber defence and governmental related constituent

⁸<https://otx.alienvault.com/>

⁹<http://www.misp-project.org/>

of the NATO member nations [20], unlike OTX that uses crowdsourcing. The importing of data can be done manually by using a template or by uploading text, OpenIOC or the sandbox results. This can be automated by using the API, PyMISP¹⁰ or ZeroMQ¹¹.

It allows for the export of data by generating Intrusion Detection System (IDS) rules like Snort and Suricata, and IOC formats like STIX, OpenIOC, plain text or CSV formatted data. MISP correlates relation between malware, events and attributes giving insight into probable advancements of a malware strain [20].

The system developed in this paper is different because it specifically focusses on comprehensive indicators on the network level and sharing and defense for them. Many of the systems mentioned above do not provide a mechanism to use the shared information (AlienVault OTX and TAXII) or do not go into enough detail with the indicators.

IV. SYSTEM DESIGN

The sharing platform was built in the form of a web application called IOC Xchange. It is a proof of concept platform and was developed to share STIX Packages with a focus on malware related indicators. The frameworks used in the design were Bootstrap¹², as the HTML, CSS, and JS framework and Flask¹³ as the Python Web Framework with SQLite for the database. Bootstrap was chosen because it has a comprehensive list of components, is highly flexible and uses a grid system for layouts. Flask was chosen because it is suited to smaller, simple web applications and it has numerous extensions that are easy to use. SQLite was chosen because of the nature of the web application and the fact that a production level database was not needed for the proof of concept task.

The Flask extensions that were used are Flask-SQLAlchemy¹⁴, Flask-Login¹⁵, Flask-WTF¹⁶ and Flask-Security¹⁷. Flask-SQLAlchemy adds support to Flask for SQLAlchemy¹⁸, which is a Python SQL tool kit and Object Relational Mapper that makes development easier and the database more secure. Flask-Login was used for user authentication and session management. Flask-WTF offers the integration of WTForms, which provides flexible form validation and rendering library. Flask-Security allows for the addition of common security mechanisms, such as session based authentication and password encryption. Another useful library used was werkzeug¹⁹, which has a function for securing file names that are about to be uploaded to a system.

The hub and spoke architecture was chosen and for the web application because it is advantageous to have a central place for the gathering of data and it allows for scaling.

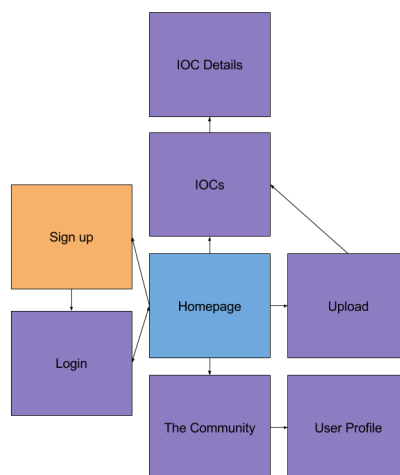


Fig. 1. Flow of the sharing platform

This is where the web application manages all the storage and information sharing of information from STIX Packages discussed in Section II-A. The hub does not simply broker information but it can also perform additional processing by converting a STIX Package (full of STIX Indicators) into certain formats of IDS or firewall rules. The structure of the site is discussed below along with the API functionality. Since important cyber threat information is to be shared, a hub-and-spoke architecture allows for a more comprehensive set of information than peer-to-peer.

The web application needed several use cases, which include signing up, logging in, uploading, downloading, viewing a list of shared files, viewing a file's indicators, viewing the site's users and viewing their profile. Figure 1 shows the basic navigation of the pages and does not include pressing the back button or the navigation bar on top of the pages except for the homepage. The concept of crowd sourcing and profiles was inspired by the AlienVault OTX and later discussed downloading of IDS and firewall rules was inspired by MISP. A user is first greeted by the Homepage and can either Login or Sign up. Users that are not logged in do not have access to the IOCs, Upload or Community page. Denying a user access to pages they are not authorised to visit is managed by the Flask-Security extension.

A. Sign up

When the Sign up option is selected and data is input into the username, email address, and password fields, the username and email are checked for duplicates in the database. If there are no duplicates, the password is hashed with a SHA 512 algorithm using the hashlib Python library. When Sign up is successful and the data has been saved to the database, the Login page is displayed as shown in Figure 1.

B. Login

For a user to sign in they must have a correct username and password combination. If either of these values are incorrect an error message is displayed. After a successful login, the Homepage is displayed and the navigation bar at the top is

¹⁰<https://github.com/CIRCL/PyMISP>

¹¹<http://zeromq.org/>

¹²<http://getbootstrap.com/>

¹³<http://flask.pocoo.org/>

¹⁴<http://flask-sqlalchemy.pocoo.org/2.1/>

¹⁵<https://flask-login.readthedocs.org/en/latest/>

¹⁶<https://flask-wtf.readthedocs.org/en/latest/>

¹⁷<https://pythonhosted.org/Flask-Security/>

¹⁸<http://www.sqlalchemy.org/>

¹⁹<http://werkzeug.pocoo.org/>

populated with options which include, Upload, IOCs, Users, Profile and Logout.

C. Viewing indicators

When the shared STIX Package files are viewed (the IOCs page from Figure 1) information about each uploaded file is used to populate a list on the page. Each row is selectable and redirects to a page that displays metadata about the STIX Package and the STIX Indicators from the files itself (the IOC Detail page from Figure 1). IOC Detail page from Figure 1 has four buttons that allow a user to download the information in different formats. These include the original STIX XML file or having the file converted to Snort rules, IPFW rules or iptables rules.

D. Upload

When Upload is selected, a page in which the suspicious file's name can be entered together with a description and the STIX Package file itself. It is recommended that the files name is the SHA 256 hash of the malware sample to make each file name unique. For a file to be uploaded, there must be no duplicates of it previously uploaded, it must be a .XML file and the file name is secured by using a function from the `werkzeug` Python library. Once a file is uploaded, it is checked to see if it contains a STIX package or not. If it is not an Error message is displayed and the file is deleted. If the file is validated a timestamp is taken, the malware name, description, and timestamp are saved to the database, along with the uploader's username and file path.

E. Viewing site users

When the Users page is loaded a selectable list of registered users is displayed. Each user has a simple profile page that consists of the user's username and a list of files they have uploaded.

F. API functionality

The web application had two API functions, upload, and download. API functions were added to make the upload and sharing scalable. For the upload function, four values are required, the path to the file being uploaded, the malware's name, the users API key and a description. HTTP POST methods are accepted and when the API is authenticated, and the file is only uploaded if it has the .XML extension and there are no duplicate files in the database. When the files have been uploaded the metadata is saved to the SQLite database and an upload successful message is sent as a reply to the HTTP POST request.

The conversion to IDS or firewall rules in a proof of concept extension to the sharing platform was implemented with an API function too. This is a way to make shared data useful for detecting and defending against threats on a network level. For the download API function, only HTTP POST requests with the three values (malware name, API key and convert type) are accepted. For a successful download, the API key must be authenticated and the given malware name validated. For each conversion type (Snort, IPFW, iptables and

None), a conversion script parses the selected file and creates a new file with rules of the chosen type. If the conversion type is None, the original XML file is downloaded.

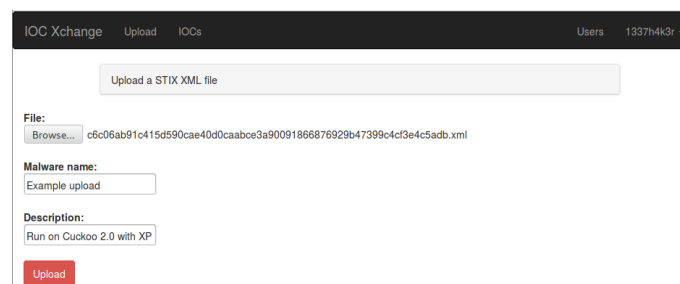
V. SYSTEM FUNCTIONALITY

The first page of the web application displayed the homepage of IOC Xchange, which has two options to choose from, Sign up or Login. A sign up page was loaded when the Sign up button was selected. A new user called 1337h4k3r was added into the username field along with an email address and password and the register button was pressed, saving the information to the database. A page with login fields was then loaded, where the username and password of the new user were entered and the Login button selected.

Since they were correct, a welcome message on the homepage was displayed. The top bar had the new options: Upload, IOCs, Users, Profile and Logout. When the Upload tab was selected, the page shown in Figure 2 was displayed. The STIX Packages uploaded in this test to the system were automatically generated by a previously developed system [3], that 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 were created from the PCAP files containing network traffic from automated dynamic malware analysis.

A STIX Package (XML file) was selected from the local system and the name and description were input into the fields. A sample of an Indicator shown in the STIX Package is shown in Figure 3. It is an Indicator for a HTTP GET request to the domain 'answeradvance.net'. The Upload button was pressed, which uploaded the file successfully. After the successful upload, the IOCs page was displayed (shown in Figure 4), with the top row displaying the file that was uploaded. The name, description, and timestamp of the upload are displayed for each file. The displayed files in Figure 4 were uploaded and automatically shared by the other users of the site.

The top row was clicked and the IOC Details page was loaded and displayed the information shown in Figure 5. This information shown included the malware name ('Example upload'), the uploader's username (shown on the green bar and links to the user's profile), the description and upload date. For each type of Indicator in the STIX Package, headings for each type were displayed (in this case it was HTTP GET or POST and IP addresses). There were six IP addresses and one HTTP GET request displayed.



The screenshot shows the 'IOC Xchange' web application interface. At the top, there is a navigation bar with 'Upload' and 'IOCs' tabs, and a user profile section for 'Users 1337h4k3r'. The main content area is titled 'Upload a STIX XML file'. It contains a 'File:' section with a 'Browse...' button and a text field containing a long alphanumeric string. Below this is a 'Malware name:' section with a text field containing 'Example upload'. The 'Description:' section has a text field containing 'Run on Cuckoo 2.0 with XP'. At the bottom of the form is a red 'Upload' button.

Fig. 2. Uploading a STIX file

```

<indicator:Title>HTTP request</indicator:Title>
<indicator:description>An indicator containing information about a HTTP request</indicator:description>
<indicator:observable id="example:Observable-8b2a61f9-b7f6-48a1-be4a-e838f3c8d9bc">
  <cybox:object id="example:NetworkConnection-22081bc7-6afe-42a4-89ce-efb3abe0e879">
    <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>/index.php</HTTPSessionObj:Value>
              </HTTPSessionObj:HTTP_Request_Line>
              <HTTPSessionObj:HTTP_Request_Header>
                <HTTPSessionObj:Parsed_Header>
                  <HTTPSessionObj:Accept>*/</HTTPSessionObj:Accept>
                  <HTTPSessionObj:Connection>close</HTTPSessionObj:Connection>
                  <HTTPSessionObj:Host>
                    <HTTPSessionObj:Domain_Name xsi:type="URIObj:URIObjType">
                      <URIObj:Value>answeradvance.net</URIObj:Value>
                    </HTTPSessionObj:Domain_Name>
                    <HTTPSessionObj:Port xsi:type="PortObj:PortObjType">
                      <PortObj:Port_Value>80</PortObj:Port_Value>
                    </HTTPSessionObj:Port>
                </HTTPSessionObj:HTTP_Request_Header>
              </HTTPSessionObj:HTTP_Request_Header>
            </HTTPSessionObj:HTTP_Client_Request>
          </HTTPSessionObj:HTTP_Request_Response>
        </NetworkConnectionObj:HTTP_Session>
      </NetworkConnectionObj:Layer7_Connections>
    </cybox:properties>
  </cybox:object>
</indicator:observable>
</indicator>

```

Fig. 3. Example of a STIX Indicator for an HTTP request

```

alert tcp $HOME_NET any -> any 80 (msg:"Malicious HTTP GET request detected
c6c06ab91c415d590cae40d0caabce3a90091866876929b47399c4cf3e4c5adb";
content:"answeradvance.net"; http_header; content:"/index.php";
http_uri; nocase; sid:234500;)

```

Fig. 6. Example of a Snort IDS rule for an HTTP request

TCP

Port	IP	Direction
21	188.165.230.79	out
50441	188.165.230.79	out
50676	188.165.230.79	out

Fig. 7. Example of how TCP Indicators are displayed

IOC Xchange Upload IOCs Users 1337h4k3r

STIX Indicators

Name	Description	Timestamp
Example upload	Run on Cuckoo 2.0 with XP	2016-04-14 22:00:12.556532
Example file	Cuckoo XP - file 5	2016-04-15 21:00:04.939088
Example file 2	Cuckoo XP - file 12	2016-04-15 21:00:47.820925
Example file 3	Cuckoo XP - file 27	2016-04-15 21:00:31.318111
Example file 4	Cuckoo XP - file 31	2016-04-15 21:00:29.221921
Example file 5	Cuckoo XP - file 54	2016-04-15 21:00:16.977591
Example file 6	Cuckoo XP - file 77	2016-04-15 21:00:02.794316
Example file 7 with FTP	Cuckoo XP - file 22	2016-04-15 21:00:48.894730
Example file 8 with FTP	Cuckoo XP - file 8	2016-04-15 21:00:31.076958
Example file 9 with FTP	Cuckoo XP - file 14	2016-04-15 21:00:21.463270

Fig. 4. Viewing a list of uploaded Indicator files

IOC Xchange Upload IOCs Users 1337h4k3r

Indicators for Example upload

Uploader: 1337h4k3r

Description:
Run on Cuckoo 2.0 with XP

Upload date:
2016-04-14 22:00:12.556532

HTTP GET or POST

Method	URI	Host
GET	/index.php	answeradvance.net

IP addresses

IP address

- 195.22.28.196
- 195.22.28.199
- 195.22.28.197
- 195.22.28.198
- 54.77.72.254
- 54.72.8.183

Download as:

Snort Rules
IPFW Rules
iptables Rules
STIX XML

Fig. 5. Viewing Indicators for a malware sample

At the bottom of the page shown in Figure 5 were four buttons labeled, Snort Rules, IPFW Rules, iptables Rules and STIX XML. Each button is for downloading the indicators in a different format. The Snort Rules button was selected and a dialogue box for saving or opening the file was displayed. An example of a Snort rule generated from the platform, for the Indicator shown in Figure 3 is shown in Figure 6. This was the same for the other buttons and the downloaded files contained the correct rules or XML as requested.

When the Users tab was selected, a page displaying registered users was shown. There were three registered users displayed and each row was selectable. When the '1337h4k3r' profile was selected from the list and redirected to the Profile page. This page consisted of a list of files the user had previously uploaded (in this case it was the example file uploaded earlier). Each row was selectable and lead to the page, from Figure 5. A user can get to their own profile by clicking their name in the top right corner and selecting 'Profile'. They can also logout through that tab.

A. Displaying Indicators

The sharing platform displays information about each indicator differently. For IP addresses, it shows rows of single IP addresses. For UDP and TCP indicators, were represented by three values, shown in Figure 7, port, IP address, and direction. If the direction is 'out', then the destination port and IP address are shown and visa versa.

For HTTP, the Method, URI, and Host are displayed. With a DNS request, the domain names are displayed. Four values were chosen to represent an FTP connection. These were the port, IP address, direction and description. Because CyBOX does not have properties to house FTP usernames, passwords or response codes, they were placed in the Indicator's description field. As seen in Figure 8, the description column includes a username and password, with information that confirms that the two values still work. There is also information on a file that was retrieved from the FTP server, called 'InstallFramework_150063j.exe'.

B. API

API Upload and Download

Port	IP	Direction	Description
21	188.165.230.79	in	Service ready for new user: ProFTPD 1.3.5 Server (Serveur FTP PC SOFT) [188.165.230.79]
21	188.165.230.79	out	Requested username: ██████████
21	188.165.230.79	out	Requested Password: ██████████
21	188.165.230.79	in	User logged in
21	188.165.230.79	in	Requested file action okay, completed.
21	188.165.230.79	out	Retrieve a copy of the file: InstallFramework_150063.exe

Fig. 8. Example of how FTP Indicators are displayed

Since the sharing platform has two simple API features, they were tested too. The upload feature was tested first using `curl`²⁰, which is a tool to transfer data to or from a server using certain protocols. The code shown in Listing 1 was used for the upload. There are four variables set, `file` (the path of the file to upload), `malware_name`, `api_key` and `description`. When the code was run, the return code showed a successful upload message. Looking at the web application, the uploaded file was displayed in on the Upload page. The Curl script can be used to bulk upload files if needed, by using a simple bash script.

Listing 1. Upload code

```
curl -v -F "file=@ioc.xml" -F
"malware_name=API_Example" -F
"api_key=65eab40b1bcd5c82c6d9e02abea5ed3"
-F "description=Uploaded_from_API"
http://localhost:5000/api/upload_ioc
```

The Download function was also tested using Curl and running the code in Listing 2. The three variables used are `malware_name`, `api_key` and `convert_type`. This code specifies that the file to be downloaded must be converted to Snort rules.

Listing 2. Download Code

```
curl -v -F "malware_name=Example_upload" -F
"api_key=65eab40b1bcd5c82c6d9e02abea5ed3"
-F "convert_type=snort"
http://localhost:5000/api/download_ioc
> Example_download.xml
```

The file downloaded successfully and contained Snort rules created from the STIX Indicators in the previously uploaded file. This Curl code can be used in a script that is used to bulk download files based on their name.

VI. CONCLUSION

In this paper, we described the functionality of the proof of concept sharing platform and discussed how each type of STIX Indicator would be represented in HTML. This included signing up to the application, logging in, uploading a STIX XML file, viewing the uploaded file and viewing other user's profiles. It was found that STIX data is easily represented using HTML and the layout was clear and simplistic. The downloading of Indicators was successful too and could be converted into three types of rules, Snort, iptables and IPFW. This is a useful part of the sharing platform as it makes the creation of defence mechanisms more efficient. Since the sharing platform had two API methods, those were tested as

²⁰<https://curl.haxx.se/docs/manpage.html>

well where a successful upload and download of files were performed.

Future work will include expanding the web application to allow more features such as improving a User's profile to include a reputation score. This will help with knowing if the information being shared is to be trusted. Allowing for the uploading or downloading of Indicators to use the OpenIOC format, because it is also a popular format like STIX.

ACKNOWLEDGMENT

This work was undertaken in the Distributed Multimedia CoE at Rhodes University, with financial support from Telkom SA, Tellabs, Easttel, Bright Ideas 39, THRIP and NRF SA (UID 75107). The authors acknowledge that opinions, findings and conclusions or recommendations expressed here are those of the authors and that none of the above mentioned sponsors accept liability whatsoever in this regard.

REFERENCES

- [1] C. Johnson, L. Badger, and D. Waltermire, *Guide to Cyber Threat Information Sharing (Draft)*. NIST, USA, October 2014. [Online]. Available: http://csrc.nist.gov/publications/drafts/800-150/sp800_150_draft.pdf
- [2] D. E. Zheng and J. A. Lewis. (2015) Cyber threat information sharing recommendations for congress and the administration. CSIS, USA.
- [3] L. Rudman and B. Irwin, "Towards a framework for automated generation of indicators of compromise through sandbox analysis of malware," Rhodes University, Tech. Rep., August 2015.
- [4] M. Sharif, V. Yegneswaran, H. Saidi, P. Porras, and W. Lee, "Eureka: A framework for enabling static malware analysis," in *Computer Security-ESORICS 2008*. Springer, 2008, pp. 481–500.
- [5] A. Provataki and V. Katos, "Differential malware forensics," *Digital Investigation*, vol. 10, no. 4, pp. 311–322, 2013.
- [6] C. Harrington, "Sharing Indicators of Compromise: An Overview of Standards and Formats," RSA Conference 2013, November 2013.
- [7] C. Geyer. (2015, July) OASIS Advances Automated Cyber Threat Intelligence Sharing with STIX, TAXII, CyBOX. Blog Post. OASIS.
- [8] S. Barnum, "Standardizing cyber threat intelligence information with the Structured Threat Information eXpression (STIX)," *MITRE Corporation*, p. 11, 2012.
- [9] F. Franssen, A. Smulders, and R. Kerkdijk, "Cyber security information exchange to gain insight into the effects of cyber threats and incidents," *e & i Elektrotechnik und Informationstechnik*, vol. 132, no. 2, pp. 106–112, 2015.
- [10] The Mitre Corporation. (2016) Use cases. [Online]. Available: <http://stixproject.github.io/usecases/>
- [11] ——. (2016) About CyBOX. [Online]. Available: <http://cyboxproject.github.io/about/>
- [12] N. ul-hassan Shirazi, A. Schaeffer-Filho, and D. Hutchison, "Attack pattern recognition through correlating cyber situational awareness in computer networks," in *Cyberpatterns: Unifying Design Patterns with Security and Attack Patterns*, C. Blackwell and H. Zhu, Eds. Springer, 2014, pp. 125–134.
- [13] G. Farnham and K. Leune, "Tools and Standards for Cyber Threat Intelligence Projects," *SANS Institute*, 2013.
- [14] M. Frazier. (2010, January) Combat the APT by Sharing Indicators of Compromise. Research Blog. FireEye.
- [15] R. Kirk, "Threat sharing—a neighbourhood watch for security practitioners," *Network Security*, vol. 2015, no. 12, pp. 5–7, 2015.
- [16] AlienVault, "Threat Intelligence Sharing & the Government's Role in It," 2015.
- [17] M. Davidson and C. Schmidt, "TAXII—Overview," 2014.
- [18] The MITRE Corporation. (2016) About TAXII. [Online]. Available: <http://taxiiproject.github.io/about/>
- [19] AlienVault. (2016) Welcome to AlienVault Open Threat Exchange! [Online]. Available: <https://www.alienvault.com/open-threat-exchange>
- [20] NCI Agency, "NCI Agency Malware Information Sharing Platform," 2013.

Lauren Rudman is in her final year of Computer Science Masters at Rhodes University researching in the field of network traffic and malware analysis.