electronics

MDPI

Article

A Comparative Analysis of Cyber-Threat Intelligence Sources, Formats and Languages

Andrew Ramsdale [1], Stavros Shiaeles [2,*] and Nicholas Kolokotronis [3]

[1] School of Computing, Electronics and Mathematics, Faculty of Science and Engineering, Plymouth University, Plymouth PL4 8AA, UK; andrew.ramsdale@postgrad.plymouth.ac.uk
[2] School of Computing, Faculty of Technology, University of Portsmouth, Portsmouth PO1 2UP, UK
[3] School of Economics and Technology, Faculty of Informatics and Telecommunications, University of Peloponnese, 22131 Tripolis, Greece; nkolok@uop.gr
* Correspondence: stavros.shiaeles@port.ac.uk

Received: 5 April 2020; Accepted: 13 May 2020; Published: 16 May 2020

Abstract: The sharing of cyber-threat intelligence is an essential part of multi-layered tools used to protect systems and organisations from various threats. Structured standards, such as STIX, TAXII and CybOX, were introduced to provide a common means of sharing cyber-threat intelligence and have been subsequently much-heralded as the de facto industry standards. In this paper, we investigate the landscape of the available formats and languages, along with the publicly available sources of threat feeds, how these are implemented and their suitability for providing rich cyber-threat intelligence. We also analyse at a sample of cyber-threat intelligence feeds, the type of data they provide and the issues found in aggregating and sharing the data. Moreover, the type of data supported by various formats and languages is correlated with the data needs for several use cases related to typical security operations. The main conclusions drawn by our analysis suggest that many of the standards have a poor level of adoption and implementation, with providers opting for custom or traditional simple formats.

Keywords: cyber-threat intelligence; threat exchange; vulnerability alerts; incident reporting; indicators of compromise; cyber-observables

1. Introduction

With the advent of the *Internet of things* (IoT), there has been an unprecedented increase of cyber-attacks, which have evolved and become more sophisticated. Adversaries now use a vast set of tools and tactics to attack their victims with their motivations ranging from intelligence collection to data destruction or financial gain. Understanding the attacker has become more complicated and even more important as this knowledge, if transformed into actionable information, can be used to adapt networks' defences in an automated manner to better protect the network against possible threats. *Cyber-threat intelligence* (CTI) focuses on the capabilities, motivations and goals of an adversary and how these could be achieved. Intelligence is the information and knowledge gained about an adversary through observation and analysis; intelligence is not just data, but the outcome of an analysis and must be actionable to meet the needs of current defensive systems that have to deal with and respond to cyber-attacks. Amongst others, examples of CTI include indicators (system artefacts or observables associated with an attack), security alerts, incident reports and threat intelligence, along with any other relevant information on recommended (or vulnerable) security tool configurations [1,2].

The efficient sharing of CTI is at the core of cyber-threat detection and prevention, as it allows building multi-layer automated tools with sophisticated and effective defensive capabilities that continuously analyse the vast amounts of the heterogeneous CTI related to attackers' *tactics*, *techniques*

and procedures (TTPs), indicators of ongoing incidents, etc. [3,4]. Given the numerous architectures, products and systems being used as sources of data for information sharing mechanisms, standardised and structured representations of CTI are required to allow a satisfying interoperability level across the various stakeholders [2]. Therefore, considerable efforts have been put during the last decade to standardise the data formats and exchange protocols related to CTI, including recent efforts aiming at promoting the *CTI for "things"* [5]; the initiative *making security measurable* (MSM) constitutes the most prominent effort toward improving CTI sharing among the various stakeholders [6].

The analysis carried out in this paper considers prominent representatives of CTI formats and languages that have been proposed and further studied in the literature, such as the *structured threat information expression* (STIX) [7], *trusted automated exchange of indicator information* (TAXII) [8,9] and *cyber observable expression* (CybOX) [10]. Among the paper's goals are to explore the capabilities of the available formats and languages and their capacity to convey various CTI types, to correlate their features with the degree to which they are used from the vast number of CTI sources and to correlate their capabilities with the needs of typical security use cases to which they are to be used. The above (and other) standardised formats and languages were believed to be the answer to the problem of not having common mechanisms for sharing cyber-threat intelligence. According to [11], STIX is the de facto standard for describing threat intelligence. In a literature review of STIX, TAXII and CybOX, several issues were identified that should be addressed to allow their wide adoption; these include:

- The headline standards of STIX, TAXII and CybOX have been superseded.
- The apparent acceptance and utilisation of the standards appeared lower than expected.
- Much of the body of knowledge found in the literature is outdated mainly due to the rapid change and development of the CTI formats and use.

To address the above issues and provide a state-of-the-art view of the CTI formats, use cases and implementations, the publicly available sources of CTI that share such data were researched along with any related formats and languages.

The organisation of the paper is as follows. We first provide a quick overview of the literature and the current state-of-the-art in Section 2, to have a knowledge base and an informed perspective on the findings and issues encountered. This is followed by Sections 3–5 that investigate CTI sources and formats and present the main result of our analysis. We conclude in Section 6.

2. Related Work

Much work has been carried out into investigating the sources, methods and platforms for sharing CTI. The science and technology used in practice, moves at a rapid pace, which results in literature becoming rapidly out of date with regards to the formats and languages currently in use. Irrespective of this, it still provides a valuable and relevant background to the research, with many of the findings still being valid regardless of the actual CTI format or platform used.

An exploratory study of software vendors and sharing perspectives was carried out in [11,12], where [12] focused more on the relationships between CTI sharing vendors and how these affect the sharing practices, whilst Sauerwein, et al. [11] targeted more on analysing threat intelligence sharing platforms and protocols. The applicable key findings are that there is no common definition of threat intelligence sharing platforms and that STIX is the de facto industry standard for describing threat intelligence. The authors of [11] carried out a broad literature review that identified 22 threat intelligence sharing platforms, comparing protocols and methods used for sharing CTI. According to Brown, et al. [13], there is an ever-increasing need to obtain greater amounts of threat intelligence, with the challenge of dealing with the large volumes of data effectively. A target-centric approach was proposed, where CTI is filtered given an understanding of the threat landscape and what the targets in an organisation are likely to be. The intelligence can be enriched from many sources to provide data that are relevant and applicable, while sharing is performed in a controlled manner, ensuring data privacy and security. The paper discusses standard and open formats for the sharing of

threat information and concludes that the adoption of STIX and TAXII by industry has led to many interoperable cyber information-sharing systems being developed. Given the vast quantity of CTI sources and feeds identified, the proposed target-centric approach merits further discussion. Another method to assess the relevancy of CTI sources according to the observables that they provide in allowing the early detection of cyber-attacks was proposed in [14]; the main idea relied on CTI content analysis and the "appearance-burst-disappearance" overall trend model. Likewise, content analysis techniques were also applied in [15], but with the different goal of introducing a new taxonomy of the CTI information conveyed by a data source: vulnerabilities, threats, countermeasures, attacks, risks and assets. In addition, this has been correlated with the type of the CTI source (i.e., blogs, forum, vendors, mailing lists, etc.) to gain some insight regarding the use of structured (or unstructured) CTI formats, the support of interfaces and APIs, the frequency of updating/sharing, the trustworthiness of the CTI and its originality. The latter is also considered in this paper, but for a much broader type of sources than those in [15], which are mostly limited (with few exceptions) to our class of external open-source intelligence sources that is next introduced.

The web-based research on cyber-threat intelligence that was carried out by Abu, et al. [16] concluded that the academic material available is limited due to the immaturity and instability in this relatively new field and therefore grey papers (as called therein) from various organisations and vendors must be the main information source. Along the same lines, Pala and Zhuang [17] reviewed research papers and approaches in cybersecurity information sharing and identified that techniques trying to optimally balance between cyber-investment/cyber-risk/privacy and CTI sharing (e.g., by using game theory) are gaining more attention. In contrast to the above approach, our research heavily relies on the direct inspection of the actual CTI obtained from various sources, with use of open-source tools whenever required and on the original documentation and articles by organisations and community sources. A survey focusing on technical aspects of threat intelligence was carried out in [18], where the types of intelligence, the benefits of sharing and the reasons for not sharing data were given. The authors also looked at the matter of quantity versus quality of CTI and the limitations in representing *indicators of compromise* (IoC), with a review of threat sharing formats and related platforms and their flexibility in sharing CTI. The paper adds to the data quantity issues found and highlights the need for quality and applicability of CTI. The analysis carried out in [18] assumes that CTI is classified into strategic, operational, tactical and technical, which differs from the one utilised in this paper and puts emphasis on CTI sharing platforms and their data enrichment, tools' integration and sharing capabilities.

On the other hand, Menges and Pernul [19] as well as Mavroeidis and Bromander [20] provided detailed analyses on the CTI sharing standards and incident reporting formats, along with certain associated threat taxonomies. More precisely, a different subset of the *malware attribute enumeration and characterisation* (MAEC), the *incident object description exchange format* (IODEF), the *vocabulary for event recording and incident sharing* (VERIS), the *extended abuse reporting format* (X-ARF), STIX and OpenIOC was considered in each paper with the analysis considering different features/criteria than those established herein. As an example, Menges and Pernul [19] was mostly concerned with general evaluation criteria (e.g., machine/human readability, interoperability, extensibility, aggregability, etc.), additional evaluation criteria (licensing, documentation and maintenance costs) and less with structural evaluation criteria (indicators, attacker, attack and defender), which are much more detailed in this paper and linked with typical security use cases. Although the latter type of criteria is rather the one that Mavroeidis and Bromander mostly considered [20], the particular criteria established (e.g., identity, motivation, goal, IoC, tool, target, strategy and TTP) allowed the comparative evaluation to be performed at a very high, non-technical level; the same criteria were used in [20] to evaluate threat taxonomies, such as CVE, CWE, CVSS, etc. Finally, Burger, et al. [21] as well as Asgarli and Burger [22] focused on segmented landscape of CTI standards and further investigated the use of CTI ontologies to allow for a better understanding of the security semantics and make inferences about ongoing cyber-security threats and incidents.

Although mainly concerned with STIX 1.x as a solution for sharing CTI, Serrano, et al. [23] highlighted several areas of importance in the context of CTI sharing. These include the legal and privacy implications in sharing CTI across borders and jurisdictions (also the focus in [24] and [25]), which have recently received great attention due to the *general data protection regulation* (GDPR), the requirement of a critical mass for CTI sharing sources that characterises its effectiveness, along with the belief that the main impediment to security data sharing is the lack of a suitable platform that addresses the issues of formats and legal boundaries for CTI data. Practices in sharing CTI were also studied in [26], where the results obtained from an online survey were used to classify potential barriers (and benefits) into areas such as *operational*, *organisational*, *economic* and *policy*; the quality and accuracy of CTI; the risk of privacy violation; the redundancy/relevancy of CTI; and the infrastructure costs were identified as the primary barriers. The lack of such a suitable platform was addressed in [27], where the *malware information sharing platform* (MISP) and the technical solutions used for sharing and synchronising threat information and taxonomies were described, as well as possible ways of extending the system's functionality. The MISP web interface and the use of the platform to present statistical information on the collected threats was discussed. Next, we further examine the MISP platform and the custom formats it uses for sharing CTI, along with the use of the *traffic light protocol* (TLP) that deals with the sharing of sensitive information.

In contrast to the aforementioned works, this paper's contributions are summarised as follows: (a) the research methodology relies on actual CTI obtained from a very large number of sources that are typically being used by today's security systems and products, instead of relying on previously published academic papers; (b) the types of sources considered are much broader, by considering internal, external and open sources to get representative results; (c) several tools/scripts were employed during the CTI collection process to allow for a comparison of the CTI against the original documentation and related technical/research papers; (d) the CTI formats and languages investigated herein are broader than those of the previous works, either by including recent ones gaining more attention (e.g., CVRF) or classical ones (e.g., DNSBL) that, although efficient in certain use cases, are usually not considered; and (e) the assessment criteria used are much more detailed and technical due to our goal in determining the extent at which typical security use cases can be supported by the existing CTI formats and languages.

3. CTI Sources

This section presents several CTI sources that have been examined, which are characterised as being *internal*, *externally sourced observables or feeds* and *externally open-source intelligence* [1,28,29]. It is important to highlight that the examination of CTIs was carried out by installing and using the tools provided from the manufactures, as well as by reading and analysing their documentation and various other online resources.

3.1. Internally Sourced

The CTI obtained from internal sources is comprised of observable events that have happened on an organisation's internal network and hosts (referred to as *threat indicators* in [30]). It can provide indicators about threats having breached the security perimeter, having broken the internal access control rules, having infected a system, or having attempted to get access to a restricted system. Statistical data provide a baseline of the normal behaviour so that any abnormality can be highlighted and investigated; possible sources are given in Table 1. More details about internal CTI sources are provided below.

System logs and events. Such information is widely available on devices and applications; it can be easily forwarded to a central facility using tools such as *Syslog* or *Windows event forwarding* (WEF). As only certain log messages and events apply to CTI, any central logging system, e.g., a *security incident and event management* (SIEM) system, should apply filters and rule-sets to extract CTI.

Table 1. Internal sources of cyber-threat intelligence.

CTI	Systems	Description
System logs and events	All systems	System activity, principally errors and security events
Network events	Network equipment, (switches, routers, firewalls)	devices connecting/disconnecting, ACL alert, login/failed login, etc.
Network utilisation and traffic profiles	Network equipment, (switches, routers, probes)	SNMP, NetFlow, RMON, etc. to Network management platform
Alerts from boundary devices	IDS/IPS, Firewall, WAF	Alerts/events collected and analysed by SIEM or vendor-specific management portal
AV, system alerts	Corporate AV software installed on host systems, (client and Server)	Corporate AV system alerts from host AV software
Human	All systems	Observed anomalies or events
Forensic	All systems	Artefacts and intelligence gathered after an event

Network events. Network devices such as routers, switches and firewalls, support *simple network management protocol* (SNMP), which can be used to send (in near real-time) event messages, known as *SNMP traps*, to a central server for processing. SNMP traps can be configured for a variety of CTI events in internal network (e.g., connections requested, login event occurring, etc.).

Network utilisation and traffic profiles. These may indicate abnormal behaviour, such as untrusted or excessive traffic from a client or between clients. Statistics are available in many forms, from simple counters in SNMP and *Remote MONitoring* (RMON) to detailed IP and protocol data from *NetFlow* and similar equipped switches and probes.

Boundary security devices. In addition to the above events, proprietary boundary security devices, such as *network intrusion prevention systems* (NIDS) and *web application firewalls* (WAF), may have their own application-specific management console that also feeds security events to a SIEM. An example of an alert generated by *Suricata* NIDS in JSON format is provided below in Listing 1.

Listing 1. Example of CTI (alert) obtained from Suricata.

```
{
“timestamp”: “2009-11-24T21:27:09.534255”,
“event_type”: “alert”,
“src_ip”: “192.168.2.7”,
“src_port”: 1041,
“dest_ip”: “X.X.250.50”,
“dest_port”: 80,
“proto”: “TCP”,
“alert”: {
“action”: “allowed”,
“gid”: 1,
“signature_id”:2001999,
“rev”: 9,
“signature”: “ET MALWARE BTGrab.com Spyware Downloading Ads”,
“category”: “A Network Trojan was detected”,
“severity”: 1
}
}
```

Anti-virus systems. Corporate anti-virus systems report malware events back to a central console, allowing a comprehensive coverage for the hosts within an organisation; as with boundary devices, this may also feed security events to a SIEM.

Human. An organisation's staff is often the quickest to recognise that something is wrong; the ability to rapidly spot and report events is something that can be achieved through user awareness and continuous professional security training programs.

Forensic. This CTI includes artefacts gathered from the investigation following a security incident and can be used to bolster security defences. The analysis of infected systems and log files can provide details about the *tactics, techniques and procedures* (TTPs) used during the attack.

3.2. Externally Sourced Observables

Locating, identifying and analysing the externally sourced observables or *feeds* formed the bulk of the research that was conducted in this work [30]. A selection of open and free to use sources of CTI was identified along with the formats and languages used, with an emphasis on sources using the STIX/TAXII standard. These community, open-source IoCs and observables typically consist of the observed malicious sources or data, e.g., IP address, domain, URL, file names and hashes. The principal use case is to explore this information to create rule sets for firewalls, network-based and host-based *intrusion detection and prevention systems* (IDPS), SIEM systems, etc., to block (or alert on seeing) the observable or a matching indicator.

To obtain samples of CTI data, the STIX sources having been identified to use the TAXII 1.x transport protocol were accessed with the *Cabby TAXII client* [31], while a simple Python script was written using the *CTI TAXII client* [32] for TAXII 2.x sources. Other simpler formats, such as text, CSV, JSON, etc., were accessed using a standard web browser or the Linux wget command to review the fields included. The CTI feeds and their respective formats were analysed and compared. Wherever available, the format documentation was downloaded from the source or authoring organisation to allow for a deep understanding of the format used and to contribute to the research and analysis of the formats and languages. Over 275 feeds were identified from the CTI sources, where the first 125 of these (all based on the STIX standard) were selected for analysis; the remaining >150 feeds identified were stored for future analysis. Table 2 shows the quantity and format of the 125 selected feeds obtained from each CTI source, where in case that a feed supports multiple formats, the most complex one was chosen. The formats and languages listed in Table 2 are further examined below (with certain indicative examples) and also discussed later in the paper.

Table 2. CTI Sources' Formats Used.

Source	Format								
	Text	CSV/RSS	JSON/XML	STIX 1.x	STIX 2.x	MISP	IDS	DNS	Total
abuse.ch	4	10	0	0	1	0	7	1	23
AbuseIPDB	0	0	1	0	0	0	0	0	1
Bambenek Consulting	0	1	0	0	0	0	0	0	1
blocklist.de	11	0	0	0	0	0	0	0	11
botvrij.eu	0	9	0	0	0	1	0	0	10
C1fApp	0	0	1	0	0	0	0	0	1
Censys	0	0	1	0	0	0	0	0	1
CINS Army (Sentinel)	1	0	0	1	0	0	0	0	2
cybercrime-tracker	1	2	0	0	0	0	0	0	3
Dshield (SANS)	3	3	1	0	0	0	0	0	7
FreeTAXII	0	0	0	0	11	0	0	0	11
Green Snow	1	0	0	0	0	0	0	0	1
HAIL A TAXII	0	0	0	9	0	0	0	0	9
Limo (Anomali)	0	0	0	0	11	0	0	0	11
Malc0de database	1	1	0	0	0	0	0	1	3
Malware Domain List	5	4	0	0	0	0	0	0	9
MISP (CIRCL)	0	0	0	0	0	1	0	0	1
PickUpSTIX (NC4/Soltra)	0	0	0	4	0	0	0	0	4
Spamhaus	3	1	0	0	0	0	0	6	10
TAXIIstand	0	0	0	1	0	0	0	0	1
ÜberTAXII	0	0	0	0	4	0	0	0	4
xavier.mertens.consulting	0	0	0	0	1	0	0	0	1

Among the above sources, abuse.ch makes several CTI feeds available through projects, such as *MalwareBazaar* and *URLhaus*, for sharing information about malware samples along with URLs being used for malware distribution, or the *SSL Blacklist* that provides information to detect malicious SSL connections and digital certificates used by botnet *command and control* (C&C) servers. The feeds provided by abuse.ch are comprehensive and are used and re-transmitted by several other providers. A typical example of the CTI shared (with the SHA1 fingerprints of the aforementioned certificates) in a CSV format is shown below in Listing 2.

Listing 2. Example of CTI obtained from abuse.ch.

```
##################################################################
# abuse.ch SSLBL SSL Certificate Blacklist (SHA1 Fingerprints)   #
# Last updated: 2020-05-03 06:46:48 UTC                          #
#                                                                #
# Terms Of Use: https://sslbl.abuse.ch/blacklist/                #
# For questions please contact sslbl [at] abuse.ch               #
##################################################################
#
# Listingdate,SHA1,Listingreason
2020-05-03 06:46:48,081cf50a56f59be9b1f9504858a225b80f233cb2,IcedID C&C
2020-05-02 07:48:30,19cf21e6326b6125b023c53df23b74060f4e786e,IcedID C&C
2020-05-02 07:41:15,e5d49e0b12012e40498cc991ae586b3ce05bf2f6,IcedID C&C
2020-05-01 18:01:48,8644711545fc8d1ba02fd4e4424290a06815c320,Adwind C&C
2020-05-01 17:59:19,20373e4d4d11ba0e839378737ee9fc49cb164bbd,ServHelper C&C
...
```

Another CTI provider is the service blocklist.de that takes reports from numerous active servers that use *fail2ban* and similar abuse blocking applications. The lists may be obtained through a direct download or via an API and are single-column text files that contain IP addresses; moreover, such information can be obtained by the DNS *real-time blackhole list* (RBL), which provides a simple DNS query response mechanism to determine the state of an individual IP address, as in the example that is shown in Listing 3.

Listing 3. Example of CTI obtained via blocklist.de with DNSRBL.

query:
host -t any 112.220.10.1.bl.blocklist.de
response:
112.220.10.1.bl.blocklist.de has address 127.0.0.21
112.220.10.1.bl.blocklist.de descriptive text "Infected System (Service: bruteforcelogin, Last-Attack: 1588509427), see http://www.blocklist.de/en/view.html?ip=1.10.220.112"

The list of IP addresses available for download by blocklist.de can also be protocol-specific (e.g., for the SSH, FTP, IMAP and SIP), targeting at bots, or other attacks such as the above brute-force attack against a web login; no metadata or other enrichment is provided. Similar information is also provided by *Spamhaus*, which is a well-known CTI source providing lists of IP address ranges that are involved in sending spam emails (SBL advisory), are compromised by malware and other exploits (XBL advisory), or belong in domains having low reputation (DBL advisory) amongst others. Further to the above, a subset of the SBL list is provided via the *don't route or peer* (DROP) list that can be used by firewalls and routers to drop malicious traffic; an example is given below in Listing 4.

Listing 4. Example of CTI obtained from Spamhaus.

```
; Spamhaus DROP List 2020/04/30 - (c) 2020 The Spamhaus Project
; https://www.spamhaus.org/drop/drop.txt
; Last-Modified: Thu, 30 Apr 2020 14:23:20 GMT
; Expires: Thu, 30 Apr 2020 15:41:23 GMT
1.10.16.0/20 ; SBL256894
1.19.0.0/16 ; SBL434604
1.32.128.0/18 ; SBL286275
2.56.255.0/24 ; SBL444288
2.59.151.0/24 ; SBL444170
...
```

On the other hand, the CTI provided from *Anomali Limo* is following the STIX 2.x standard and is delivered by means of the STAXX open source platform and Limo TAXII feed. The compliance with the STIX 2.x format is somewhat lazy, since many of the indicators' metadata are presented in the description field. Several collections are available, providing details about ransomware, cyber-crime, emerging threats (compromised or C&C servers), malware domains, phishing URLs, etc., but some of the feeds are re-transmissions of other sources (e.g., from abuse.ch).

3.3. External Open-Source Intelligence

For this type of CTI, we concentrated on *open sources of threat intelligence* (OSINT) from publicly available sources that contributed to building and understanding the threat landscape; although these tend to be more human (and more strategic, as highlighted in [30]) than machine-readable, they are often unstructured. Typical examples are: an announcement of a large data leak compromising user data that could be used to access other systems, in phishing attacks or in geopolitical tensions that may increase the risk of cyber-attack. Table 3 provides a brief list and description of the CTI sources that were identified.

Table 3. Externally sourced intelligence.

Source	Description
News feeds	News articles covering ongoing threats
Vulnerability	Alerts and advisories
Search automation	Using search technologies to find vulnerable systems: Google dorks, Shodan, etc.
Anti-virus vendors	Information, alerts, news feeds on malware activity and threats
Communications	Monitoring communication channels for intelligence: Slack, IRC, Twitter, etc.
Dark web	Intelligence available directly from the criminal underworld

A wealth of CTI information was available in the plentiful supply from news feeds, alerts, *antivirus* (AV) vendors, etc. In most of the cases, it was also available in RSS format, which is machine-readable; however, the news or alerts content typically contains a link redirecting to a free format web page that does not easily lend itself to automated consumption and understanding despite the considerable advances in the areas of *natural language processing* (NLP) and *artificial intelligence* (AI). Typical examples of such sources include CERT-EU, Schneier on security, Krebs on security, and SANS institute, amongst others.

Advisories and vulnerability alerts are sources having a standardised CTI format, in many cases using the *common vulnerabilities and exposures* (CVE) and *common weaknesses enumeration* (CWE), as well as the *common vulnerability reporting framework* (CVRF), which is next reviewed. This information is typically associated with a severity measure in the format of the *common vulnerability scoring system* (CVSS) and is also linked with the systems affected by the vulnerability through the *common platform enumeration* (CPE), therefore greatly helping in the dissemination of threat intelligence but with some limitations. Typical examples of such sources include the *national vulnerability database* (NVD), Cisco

security advisories, Microsoft security portal, Oracle security advisories, Red Hat security advisories, SecurityFocus, etc. In contrast to the previous type of external OSINT sources, these ones contain (or can readily generate) actionable security information. For example, NVD's data feeds, apart from the incorporation of the CVSS string (giving granular information about a vulnerability's preconditions and impact) also includes labels to any external references, such as *exploit*, *patch*, *mitigation*, *technical description* and *product*, which can direct tools automating the extraction of actionable information. An example from NVD's feed in JSON format is provided in Listing 5.

Listing 5. Example of CTI obtained from NVD (truncated/simplified for illustration purposes).

```
{
"cve" : {
"CVE_data_meta" : {
"ID" : "CVE-2020-0001"
},
"problemtype" : {
"value" : "CWE-269"
},
"references" : [ {
"url" : "https://source.android.com/security/bulletin/2020-01-01",
"tags" : [ "Vendor Advisory" ]
} ],
/* vulnerability description */
},
"configurations" : {
"cpe_match" : [ {
"vulnerable" : true,
"cpe23Uri" : "cpe:2.3:o:google:android:10.0:*:*:*:*:*:*:*"
} ]
},
"impact" : {
"cvssV3" : {
"version" : "3.1",
"vectorString" : "CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H",
"attackVector" : "LOCAL",
"attackComplexity" : "LOW",
"privilegesRequired" : "LOW",
"userInteraction" : "NONE",
"scope" : "UNCHANGED",
"confidentialityImpact" : "HIGH",
"integrityImpact" : "HIGH",
"availabilityImpact" : "HIGH",
"baseScore" : 7.8,
"baseSeverity" : "HIGH"
},
"exploitabilityScore" : 1.8,
"impactScore" : 5.9
}
}
```

The dark web search focused on finding intelligence, tools and services that are not available on the surface web. Our analysis was conducted using a TOR browser running on a disposable virtual machine to provide some insulation from malicious content. The speed and reliability of connections to .onion sites hampered and frustrated progress. Access to several forums was granted

by using anonymised email addresses but it was quite limited without first having gained trust in the community.

4. CTI Formats and Languages

Many CTI formats were identified from CTI sources and the literature; these were selected for further analysis based on their popularity in the literature or the source feeds. Where available, the original specifications, documents, schemas, etc., were examined by installing the right tools and applications. Samples of the formats were identified either from the CTI sources under investigation or the literature. The formats and languages have been classified into four main categories:

- Standards that have been specifically published for representing the CTI
- Custom application-specific or vendor-specific formats
- Commonly used standards that were not designed for representing the CTI
- Legacy formats, commonly referred to in the literature, but no longer being supported or used

A brief overview of the ones selected for further analysis is provided in the following subsections.

4.1. CTI Standards

STIX is one of our principal research subjects; it is a rich and extensive XML format that was first released in 2012 [33], with the minor revision 1.2 being released in 2015. The aim of STIX was to be a flexible and expressive language for representing cyber information. Where existing formats were used, e.g., MAEC [34], the objective was to *integrate rather than duplicate* them [7]. This provided a highly flexible format that ultimately led to its downfall, as the nested structures present in the XML documents became too complex and difficult to parse. STIX 1.2 was superseded by the 2.0 and in 2017 by 2.1 release. TAXII is the preferred, but not compulsory, transport mechanism for STIX [35]; there are different versions of TAXII for each release of STIX, which are not compatible with each other.

CybOX provides STIX 1.x the means to express cyber observables, events and other properties [10]. With the advent of STIX 2.1, CybOX has been integrated and is now part of the STIX standard. The principal differences between STIX 2.x and STIX 1.x are in the serialisation from XML to JSON that was designed to make the protocol more lightweight and much simpler for programmers [35]. The structure in STIX 2.x is flat rather than nested, with *STIX domain objects* (SDO) defined at the top level of the document to simplify parsing and storage; the relationship between the SDOs is accommodated by the introduction of a *STIX relationship object* (SRO) [36]. The CybOX objects have become *cyber observable* objects in STIX 2.x (under CybOX 3.0 release [37]) along with MAEC, therefore considerably decreasing complexity. Such changes were accompanied by a change in the management of the STIX project, which moved from MITRE to the OASIS CTI technical committee [38]. The MAEC 5.0 standard was designed for characterising malware using attributes such as behaviours, artefacts and relationships between malware samples [34,39]. This latest release was updated in line with STIX 2.x to maintain compatibility using the same cyber observable objects and JSON serialisation.

CVRF is another standard, whose format is machine-readable, aiming for the submission and distribution of vulnerability advisories and reports [40]. The utilisation of CVRF by MITRE's CVE repository, the principal registry of vulnerabilities and exposures, along with active support and feeds from vendors, such as Cisco, Oracle and Red Hat, are expected to help to establish CVRF as the de facto standard for the distribution of vulnerabilities and security advisories.

4.2. Application and Vendor Specific Formats

CESNET operates a large network infrastructure providing service to higher education and research establishments throughout Czech Republic; it created the *intrusion detection extensible alert* (IDEA) to overcome the complexities of other CTI formats [41]. IDEA aims at the sharing of CTI data that are varying in nature, thus it has to be flexible, extensible while staying simple. The MISP format is the native protocol for communication between the MISP platform instances [42]; this JSON format

is highly extensible and widely used by the MISP platform. The *collective intelligence framework* (CIF) is another widely used CTI aggregation and sharing platform that provides its JSON format for sharing CTI [43]. Finally, IDS/IPS rules are a long-lived CTI format that can be directly consumed by IDS/IPS applications such as Snort [44] and Suricata [45].

4.3. Commonly Used Standards

These formats were never designed or intended for use as a CTI sharing medium; despite this, the *DNS block list* (DNSBL), *DNS real-time black hole list* (DNSRBL) and Text/CSV are the oldest and most widely used formats identified. More precisely, DNSBL and DNSRBL are not downloadable lists of CTI host IPs [46]. Instead, they provide a rapid and efficient DNS-based request/response protocol to determine if an IP or domain exists on a blacklist or whitelist. It is likely one of the oldest methods used to get useful CTI information and is typically used by e-mail spam and malware filters.

Really simple syndication (RSS) is a lightweight XML format that is designed for the distribution of news items [47]. This format has been adopted by several sources for the distribution of CTI with detailed data available from a central repository. On the other hand, Text/CSV is the simplest and most widely used format of all the CTI source feeds sampled, either a single column text list of IPs or URLs (e.g., in the case of black lists), or as a rich, multi-IoC comma or tab-separated variables; they provide all the data in the most efficient and compact manner of any format.

4.4. Legacy Formats

The analysis of the final three CTI formats that we noted from the literature was curtailed due to the absence of current development, no active support or not being identified in any CTI source feeds examined.

Originally created by Mandiant Inc., under openioc.org, the OpenIOC format was designed to provide a common methodology and format for describing host-based or network-based indicators of compromise [48]. The legacy Mandiant resources and/or tools are available on GitHub, but there is currently no apparent activity [49]. The IODEF format was introduced by the Internet Engineering Task Force in RFC 5070 [50]; its current version 2 is described in RFC 7970. It is an XML-based format for exchanging CTI that is reported in the literature, but no evidence was identified about its current support, despite the second version's activity in 2016. Finally, the *open threat partner exchange* (OpenTPX) is an open-source and well-documented JSON format designed for sharing CTI [51]; no feeds were identified and there is no apparent evidence of updates since 2015.

5. Analysis

This section is mainly focused on externally sourced CTI feeds found in Sections 3 and 4. These sources are discussed after a brief analysis of the other CTI sources from our research.

5.1. Internally Sourced CTI

The CTI from internal sources appears to have a quite comprehensive coverage from the HIDS, SIEM and antivirus software provisions available; the majority of these were commercial offerings. It appears that the use of CTI, obtained from network activity such as network traffic flows, DNS requests, DHCP, ARP etc. (excluding NIDS), is not widely utilised and no further analysis was carried out to determine the effectiveness of current solutions on this type of CTI.

5.2. External Open Source Intelligence

The CTI examined from external open-source intelligence (OSINT) showed a very different context comparing to the machine-readable sources and formats. The analysis and application of this CTI is predominantly a manual process, converting this human-readable CTI into machine actionable formats where some of these were available, with some limitation, in machine readable formats such

as RSS and CVRF. Advances in natural language processing and AI offer significant opportunity in this area. The availability and structure of vulnerabilities and exposures through the CVE standard is well known and widely used [39] but the main drawback of this system is the limited applicability of the information available in a standard format. It should be noted that some vendors provided CVE feeds (e.g., [52–54]) that were quite comprehensive in what the applicable software versions were. The consistency and quality of the CTI that was identified from the dark web was found to be poor and mired in unsavoury content, mostly due to the lack of indexing and controller access to forums and credible resource. As much of the malicious activity originates from those who inhabit the dark web, it cannot be ignored as a potential source of intelligence.

5.3. CTI Source Feeds, Formats and Languages

The analysis carried out on the CTI source feeds revealed several different types of formats including single-column text feeds, multi-column, rich CSV feeds and more complex formats such as STIX and RSS. Many of these feeds, particularly those available in the more complex formats, were found to be retransmissions of simpler plain text feeds from other CTI sources. Examination of the feeds for evidence of originality (instead of being retransmissions) was not always possible. It is worth noting that some sources were found to be informative, giving details of how or where the CTI data were obtained and, in some cases, how agents could be downloaded, etc. A selection of sources, typically CSV or RSS feeds, provided web portal interfaces to search and examine the CTI data in greater depth. Figure 1 gives an overview of the originality for the threat feeds examined.

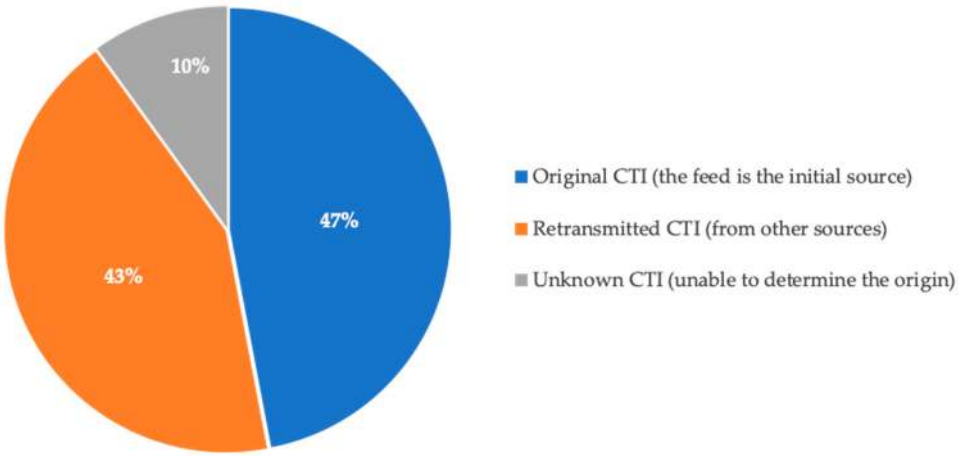

Figure 1. CTI source originality.

In the retransmission of CTI data, we found that some original source data can be lost or corrupted, which typically was attributed to the poor formatting, dates having been replaced so misrepresenting the freshness of the data, retransmitted or aggregated data appearing as a shadow sighting and giving false significance to the threat. We also observed a common practice of splitting the rich array of CTI types associated with a threat into separate, un-associated types, e.g., IP, domain, etc., diminishing the value of the original cohesive dataset.

In Figure 2, we illustrate the range of CTI types that were represented in the analysed CTI source feeds. IP addresses were the most common type, followed by the description of the threat or malware type and the URLs. From our analysis of the formats we knew that the rich intelligence source feeds could provide a more comprehensive dataset than that available from a simple block list. We compared how many of the sources using complex data formats provided rich CTI feeds. Here, we define *rich* as the CTI having more than two types represented in the feed, otherwise we consider it as being *sparse*. Our results are represented below.

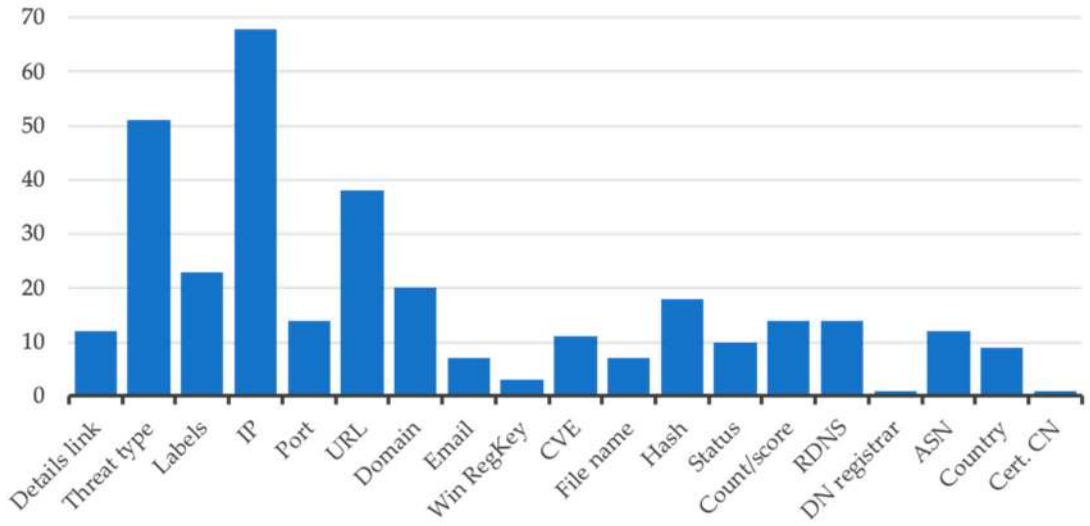

Figure 2. CTI types represented.

As highlighted in Figure 3, the capability of STIX to represent complex and rich CTI is somewhat underutilised, with most samples containing only sparse CTI. We carried out further analysis of the STIX 1.x format and compared the efficiency found in retransmitted CTI feeds. For example, a single entry <item> in the RSS *Malc0de database feed* [55] consumed 307 bytes. In contrast, the STIX 1.1 feed representing the indicators of same single entry from *PickUpSTIX* [56] consumed 18,153 bytes. Thus, it is clear that the used XML came with significant overhead and complexity.

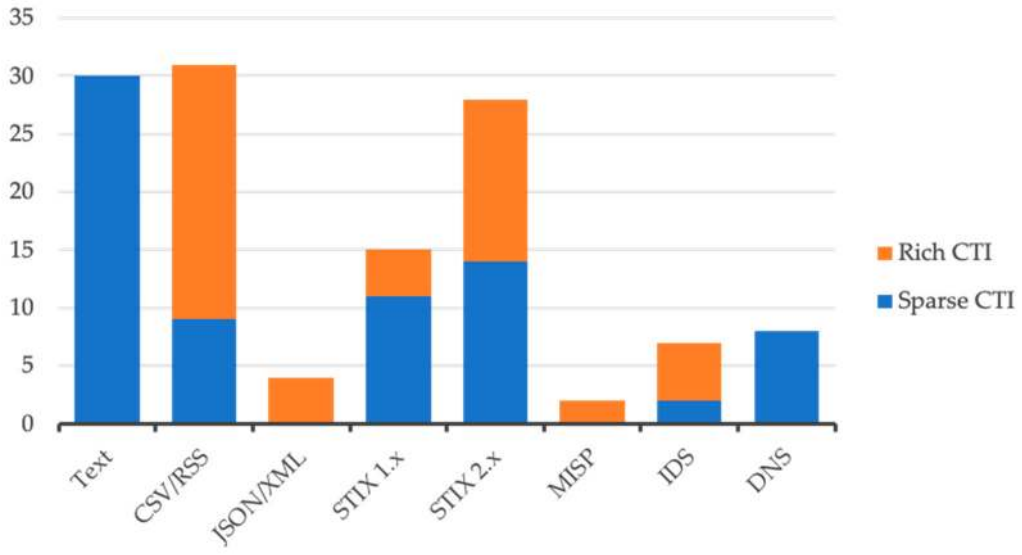

Figure 3. Rich vs. sparse CTI.

From the documentation of STIX 2.x, it is known that it can provide a more succinct representation than its 1.x predecessors. We still found that only half of the feeds analysed contained rich CTI data. A common approach taken was to put data in the description or title attributes rather than add additional observable objects or indicators to the feed. We refer to this as the *lazy implementation* of STIX format. We did note that the STIX feeds containing original content tend to be richer and much better implemented than those simply retransmitting data from other sources.

Complexity was one of the prime reasons for moving from STIX 1.x to 2.x, where the need for keeping things simple is also stated as a goal in MISP, CIF and IDEA formats. When analysing complex

CTI represented in MISP and STIX 2.x documentation, the strength of the formats to cross reference CTI comes to the fore. When we compare this to the implementations of simpler but still rich CTI, e.g., containing IPs, file names, file hashes and URLs, that are indicators for a strain of malware. However, without the need of TTPs, sequence of events, actor identities, etc., we see that the simpler formats can better express these.

To further examine how the use of the STIX versions varied between the providers, a common original source was chosen that was retransmitted by both STIX 1.x, 2.x sources. For our comparisons, the abuse.ch ransomware tracker feed was used [57]. The STIX 1.1 feed was sourced from PickUpSTIX [58], which contains better source metadata compared to with the *Anomali Limo* feed [59].

STIX 1.x and 2.x have similar capabilities to represent the data complexity as can be easily seen from Table 4. It was concluded that the Limo source appears to have a somewhat *lazy implementation* and further analysis was conducted on the STIX 2.x sources to reveal if this is a common practice or not. For this, sixteen samples of TAXII collections were examined from three STIX 2.x source providers to compare how well they utilised the capabilities of the format and structure. The observed data or indicator objects were analysed for containing multiple IoC types in the file (e.g., IP, URL, MD5, etc.); multiple IoC in an either observed-data.objects or indicator.pattern objects; and examples of rich content, e.g., multiple IoC, related objects, etc.

Table 4. STIX, PickUpSTIX and Limo metadata comparison.

Data	PickUpSTIX (STIX 1.1)	Limo (STIX 2.x)
Terms of use	Included per STIX package	-
TLP	TLP White per STIX package	Common Marking definition TLP: Green
Producer Description	Aggregator of Malware Sites	-
Producer Role	Aggregator	-
Producer Timestamp	Timestamp	-
Producer feed URL	Ransomware feed URL	-
Indicator Title	Description and IoC URL	Threat Stream ID, type, state, org, source
Observable Title	IoC	IoC
Observable condition/pattern	IoC	IoC
Observables per Indicator or related group of indicators	Multiple: IP, ASN, file, hash, URL, etc.	Single IoC type per feed (IP, Domain)
Labels	Unclassified (Public) marking	Malicious activity Threatstream severity Threatstream confidence

Our results in Table 5 indicate that the analysed STIX 2.x samples gained only a little advantage from using the STIX format.

Table 5. STIX 2.x Feature Use.

Source	Multiple IoC Types Per File	Multiple IoC Types Per Indicator	Rich CTI/Indicators
Limo (Anomali)	1 of 9 collections	None	1 of 9 collections
xavier.mertens.consulting	None	None	None
ÜberTAXII	5 of 6 collections	4 of 6 collections	3 of 6 collections

From the CTI samples identified in our research, many simpler formats such as CSV and RSS had grouped indicators for a given threat with a common label or tag. STIX uses a combination

of observed data structures, indicator patterns and relationships. The STIX *bundle* object is only a container and does not imply any relationship between the objects contained therein; a *relationship* object is required to represent this, using the UUIDs of the related objects, along with its own UUID, markings, originator, etc. This can result in a complex document to represent a collection of CTI related to a single threat. This is an area in which the MISP format excels; the sharing of data between MISP instances is threat-centric. Here, a single *event* file contains all the CTI for a threat; UUIDs are used to cross-reference and form relationships the same as STIX; and the attribute array structures are similar to STIX observables. However, the relationships are embedded with no additional objects or complexity required.

We find a similar situation with STIX markings when compared to MISP tags. In STIX, a marking definition is typically a global object and the indicator objects reference these directly. MISP, which has a rich tag and taxonomy implementation, embeds the tag objects directly. This is very simple but creates the potential for inconsistency between versions of the same tag. As the name suggests, *universally unique identifiers* (UUID) RFC4122 provide unique IDs [60]. Several of the CTI formats examined use these to identify and reference CTI data, markings, relationships and more.

CVRF was found to be a rich format that can meet the need to share vulnerabilities; the addition of a revision history within the vulnerability structure would provide a clearer versioning of individual vulnerabilities. The biggest weaknesses observed was the limited compliance from major influencers and the dilution of the format with multiple, equally suitable alternatives and insufficient target data and remediations in a consistent and standardised manner. As noted above, there is good vendor support for identifying the applicability of a vulnerability and remediations.

MAEC has good support from Sandbox providers, although there is a dilution from the use of older versions and the widespread availability of platform-specific API formats. MAEC 5.0 leverages STIX 2.x cyber observables, types and languages, but there is no evidence of reciprocal support with no facility to reference or include MAEC content in STIX 2.x, as was available in STIX 1.x.

The platform or API custom formats such as MISP, IDEA, CIF, etc., had an enthusiastic use of the formats, and they were found to be better suited to their given use case and able to represent the CTI observables and indicators in a succinct yet comprehensive manner. The MISP format has grown from real-world use; the MISP project sites over 6K installations of the MISP platform, illustrating the wide support in both community and government organisations.

In Tables 6–8, the various CTI formats and languages that were researched and analysed are compared to determine how well they are able to convey CTI for different use cases. The criteria are applied to the representation of a single, complete cyber observable, where a single observable can be an event, indicator or similar such single entry, line or item in a list or structure. For example, the CTI indicating the presence of a malware compromise, source of the infection (IP, domain, URL, file, hash, etc.), the destination or target (IP, hostname, domain, vulnerability, etc.) and threat details (malware name, family, type, etc.). The test applies to dedicated fields or columns that are machine readable and unambiguous, inclusion of CTI data fields in general purpose descriptions is ignored.

Table 6. Format and Languages, Assessment Criteria.

Criteria/Feature	Assessment Criteria	Notes
Blocklist	Provides effective and simple representation of a blocklist. This can be an IP/domain list, or a go/no–go request/response mechanism	-
IP v4 Address	An IP v4 address or network and mask e.g., CDIR format or with netmask	▲ for supporting IP ranges/multiple IP's
IP v6 Address	An IP v6 address or network and mask e.g., CDIR format or with netmask	▲ for supporting IP ranges/multiple IP's
Hardware/product	Hardware or product information, system make, model, MAC address, etc. Expect 2	▲ for more, and ?▽ for less
Email address	Represent an email address, typically a known malware 'from' address or C&C address	▲ for multiple addresses
Hostname	The hostname	-
URL/URI	URL	-
Domain	Domain (FQDN)	▲ for details, RDNS
Attacker/Target	Specify the data refers to the attacker or network source and/or the Target or destination	▲ for source and destination details
Vulnerability	Details of a vulnerability, e.g., CVE or reference to similar source, OS/SW vendor etc.	-
Malware or Threat Type	Provide the name of the malware or threat	▲ for details of role, family, type
Ransomware	In addition to malware, specific ID as ransomware	▲ for details on virus total, etc.)
File	Details of a malicious file, e.g., file name, source path, destination path, file hash, alternate names, virus total, etc. Expect 2 or 3	▲ for more, and ▽ for less
Detailed system IoCs	Details of observable artefacts or indicators of system compromise, e.g., Windows registry values, files, executables, libraries infected, hashes. Expect 2 or 3	▲ for more, and ▽ for less
DDoS	Identify the CTI as belonging to DDoS, or indicating DDoS. May include: C&C server, botnet description, DDoS type, IP lists, ASN, IP/Port and rate or counts, Expect 2	▲ for more, and ▽ for less
Compromised host, RAT	Identify CTI as indicating or observed compromised host, Remote Access Trojan, or similar 'owned' host, network, website, etc. Not a bot net. Expect a host identifier (IP, URL) and threat/compromise.	-
Botnet	Identify the CTI as belonging to a botnet, should include botnet name along with indicators/observables, C&C servers, bots, target device/OS, etc. Expect 2	▲ for more, and ▽ for less
Spam	Identifies CTI as being concerned with Unsolicited Commercial Email, may include domains, IP, email addresses, subject lines, etc.	-
Phishing	Identifies CTI as being concerned with Unsolicited malicious email aimed at compromising, or some malicious act. May include domains, IP, email addresses, subject lines, file detail	-
Software	Details of software, operating system, version, etc. Expect 2 (e.g., OS and version)	▲ for more, and ▽ for less
Time Stamps	Timestamps such as: data produced, first seen, last seen, window, etc. Expect 2 or 3	▲ for more, and ▽ for less
CTI Source	Accreditation of the CTI source	▲ for references, or the collector/agent
Complexity	A measure of not being over complex, effectively doing what it says on the tin without being over packaged	▲ if succinct, and ▽ if over complex
Rich CTI data	The Format or language can represent 10 CTI attributes	▲ for more, and ▽ for 8–9
Patterns	Patterns to match observed data, e.g., LIKE text, Regular Expressions, Hex bytes, etc. Expect 2	▲ for more, and ▽ for less
Identity	Identify a person, user, threat actor or organization. Can include name, location, function, etc. Expect name and function/type	▲ for more, and ▽ for less
Course of Action	What to do, remediation, etc. to protect from a threat or fix a vulnerability, expect text and references	▲ for more, and ▽ for less
Versioning	The means to know that the CTI has been updated	-
Author	Organization, group or person who created this CTI, ref to ID is acceptable.	-
Confidence, count	Confidence, rating or simple count of observations	-
Markings	TLP or similar security of distribution marking, Tags, etc.	-
Artefact	Contain encoded CTI artefact data or link to data.	

Table 7. Formats and languages, use case and features.

Typical Use Case							Criteria/Feature	Formats and Languages											
Email Blocklist	Spam/Email Filter	Firewall/Router ACL	NIDS	HIDS/SIEM	Malware Analysis	Human, SOC, DB		STIX 1.x	STIX 2.x	MAEC	CVRF	IDEA	CIF (Platform API)	MISP (Platform API)	Snort/Suricata Rules	DNSBL	RSS	Text CSV	Text List
▲	▲	▲	▽	▽		▽	Blocklist					▽			▽	▲	▽	✖	▲
✖	✖	✖	✖	✖	✖	✖	IP v4 Address	✖	✖	✖		▲	▲	▲	▲	✖	✖	✖	✖
✖	✖	✖	✖	✖	✖	✖	IP v6 Address	✖	✖	✖		▲	▲	✖	▲	✖	✖	✖	✖
			▽	✖	✖	✖	Hardware	▽	▽	▽	▲								
✖	✖		✖	✖	✖	✖	Email address	✖	✖	✖		✖	✖	▲		✖	✖	✖	✖
			✖	✖	✖	✖	Hostname	✖	✖	✖		▲		✖	✖				
	✖		✖	✖	✖	✖	URL/URI	✖	✖	✖		▲		✖	✖	✖	✖	✖	✖
✖	✖		✖	✖	✖	✖	Domain	✖	✖	✖		▲	▲	✖	✖	✖	✖	✖	✖
			✖	✖		✖	Attacker/Target	✖	✖	✖		▲		✖	✖				
			✖	✖	▲	✖	Vulnerability	✖	▽	▲	▲	✖		▲					
	▽		✖	✖	▲	✖	Malware/Threat Type	✖	▽	▲	▲	▲		✖			✖	✖	
	▽		✖	✖	▲	✖	Ransomware	✖		▲		✖		✖			✖	✖	
	▽		✖	✖	▲	✖	File	▲	▲	▲		▲		▲	✖		✖	✖	
				▲	▲	✖	Detailed system IoCs	▲	▲	▲		▽		▲	✖				
		✖	✖			✖	DDoS			✖		✖		▲	✖		▽	▽	▽
			✖	✖	✖	✖	Compromised host			▽		▲		▽			✖	✖	▽
	▽		✖	✖	✖	✖	Botnet			▲		✖					▽	▽	▽
	▲					✖	Spam			✖		✖				✖	✖	✖	✖
	▲		✖	✖	✖	✖	Phishing			✖		✖				▽	▽	▽	▽
				▲	✖	✖	Software	✖	✖	✖	▲								
				✖	✖	✖	Time Stamps	✖	✖	✖	▲	▲		✖			✖	▽	▽
						✖	CTI Source	✖	✖	✖	▲	▲		✖			▽	▽	
▲	▲	▲	▲	✖	✖	✖	Complexity	▽	✖	✖	✖	▲		✖	✖	▲	▲	▲	▲
	▽		✖	✖	▲	▲	Rich CTI data	▲	▲	▲	▲	▲		▲	✖		▽	✖	
	✖	▽	▲	▲	✖		Patterns	✖	▲	▲				✖	▲				
	✖			✖	✖	✖	Identity	▲	▲		✖			▲					
				✖	✖	✖	Course of Action	✖	▽		▲			▲	✖	✖			
	✖		✖	✖	✖	✖	Versioning	✖	✖		▲	✖	✖				▽	▽	▽
	✖		✖	✖	✖	✖	Author	✖	✖		✖			✖					
				✖	✖	✖	Confidence, count	✖	✖		✖	▲	✖	▽	▲		▽	▽	
				✖	✖	✖	Markings	✖	✖		✖		▲	▲	✖		▽	▽	
	✖		✖	✖	✖	✖	Artifacts	▲	▲	▲		▲		✖	✖				

Table 8. Typical use case and example CTI.

Typical Use Case	Example CTI
Email Blocklist	Simple block based on sender email address, domain or IP
Spam/Email Filter	Complex block based on sender IP, domain, email address, mail content, attachments, links, etc.
Firewall/Router ACL	IP address, port, may use connection rate (DDoS) or mask/simple patterns
NIDS	Complex, source/destination, addresses, URLs, file content, Malware IoC, Source reputation, etc.
HIDS/SIEM	Complex, source/destination, addresses, URLs, file content, Malware IoC, Source reputation, system IoCs (registry, files, paths).
Malware Analysis	Complex, known sources, poor reputation, email, file content, etc.
Human, SOC, DB	Complex dataset to build threat picture and analyse threats.

In Table 7, the formats and languages are graded on how well the test criteria have been met as per the following key: a blank means that the criterion or feature is neither met nor supported; the

'▽' symbol means that the feature is partially supported and some but not all criteria are met; the '✖' symbol means that the criteria are met or the feature is supported in a satisfactory manner; and the '▲' symbol means that the requirement criteria and feature requirements are exceeded. Table 8 below describes some very typical example use cases and examples of the types of CTI that those use cases may consume.

From the analysis of the various use cases, CTI formats and sampled feeds, it became clear that some were better suited at representing CTI for a given use case, e.g., due to being simpler or richer. This is illustrated in Table 9, where the available formats and languages are correlated against the security use cases according to the information that is given in Table 7.

Table 9. Formats and languages suitability per use case.

Formats and Languages	Typical Use Case						
	Email Blocklist	Spam/Email Filter	Firewall/Router ACL	NIDS	HIDS/SIEM	Malware Analysis	Human, SOC, DB
STIX 1.x	0.67	**0.74**	0.50	0.70	**0.72**	**0.70**	0.74
STIX 2.x	0.67	0.68	0.50	0.61	0.66	**0.70**	0.65
MAEC	0.67	0.63	**0.67**	0.78	0.66	**0.70**	0.71
CVRF	0.00	0.26	0.00	0.26	0.45	0.48	0.45
IDEA	0.83	0.68	**0.67**	**0.87**	**0.72**	0.63	**0.77**
CIF (platform API)	0.67	0.26	0.33	0.22	0.24	0.26	0.23
MISP (platform API)	0.67	0.68	**0.67**	0.65	0.69	0.67	0.71
Snort/Suricata rules	0.50	0.42	**0.67**	0.52	0.52	0.41	0.48
DNSBL	**1.00**	0.37	**0.67**	0.30	0.28	0.26	0.29
RSS	0.83	0.58	0.50	0.48	0.41	0.30	0.42
Text CSV	0.83	0.58	0.50	0.52	0.41	0.26	0.39
Text list	**1.00**	0.42	**0.67**	0.30	0.24	0.22	0.26

For each use case, the format or language achieving the highest suitability score is shown in boldface, with the score ranging from 0 (lowest) to 1 (highest). The expression used for computing the suitability score $s(f,u)$ of any format or language f against some use case u is given by

$$s(f,u) = \frac{1}{n(u)} \#\{a \in c(u) : f(a) \text{ covers } u(a)\}$$

where the set $c(u)$ is comprised of the criteria/features being applicable for the use case u, whose number is $n(u)$. $f(a)$ and $u(a)$ denote the level at which the criterion/feature a is supported and required, respectively. The ordering '▽', '✖', '▲' of the symbols in increasing support of features allows us to determine if the needs of a particular use case are being met. Let us take the *email blocklist* use case as an example, that is we have $u =$ "email blocklist" in the above expression. According to Table 7, this use case requires the features

$$c(u) = \{\text{Blocklist, IPv4 address, IPv6 address, Email address, Domain, Complexity}\}$$

and hence $n(u) = 6$. It is immediately seen in Table 7 that STIX 1.x protocol can adequately support only four out six features and therefore for $f =$ "STIX 1.x" we get $s(f,u) = \frac{4}{6} = 0.67$, which is also depicted in Table 9. Regarding the two features not counted for in STIX 1.x, namely *Blocklists* and *Complexity*, we see that the former is not supported while the latter implies that the protocol is unnecessarily over complex in the way that the information is provided (as stated in the assessment criteria of Table 6). It is interesting to note that the IDEA format (followed by STIX 1.x and MISP) is found to be the most suitable for the majority of the use cases considered, whereas it is located among the next most suitable formats and languages for the remaining ones—something that clearly justifies its design goals. On the other hand, Table 9 shows that the use case of "Firewall/Router ACL" is the one that most formats and languages can largely support.

The direction of the information flow is also a factor in the original design and the use of several of the formats were examined. Table 10 shows the flow direction and the formats noted as most suitable.

Table 10. Format suitability.

Direction	Suitable Format
From sensor/detection, (probe, IDS, log, alert, honeypot, etc.) to CTI collection or aggregation system.	IDEA, MAEC, text (device specific), CSV, custom JSON, proprietary, etc.
Between or extraction from CTI collection or aggregation systems.	STIX, MISP, MAEC, CVRF, CSV, custom JSON.
From CTI collection or aggregation systems to consuming cyber protective systems or devices.	CSV, IDS rules, Text blocklist.

CTI data from sensors or detection mechanisms tend to be specific to the source type or detection mechanism used. IDEA is a custom format designed to transport CTI from sensors to a central system. MAEC is quite popular with honeypot providers. CTI collection and aggregation systems, or extraction of data from them, are best suited to the formats that can provide the best fit for the data being shared or extracted. Such examples are a simple CSV for bulk IP data; CVRF for vulnerabilities; and STIX, MISP and custom JSON formats for a rich representation of CTI. The format used to distribute CTI to cyber protective systems or devices needs to be one that can be directly consumed, e.g., IDS rule sets, IP/domain lists, MD5 signatures, etc. When examining the suitability of the various formats and given the original use case and design criteria for the formats, the results are as we expected; this does not make any one format better than any other, it depends on the use and the requirements.

6. Conclusions

Through research and analysis, it quickly became apparent that the quantity of CTI sources and formats is vast. As noted above, more than half of the threat intelligence feeds sampled from these sources were either retransmitted or of unknown origin. The support for STIX is apparent in many platforms and the consensus from the research would suggest it has industry and community support. However, its use is not widespread and often poorly implemented. The trend is to use API or platform-specific formats that are a better fit with the given use case.

The question of which format to use depends on the use case; the creation, coding and use of custom JSON formats is a quick and simple way to meet requirements of a specific use case, or there may be a preference to adhere to existing standards. Our recommendation would be to use the best fit; the evidence from the research has shown that even the producers and key supporters of standards still produce their own, lightweight, custom JSON formats, regardless the time scales, processes and ratification needed by standards.

Our recommendations on the distribution and sharing of CTI is to follow the best practice, where applicable, with the common descriptors and conventions in the language. It was found that relying on the IDEA format (and possibly MISP or STIX) might constitute a best practice for the majority of the security use cases considered due to its ability in meeting their CTI needs. In addition, most of the formats are capable of supporting access control services being offered by means of a firewall or router.

Many of the issues we encountered with the quality and the distribution of CTI could be reduced by including the origin and freshness/timestamp data in feeds, keeping threat data complete. Clearly, the vast number of CTI sources offer an opportunity for further research into assessing and improving the quality of CTI feeds. Where resources are constrained, e.g., in IoT devices, better association between the threat and target surface could provide focused CTI able to more effectively protect these devices.

Author Contributions: A.R., S.S., and N.K. contributed equally. The authors read and approved the final manuscript as well as the authors order. All authors have read and agreed to the published version of the manuscript.

Funding: This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 786698. This work reflects authors' view and the agency is not responsible for any use that may be made of the information it contains.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Roberts, S.J.; Brown, R. *Intelligence–Driven Incident Response*; O'Reilly Media: Sevastopol, CA, USA, 2017.
2. Menges, F.; Sperl, C.; Pernul, G. Unifying cyber threat intelligence. In *Trust, Privacy and Security in Digital Business (TrustBus), Lecture Notes in Computer Science*; Springer: Berlin, Germany, 2019; Volume 11711, pp. 161–175.
3. Poputa–Clean, P. SANS Institute, Automated Defense, Using Threat Intelligence to Augment Security. Available online: https://www.sans.org/reading--room/whitepapers/threats/automated--defense--threat--intelligence--augment--35692 (accessed on 3 April 2020).
4. Appala, S.; Cam–Winget, N.; McGrew, D.A.; Verma, J. An actionable threat intelligence system using a publish–subscribe communications model. In Proceedings of the 2nd ACM Workshop on Information Sharing and Collaborative Security, Denver, CO, USA, 12–16 October 2015; pp. 61–70.
5. Wagner, T.D. Cyber Threat Intelligence for "Things". In Proceedings of the 2019 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (Cyber SA), Oxford, UK, 3–4 June 2019; pp. 1–2.
6. MITRE Corp. Making Security Measurable. 2018. Available online: https://msm.mitre.org/ (accessed on 3 April 2020).
7. Barnum, S. Standardizing cyber threat intelligence information with the Structured Threat Information eXpression (STIX). 2014. Available online: http://www.standardscoordination.org/sites/default/files/docs/STIX_Whitepaper_v1.1.pdf (accessed on 3 April 2020).
8. Connolly, J.; Davidson, M.; Richard, M.; Skorupka, C. Trusted Automated eXchange of Indicator Information (TAXII™). 2012. Available online: http://taxii.mitre.org/about/documents/Introduction_to_TAXII_White_Paper_November_2012.pdf (accessed on 3 April 2020).
9. OASIS Open Introduction to TAXII. 2018. Available online: https://oasis--open.github.io/cti--documentation/taxii/intro.html (accessed on 3 April 2020).
10. MITRE Corp. Cyber Observable eXpression (CybOX™) Archive Website. 2017. Available online: http://cyboxproject.github.io/ (accessed on 3 April 2020).
11. Sauerwein, C.; Sillaber, C.; Mussmann, A.; Breu, R. Threat Intelligence Sharing Platforms: An Exploratory Study of Software Vendors and Research Perspectives. In Proceedings of the 13th International Conference on Wirtschaftsinformatik, St. Gallen, Switzerland, 12–15 February 2017.
12. Zrahia, A. Threat intelligence sharing between cybersecurity vendors: Network, dyadic, and agent views. *J. Cybersecur.* **2018**, *4*, 1–16. [CrossRef]
13. Brown, S.; Gommers, J.; Serrano, O. From Cyber Security Information Sharing to Threat Management. In Proceedings of the 2nd ACM Workshop on Information Sharing and Collaborative Security, Denver, CO, USA, 12–16 October 2015; pp. 43–49.
14. Liu, R.; Zhao, Z.; Sun, C.; Yang, X.; Gong, X.; Zhang, J. A Research and Analysis Method of Open Source Threat Intelligence Data. In Proceedings of the 3rd International Conference of Pioneering Computer Scientists, Engineers and Educators (ICPCSEE), Changsha, China, 22–24 September 2017; Part I, Communications in Computer and Information Science. Springer: Berlin, Germany, 2017; Volume 727, pp. 352–363.
15. Sauerwein, C.; Pekaric, I.; Felderer, M.; Breu, R. An analysis and classification of public information security data sources used in research and practice. *Comput. Secur.* **2019**, *82*, 140–155. [CrossRef]
16. Abu, M.; Selamat, S.; Ariffin, A.; Yusof, R. Cyber Threat Intelligence—Issue and Challenges. *Indones. J. Electr. Eng. Comput. Sci.* **2018**, *10*, 371–379.
17. Pala, A.; Zhuang, J. Information sharing in cybersecurity: A review. *Decis. Anal.* **2019**, *16*, 1–25. [CrossRef]
18. Tounsi, W.; Rais, H. A survey on technical threat intelligence in the age of sophisticated cyber attacks. *Comput. Secur.* **2018**, *72*, 212–233. [CrossRef]
19. Menges, F.; Pernul, G. A comparative analysis of incident reporting formats. *Comput. Secur.* **2018**, *73*, 87–101. [CrossRef]
20. Mavroeidis, V.; Bromander, S. Cyber threat intelligence model: An evaluation of taxonomies, sharing standards, and ontologies within cyber threat intelligence. In Proceedings of the 2017 European Intelligence and Security Informatics Conference (EISIC), Athens, Greece, 11–13 September 2017; pp. 91–98.

21. Burger, E.W.; Goodman, M.D.; Kampanakis, P.; Zhu, K.A. Taxonomy model for cyber threat intelligence information exchange technologies. In Proceedings of the ACM Workshop on Information Sharing & Collaborative Security (WISCS), Scottsdale, AZ, USA, 3 November 2014; pp. 51–60. [CrossRef]
22. Asgarli, E.; Burger, E. Semantic ontologies for cyber threat sharing standards. In Proceedings of the 2016 IEEE Symposium on Technologies for Homeland Security (HST), Waltham, MA, USA, 10–11 May 2016; pp. 1–6.
23. Serrano, O.; Dandurand, L.; Brown, S. On the Design of a Cyber Security Data Sharing System. In Proceedings of the 2014 ACM Workshop on Information Sharing & Collaborative Security, Scottsdale, AZ, USA, 3 November 2014; pp. 61–69.
24. Sullivan, C.; Burger, E. "In the public interest": The privacy implications of international business-to-business sharing of cyber-threat intelligence. *Comput. Law Secur. Rev.* **2017**, *33*, 14–29. [CrossRef]
25. Wagner, T.D.; Mahbub, K.; Palomar, E.; Abdallah, A.E. Cyber threat intelligence sharing: Survey and research directions. *Comput. Secur.* **2019**, *87*, 101589. [CrossRef]
26. Zibak, A.; Simpson, A. Cyber threat information sharing: Perceived benefits and barriers. In Proceedings of the 14th International Conference on Availability, Reliability and Security, Canterbury, UK, 26–29 August 2019; pp. 1–9. [CrossRef]
27. Wagner, C.; Dulaunoy, A.; Wagener, G.; Iklody, A. MISP: The Design and Implementation of a Collaborative Threat Intelligence Sharing Platform. In Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security, Vienna, Austria, 24 October 2016. [CrossRef]
28. Skopik, F. *Collaborative Cyber Threat Intelligence: Detecting and Responding to Advanced Cyber Attacks at National Level*; CRC Press: Boca Raton, FL, USA, 2018.
29. Farnham, G. *Tools and Standards for Cyber Threat Intelligence Projects*; SANS Institute InfoSec Reading Room: Bethesda, MA, USA, 2013.
30. Friedman, J.; Bouchard, M. *Definitive Guide to Cyber Threat Intelligence*; CyberEdge: Annapolis, MD, USA, 2015.
31. EclecticIQ. Cabby—TAXII Client Implementation. 2018. Available online: https://github.com/EclecticIQ/cabby (accessed on 3 April 2020).
32. OASIS Open. OASIS TC Open Repository: TAXII 2 Client Library Written in Python. 2018. Available online: https://github.com/oasis--open/cti--taxii--client (accessed on 3 April 2020).
33. MITRE Corp. The MITRE Corporation. 2018. Available online: https://www.mitre.org/ (accessed on 3 April 2020).
34. MITRE Corp. About MAEC. 2018. Available online: http://maecproject.github.io/about--maec/ (accessed on 3 April 2020).
35. OASIS Open. Introduction to STIX. 2018. Available online: https://oasis--open.github.io/cti--documentation/ (accessed on 3 April 2020).
36. OASIS. Introduction to STIX. 2018. Available online: https://oasis--open.github.io/cti--documentation/stix/intro (accessed on 3 April 2020).
37. OASIS. OASIS CTI CybOX Subcommittee. 2018. Available online: https://www.oasis--open.org/committees/tc_home.php?wg_abbrev=cti--cybox (accessed on 3 April 2020).
38. OASIS. OASIS Cyber Threat Intelligence (CTI) TC. 2017. Available online: https://www.oasis--open.org/committees/tc_home.php?wg_abbrev=cti (accessed on 3 April 2020).
39. MITRE Corp. CVE—Common Vulnerabilities and Exposures. 2018. Available online: http://cve.mitre.org/index.html (accessed on 3 April 2020).
40. OASIS Open. CSAF Common Vulnerability Reporting Framework (CVRF) Version 1.2. 2017. Available online: https://docs.oasis-open.org/csaf/csaf-cvrf/v1.2/cs01/csaf-cvrf-v1.2-cs01.html (accessed on 3 April 2020).
41. CESNET. Intrusion Detection Extensible Alert. 2018. Available online: https://www.cesnet.cz/en/index (accessed on 3 April 2020).
42. CIRCL. Malware Information Sharing Platform MISP—A Threat Sharing Platform. 2018. Available online: https://www.circl.lu/services/misp--malware--information--sharing--platform/ (accessed on 3 April 2020).
43. CSIRT Gadgets LLC. CSIRT Wiki, Getting Started—Welcome to the CSIRTG–EX Software Development Kit. 2018. Available online: https://github.com/csirtgadgets/csirtg/wiki (accessed on 3 April 2020).
44. Cisco. Snort. 2018. Available online: https://snort.org/ (accessed on 3 April 2020).
45. OISF. Suricata Open Source IDS / IPS / NSM engine. 2018. Available online: https://suricata--ids.org/ (accessed on 3 April 2020).

46. Spamhaus. Understanding DNSBL Filtering. 2018. Available online: https://www.spamhaus.org/whitepapers/dnsbl_function/ (accessed on 3 April 2020).
47. Winer, D. RSS 2.0 Specification. Available online: https://cyber.harvard.edu/rss/rss.html (accessed on 3 April 2020).
48. FireEye, Inc. Free Security Software—IOC Tools (Indicator of Compromise). Available online: https://www.fireeye.com/services/freeware.html (accessed on 3 April 2020).
49. Mandiant. GitHub Repository. Available online: https://github.com/mandiant (accessed on 3 April 2020).
50. Danyliw, R. Internet Engineering Task Force (IETF), RFC 7970. The Incident Object Description Exchange Format Version 2. Available online: https://tools.ietf.org/html/rfc7970 (accessed on 3 April 2020).
51. Lookingglass. Welcome to the OpenTPX Project! Available online: https://opentpx.org/ (accessed on 3 April 2020).
52. Cisco Security Alerts. Available online: https://tools.cisco.com/security/center/cvrf_20.xml. (accessed on 3 April 2020).
53. Oracle Security & Patch Update Advisories. Available online: http://www.oracle.com/ocom/groups/public/@otn/documents/webcontent/1932662.xml. (accessed on 3 April 2020).
54. Red Hat Security Advisories. Available online: https://www.redhat.com/security/data/cvrf/ (accessed on 3 April 2020).
55. Malc0de Database. Available online: http://malc0de.com/database/ (accessed on 3 April 2020).
56. NC4 Soltra. Connecting to PickupSTIX. Available online: https://www.soltra.com/en/documentation/ctx--soltra--edge/connecting--to--pickupstix/ (accessed on 3 April 2020).
57. Abuse.Ch. Ransomware Tracker. 2016. Available online: https://ransomwaretracker.abuse.ch/tracker/ (accessed on 3 April 2020).
58. NC4 / Soltra LLC, PickUpStix. Available online: https://www.soltra.com/en/documentation/ctx--soltra--edge/connecting--to--pickupstix/ (accessed on 3 April 2020).
59. Anomali, Limo—Free Intel Feed. Available online: https://www.anomali.com/platform/limo (accessed on 3 April 2020).
60. Leach, P.; Mealling, M.; Salz, R. RFC4122, A Universally Unique IDentifier (UUID) URN Namespace. Available online: https://tools.ietf.org/html/rfc4122 (accessed on 3 April 2020).