Automated Defense

Using Threat Intelligence to Augment Security

GIAC (GCIH) Gold Certification

Author: Paul Poputa-Clean, paul.poputaclean@gmail.com
Advisor: Mark Stingley

Accepted: January 15 2015

Abstract

Threat Intelligence has become the 2014 security buzzword. While there are some valiant efforts to create Threat Intelligence on the open source and commercial front, the ingestion and utilization of Threat Intelligence is still a fringe science in a fragmented market. There seems to be a proliferation of products flooding the Threat Intelligence data repository market, but the meaningful integrations appear to still be lacking. This paper will describe the current environment of the Threat Intelligence industry, the areas of current research in sharing and using Threat Intelligence, as well as some potential future use cases for Threat Intelligence to further streamline the Network Security Monitoring and Incident Response processes. Lastly, some code examples should help the reader kick start a basic Threat Intelligence program.

Introduction

Threat Intelligence means different things to different people. Gartner defines it as: “evidence-based knowledge, including context, mechanisms, indicators, implications and actionable advice, about an existing or emerging menace or hazard to assets to that can be used to inform decisions regarding the subject's response to that menace or hazard.”(McMillan, 1)

There are a few ways of looking at Threat Intelligence, but a very useful classification scheme is to look at it as Strategic Intelligence versus Tactical Intelligence. Anton Chuvakin describes the reasons behind the classification in details. Some of the criteria to classify intelligence include:

- Gathering Methods –Threat Intelligence ranges from information gleaned from a honeynet on the Internet all the way to closely guarded secrets stolen by operatives embedded in the adversary's country or even organization.
- Cost – the subscription model as well as the price for the sources
- Main Usage – the primary utility of the intelligence
- Target Audience – the position in the customer organization that is most likely to take advantage of the intelligence.
- Specificity – the level of details in the Threat Intelligence. For example the IP address (194.201.253.5) is significantly more specific than to say the Zeus configuration file is hosted on infected web hosting servers in the UK[1]. Specificity also translates into the ability for machines to ingest the intelligence (specific intelligence), or for humans to make medium to long term decisions based on vague intelligence.
- Lifespan – the expected life of the Threat Intelligence. The interplay between attackers, defenders, and law enforcement will render some of the attacker's infrastructure dormant or obsolete. Therefore specifics about the infrastructure

[1] http://www.malwaredomainlist.com/mdl.php?search=194.201.253.5&colsearch=All&quantity=50

might grow obsolete in the short term, while attackers' tactics, techniques and procedures will be valid for longer periods. For state-based attackers, the overall information strategy and targets will have the highest lifespan.

On one hand, Strategic Threat Intelligence is very high-level and geared towards strategic planners, executives and threat analysts. It is very expensive, as a significant degree of human analysis goes into creating it. The companies producing Strategic Intelligence typically have operatives that interact with the adversary as well as their infrastructure. Strategic Intelligence describes general attacker trends and long term plans and strategies, including their Tactics, Techniques, and Procedures (TTPs), tools preferred, and industries targeted. Ambiguity is acceptable at this level, making it not very machine-friendly.

On the other hand, Tactical Intelligence is very low-level. It varies in price, but is easier to quantify than Strategic Intelligence. Examples of Tactical Intelligence include hashes, IP addresses, and file names. It is very specific and geared towards automatic ingestion. As such, the life expectancy of the observables is significantly lower since the attackers are able to change these indicators fairly easy.

At the most pedestrian level, Threat Intelligence has come to be synonymous with indicators or observables. MITRE defines observables in the context of an observable event or property related to cyber security activities (MITRE Cybox Handout, 1). They are relevant in the context of the Tactical Intelligence. Historically, observables have been focused on IP addresses and domains. They have been used in blacklists in the routing interface or in blacklists in the web filter. As the defensive measures are evolving, higher-level languages are developing to refer not only to the basic observables, but specific order of occurrence as well.

Storing and Sharing Threat Intelligence

One of the main issues with Threat Intelligence is the ability to track it and use it for context. Organizations end up with very distributed storage for its Threat Intel including pieces of paper, text files, Excel spreadsheets and emails. While this approach helps with the extemporaneous storage and dissemination of Threat Intelligence, it breaks down with time and quantity of observables. The first attempts to organize the Threat

Intelligence observables and context include Excel, SharePoint, SIEM watch lists, and home-built repositories.

However, storage is just the beginning of the problem. Sharing becomes even more challenging without a central platform. The sharing classification is based on the analyst's whim. The information often takes the form of emails, text messages, text files, Excel spreadsheets, and PDF documents. While it might be sufficient for a small number of observables, this method becomes increasingly less scalable the more observables are shared with the more people. A more organized, machine-readable sharing format is necessary to achieve the full benefits of Threat Intelligence.

Several high-level languages have been developed in attempts to facilitate precise sharing:

- Open Indicators of Compromise (openIOC) - Mandiant developed the openIOC format to help in tracking advanced adversaries, their campaigns, and their tools. They open sourced the format to facilitate using Threat Intelligence in the fee tools offered by Mandiant (IOC Editor, Redline, IOC Finder). The format is becoming more and more popular as a sharing platform with other commercial vendors.
- Structured Threat Information Expression (STIX) – is an effort by MITRE to create a “fully expressive, flexible, extensible, automatable, and as human-readable as possible” language (MITRE STIX). It is gaining popularity with multiple tools and libraries, driven mainly by the financial services industry and the Federal Government.
- Cyber Observables (CybOX) – is a standardized schema for describing observables created by MITRE. STIX acts as a wrapper for the intelligence recorded in the format.
- Trusted Automated Exchange of Indicator Information (TAXII) – is a set of services and messages created to describe the protocol of sharing Threat Intelligence. This protocol was designed to facilitate the sharing of Intel in the STIX and CybOX formats.

- The Incident Object Description and Exchange Format (IODEF) was developed in 2007 by the IEFT as a helper tool for Computer Incident Response Teams. It is based on XML and is focused on describing cyber incidents. This standard seems to be implemented by a few tools, including Foundstone and DFLabs.

The schemas for openIOC and STIX are still evolving as their acceptance increases. Their strength comes from their extensibility and their ability to describe the incidents, the actors, and the observables. The biggest gap in the current environment is the inconsistent support from the Threat Libraries and the commercial tools.

Regardless of the format, machine-driven sharing is the next hurdle in using Threat Intelligence. Moving away from email and human-readable files as the main mechanism for sharing, the basic open source sharing platforms are CIF federations, or TAXII services. TAXII allows the servers to share STIX documents automatically. Some of the commercial Threat Intelligence Platform vendors are looking at sharing as a social platform, creating trust groups that allow customers to share intelligence amongst each other. The platform security allows better controls on sharing, but it limits automation functions to paying customers.

Before delving into specific libraries, it is worth covering the Traffic Light Protocol. The US-CERT developed it as a set of designations that help to classify information based on its sensitivity. It has four levels: TLP: WHITE, TLP: GREEN, TLP: AMBER, and TLP: RED. The restrictions for sharing range from information shareable without restrictions (TLP: WHITE) to restrictions against sharing the information with anyone outside the current information exchange (TLP: RED). The framework for sharing is currently used by some industry-based communities, as well as the Federal Government.

Threat Libraries / Threat Intelligence Platforms

A concept that is gaining popularity in solving the collection, storage, and sharing problems in the Threat Intelligence space is the Threat Library or Threat Intelligence Platform. Its main functions are to collect the intelligence from Open Source Intelligence (OS-INT) as well as commercial feeds, store it in a secure, flexible, and easily accessible manner, enable integrations with defensive tools, facilitate controlled sharing with other enterprises depending on various criteria, and provide some fusion capabilities to use the context, alerts and the Threat Intelligence itself to create more precise observables (ThreatConnect, 17).

Recently, a number of open source projects and commercial enterprises have begun to gain popularity in this space as they promise a more organized storage of the observables and an improved context around the alerts.

Some notable Open Source Intelligence Libraries include:

- The Collective Intelligence Framework (CIF - https://code.google.com/p/collective-intelligence-framework/). This framework was developed by REN-ISAC, the educational Information Sharing and Analysis. It was developed to help ingesting IP addresses and domain names, with some support for hashes. Written in Perl, it stores the observables in postgresSQL and provides web API as well as Chrome and Firefox extensions. It provides a good method to ingest basic indicators and has a great way to output the indicators into a few usable formats. It can output into multiple formats and integrate with various tools including Snort, Bro, Bind, TippingPoint, and Elsa. Sharing is facilitated among different CIF instances via the Federation Service
- Collaborative Research into Threats (CRITs - http://crits.github.io/threat_sharing.html). The MITRE Corporation has been working on a Threat Intelligence library and has been offering it free of charge, with some legal restrictions. In 2013 they open sourced the project. CRITs integrates with TAXII servers to facilitate sharing intelligence, and allows manual input of STIX files, as well as domains, IPs, samples, emails, and other indicators. To share the information, CRITs will allow to output CSV, STIX, and JSON. Another feature of CRITs is the ability to adjust the confidence and impact of the indicators, which, combined with the extensive REST API, allows the defenders to create multiple dynamic lists that they can use to update specific systems.
- Mantis (http://django-mantis.readthedocs.org/en/latest/). In 2013 Siemens open sourced their effort into Intelligence Libraries and presented it at a 2014 FIRST

conference. It is able to import and process most of the current high language formats (IODEF, openIOC, STIX).

- Malware Information Sharing Platform (MISP – http://www.misp-project.org/, http://www.circl.lu/services/misp-malware-information-sharing-platform/). NATO developed MISP to help in tracking and analyzing rare malware. It integrates with ArcSight, IDS (Snort), various sources (importing and exporting openIOC), GFI Sandbox, as well as XML, CSV, and a RESTful API. Sharing occurs in a MISP federation.
- Avalanche | SOLTRA EDGE (http://www.soltra.com/) – Financial Services Information Sharing and Analysis Center (FS-ISAC) had an initiative to create a common platform to share indicators of compromise. Avalanche emerged as a result from the initiative. Though it started as a free model, it now developed into a quasi-commercial product supported by Soltra. It was designed to facilitate sharing between the member organizations of FS-ISAC, but it is making some inroads in other information sharing groups.

There are multiple commercial libraries in this fairly young market. It is apparent that each vendor started from a different standpoint, though the market maturation is encouraging all of the vendors to adopt a common set of features. Among these core features the following seem to gain importance:

- Integration with SIEM and SIEM-like systems — this seems to be the most evident integration point, as the SIEM will be the de-facto fusion platform for security events and intelligence. The SIEM integration is facilitated by the fact that SIEMS collect and correlate various logs and allow the security analyst to pivot to more details regarding the discovered indicator by using various integrations.
- Integration with other detective controls – some of the Threat Intelligence Libraries are able to integrate with Bro, Moloch, or Snort, making it easier to alert on detected indicators.
- Integration with preventive controls – next generation firewall blacklists are a

great case for enhancing the system's utilization and maximizing current investments. Various vendors provide APIs or frameworks to enable the integration.

- Integration with other products created by the same vendor – the Intel Security Threat Intelligence Exchange (TIE) and the Palo Alto WildFire platforms attempt integration and Threat Intelligence exchange between the different security offerings created by the respective vendors.
- Data enrichment – services ranging from integration with other intelligence platforms, to seamlessly ingesting OS-INT and commercial feeds, to having a sandbox service, and to mapping the malicious indicators.
- Sharing – ability to share intelligence with other organizations inside and outside of the vendor's platform is gaining popularity as well. It is worth noting that the automation and integrations will generally not be available to a non-customer.

An interesting force currently driving this market is the creation of industry-based Information Sharing and Analysis Centers (ISAC). They are encouraging a clustering around particular storage and sharing platforms on a per industry basis. The Financial Services ISAC seems to be leading the way in sharing Threat Intelligence with the creation of Avalanche / Soltra Edge and their extensive use of Threat Intelligence. They also seem to be the most open about sharing Threat Intelligence amongst themselves. Other ISACs take advantage of discounted models from some of the commercial vendors in order to create a similar sharing environment. The increased sharing reinforces the necessity of a common platform or format for the observables.

Uses for Threat Intelligence

The driving focus of using Threat Intelligence is to gain an advantage over the adversary by blocking or delaying their attacks, detecting their presence, or degrading their infrastructure. Different verticals are able to perform different functions to thwart the attackers.

David Bianco has a great way of describing the impact defenders are able to make on the offensive side with the Pyramid of Pain illustration:

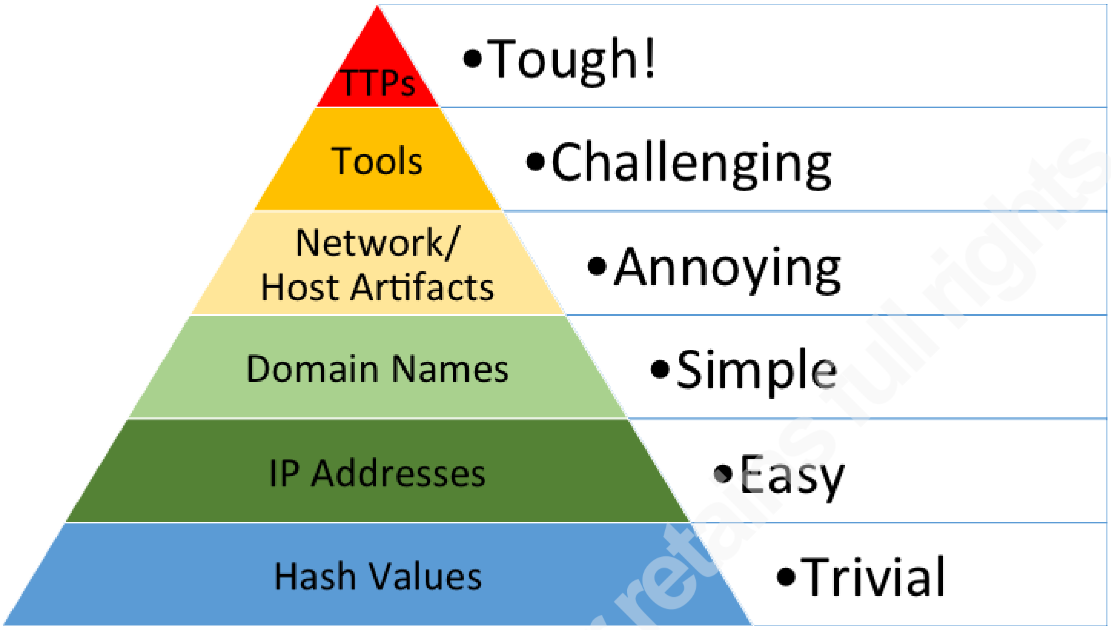


Figure 1: David Bianco - Pyramid of Pain

Each level of the pyramid are types of observables that defenders can include in their blocking mechanisms. The vertical axis is the amount of work required for the attacker to replenish their arsenal after the defenders successfully blocked the indicators.

A hash value is not a very durable indicator as attackers are able to generate polymorphic code and even accidentally change hashes by introducing null bytes in the code. A couple of mitigating techniques might help with that issue, specifically SSDEEP and IMPHASH. Both methods take the approach of grouping together similar files. SSDEEP will yield a percentage of similarity for the files based on common bits of code (Korblum), while IMPHASH will hash the import table of the executable files and look for similarities (Mandiant). Unfortunately, most of the security vendors have yet to adopt these methods, as they are quite a bit more difficult to successfully use than the simple MD5 match.

On the other end of the spectrum, Bianco argues that, if it were possible, blocking and detecting Tactics, Techniques, Procedures (TTPs) would bring the most pain to the adversary. Successfully detecting or blocking based on TTPs would force the adversaries to completely change the way they do business, severely increasing their costs. If we were able to reliably detect and block the adversary's TTP of sending in a spear phish containing a malicious PDF, then moving horizontally to the web servers, implementing a webshell, then extracting valuable data via ICMP packets, the adversary would not only need to change the tactical parts of their infrastructure (programs used, IP addresses, domains and URLs), but they would also need to change the way they use the programs already on the victim machines, and the initial point of compromise. This would mean retraining of the staff, setting in place other methodologies, and developing or learning new tools. Obviously, this would be a lot more expensive and time consuming than having to change a couple of jump servers.

When overlaying the Pyramid of Pain model with the Kill Chain, Bianco creates a very interesting coverage map for indicators, as well as a good starting point for creating key performance indicators for Threat Intelligence driven defense. The Kill Chain methodology to Threat Intelligence Based defense is detailed by Hutchings, Cloppert and Amin in Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. They assessed that rather than continuing the vulnerability-based signature creation, defenders should focus on trying to detect the attackers at each stage of their process. Bianco builds on this work by mapping out indicator types and defensive measures to each portion of the chain (Bianco, What do you get when you cross...). For example, an MD5 hash can be covered by multiple defensive systems in different phases of the Kill Chain. The Reconnaissance and Weaponization phases happen outside of the defender's view, so no hashes can be detected. Delivery, on the other hand, will involve getting the executable code to the victim's network, so Host Intrusion Detection Systems, in-line sandboxes, and Host Intrusion Detection Systems should see the hash. In the Exploitation phase, Host Anti-Virus and Host Intrusion Prevention Systems would be able to take advantage of the hash value. Once the successful exploitation occurs and the Command and Control (C2) communication starts, the primary way MD5 hashes are useful is to identify lateral movement if it crosses the purview of an IDS, or to look for important files being exfiltrated via Network Data Loss Prevention. Performing this exercise for multiple observables will help defenders answer the basic questions regarding levels of coverage for various Threat Intelligence observables.

An important subject to consider when using Threat Intelligence is the observables' life cycle. Attackers may take parts of their infrastructure offline and start using other compromised computers to attack, rendering lists of observables obsolete very quickly. To allow for this cat and mouse game, the Threat Intelligence consumers should plan on creating processes to continuously reassess the categorization the observables based on currently confirmed confidence levels and impact. More specific indicators tend to have shorter lifespans and will require more constant updating and verification.

In How to Collect, Refine, Utilize and Create Threat Intelligence, Anton Chauvakin identifies the following several cases for Threat Intelligence, focusing on the differences between Tactical and Strategic Intelligence. The use cases might be able to be contained in several larger buckets:

- Security Planning
- Enterprise Security Monitoring (detection)
    - Prevention
- Incident Response
    - Alert Triage
    - Threat Detection
- Threat Intelligence Fusion:
    - Threat Discovery
    - Threat Assessment

These use cases can be further described from the viewpoints on specificity, type of intelligence, the desired outcome from ingesting and using the intelligence feeds, as well as a list of the performance indicators of using the intelligence in the defensive activities. The specificity of the performance indicators will probably follow the specificity of the observables.

Use Case	Specificity	Strategic / Tactical	Product	Key Performance Indicators
Security Planning	Low	Strategic	Security Vision, Response Plans, Security Roadmaps	Success in response to targeted attack
Threat Intelligence Collection and Fusion	Low	Both	Threat Intelligence Reports and Indicators	
Incident Response	Medium	Both	Incident Response	Time to containment, correct identification and scoping of incidents
Enterprise Security Monitoring	High	Tactical	Blocks, Alerts, Context	Time to detection, time to escalation, false positive rate for alerts

Table 1: Threat Intelligence use cases

1.1.1. Strategic Planning

The broadest use for Threat Intelligence is planning for the future. Strategic Intelligence is far more useful in this scenario. Knowing where the adversaries will spend their resources would be a great advantage in planning the future capabilities the potential target will have to develop. For example, if the attackers will focus on using Oracle-specific attacks and the target has a strong Oracle install, a wise investment decision would be to try to protect that install, even at the expense of mitigating other (preferably low) security risks. This knowledge, along with more specific intelligence on the TTPs of the attackers might dictate future staffing models for the defenders, as well as refocusing of skillsets, acquired technologies, and, in some cases, the modification of end-user behavior.

1.1.2. Threat Intelligence Fusion

Should an organization stand up a Threat Intelligence function, several tasks will be necessary in order to properly support Threat Intelligence-driven Defense. First of all, the organization will have to define an overall strategy for Threat Intelligence. Second, the collection of Threat Intelligence should be established into a process (even if it includes downloading blacklists available on the Internet). Furthermore, after collecting the raw intelligence, the organization should enrich and contextualize the Threat Intelligence – a Threat Library will come in handy here. After collecting and storing the Threat Intelligence, the organization should have a procedure for disseminating the Intelligence to the appropriate controls. Overriding the entire process is the fusion function for Threat Intelligence. This function means both assessing the impact of the observables to the organization (by attributing them to actors and assessing the actor's activity in defending organization's vertical), and setting the lifespan of the observables. The Threat Intelligence fusion is a self-perpetuating concept. Some organizations can afford to staff a department to perform all the functions, while others will be better off to take advantage of commercial offerings for parts or all the Threat Intelligence Fusion function.

1.1.3. Incident Response

Incident Response should be able to take advantage of both Strategic and Tactical Intelligence. Once an incident has been declared, Tactical Intelligence is useful in determining the scope of the incident. If the MD5 hash of the tool used by the attacker will pop up somewhere else in the organization after a sweep, the scope will have to be extended to that other machine. Once the IR team is able to attribute the threat to an actor or set of actors, Strategic Threat Intelligence that details their TTPs will help the target organization determine the directions it needs to focus its hunting in order to find other targets or compromised computing assets – if the adversaries prefer web shells, the web servers should be a great starting point for sweeps and hunting activities.

1.1.4. Enterprise Security Monitoring

David Bianco defines Enterprise Security Monitoring as Network Security monitoring with internal and threat contextual information (Bianco, ESM). Network Security Monitoring is Intrusion Detection with contextual information for the alerts. Put another way, Enterprise Security Monitoring is being able to start with an alert and provide the appropriate context regarding the network traffic, host details, and organizational membership.

Defenders should be able to take a host antivirus alert and start correlating it to network traffic, to logs, and to Threat Intelligence in order to determine whether the alert is a false positive, a commodity piece of malware, or part of an advanced attack. Based on the contextual information, the analyst should be able to quickly escalate the alert into an incident and start the Incident Response process. Threat Intelligence is beneficial in augmenting the basic signature-based alert and in performing an initial assessment on which incident response path should be taken. Defenders can search VirusTotal for the hash in a detection, and if it is associated with activity from any adversary organizations currently targeting the analyst's organization, the intense Incident Response path should be taken.

Threat Intelligence integrations should provide the Security Operation Center (SOC) analysts with information for better hunting activities. If the organization has the resources to dedicate to hunting, the SOC analysts would be able to take the vague pieces of available intelligence and to search the enterprise for suspicious activity.

Tactical Threat Intelligence in ESM and Incident Response

Having covered multiple facets of what Threat Intelligence is and the general use cases, it is appropriate to focus on a more detailed use case of tactical threat intelligence, observables automatically tracked by technical controls.

Not all Threat Intelligence is appropriate for a blocklist. Assuming that the Threat Intelligence is already vetted and the obvious bad observables are removed, the value of the observables is still a variable. It is useful to look at the Tactical Indicators from the perspectives of the impact to the organization and the confidence in the indicator. The impact is determined by the potential damage to the organization if the attackers are successfully using this observable. For example, a crypto-locker Command and Control (C2) IP address would have a high impact for a company running Microsoft Windows workstations due to the destructive nature of the threat. An IP known to probe for MS14-066 would be of low impact to a company running Linux-based web servers. The second perspective of confidence in the indicator refers to the organization's ability to vet the Threat Intelligence and to confirm the observable as malicious. An OS-INT feed from the

Internet should be regarded with a lower confidence level than a feed from a reputable commercial Threat Intelligence provider. Both impact and confidence will be functions of the fusion capabilities of the organization and should be changed as the Threat Intelligence landscape changes.

Based on impact and confidence, defenders can build a matrix to help in the decision where to apply the observable:

Impact \ Confidence	Unknown	Low	Medium	High
Benign	ignore	context	context	context
Low	context	context	context	context
Medium	context	alert	alert	alert
High	context	alert	block	block
Critical	context	alert	block	block

Table 2 Confidence matrix

From here, the defender organization can have its Threat Library and distribute the tactical indicators to the appropriate systems based on the tolerance for false positives:

Action	Defensive Technologies
Block	IPS, Next Generation Firewall, Web Filter, Host IPS
Alert	IDS (Bro, Snort, commercial), SIEM, Host IDS
Context	Threat Library integrations
Ignore	/dev/null

Table 3: Confidence-based actions

An important characteristic of actionable Threat Intelligence is the integration with the defensive systems (Holland, 3). The quality of the blocks and alerts will become consistent and repeatable and will not be contingent on the analysts' process to add the indicators. This automatic process should also help keep track of the observables and free up the analysts' time for hunting and other fusion functions.

When talking automation, a defender has various options depending on the capabilities and maturity of their organization. The most basic application of Threat Intelligence is to enable the vendor-provided feeds in the defensive tools, and in most cases activating this service should be as easy as clicking checkbox. Antivirus, Web Filter, and Next Generation Firewall vendors will be able to ingest vendor-specific intelligence or perform lookups to cloud services. For example, a query against VirusTotal will sometimes prove more useful than the signatures as the malware authors will upload different samples to find one that won't be caught by the targeted engine.

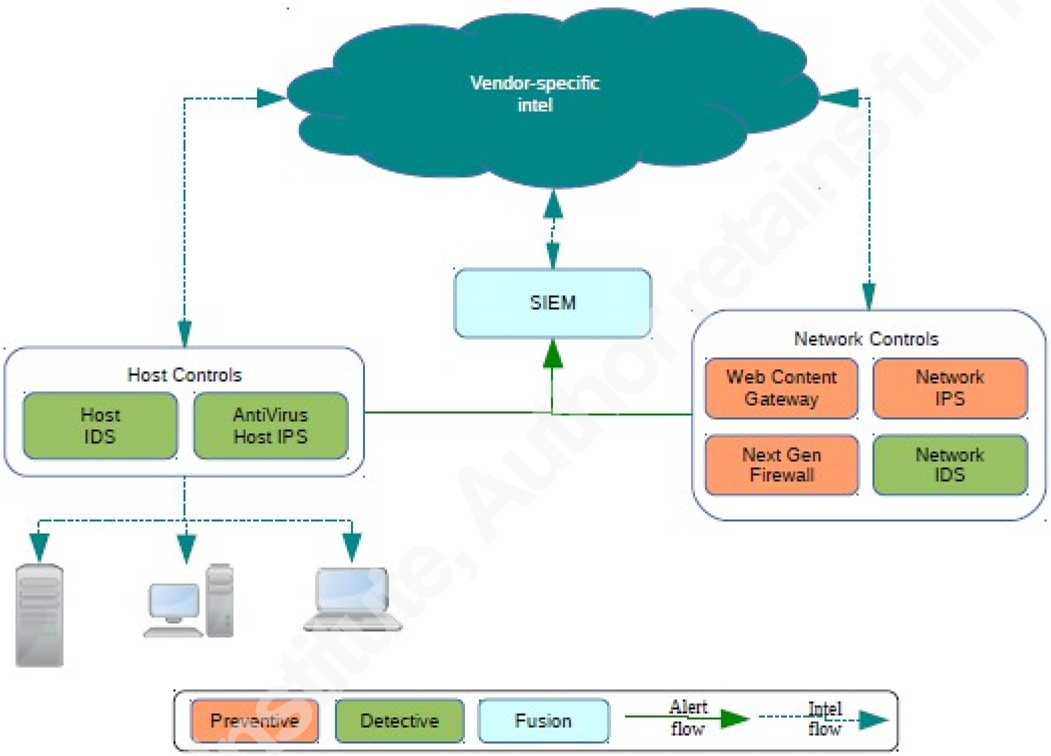


Figure 1: Enabling default Intel Feeds on Security devices

The vendor-specific Intel approach will take advantage of the Threat Intelligence capabilities of the various tools and be trivial to implement. Unfortunately, this approach will not be very effective at protecting against targeted threats, especially if those actors are using infrastructure that is otherwise not malicious. Nevertheless, it can be a good start.

The next step of including Threat Intelligence in defensive capabilities is to add Threat Intelligence manually to the blocking and alerting mechanisms. Intelligence can be collected by scripts from OS-INT and commercial intelligence sources. Using this data, several systems could be updated with signatures and blacklists on an ad-hoc basis.

Organizations will resort to this if they are currently building the Threat Intelligence function or if they have limited personnel.

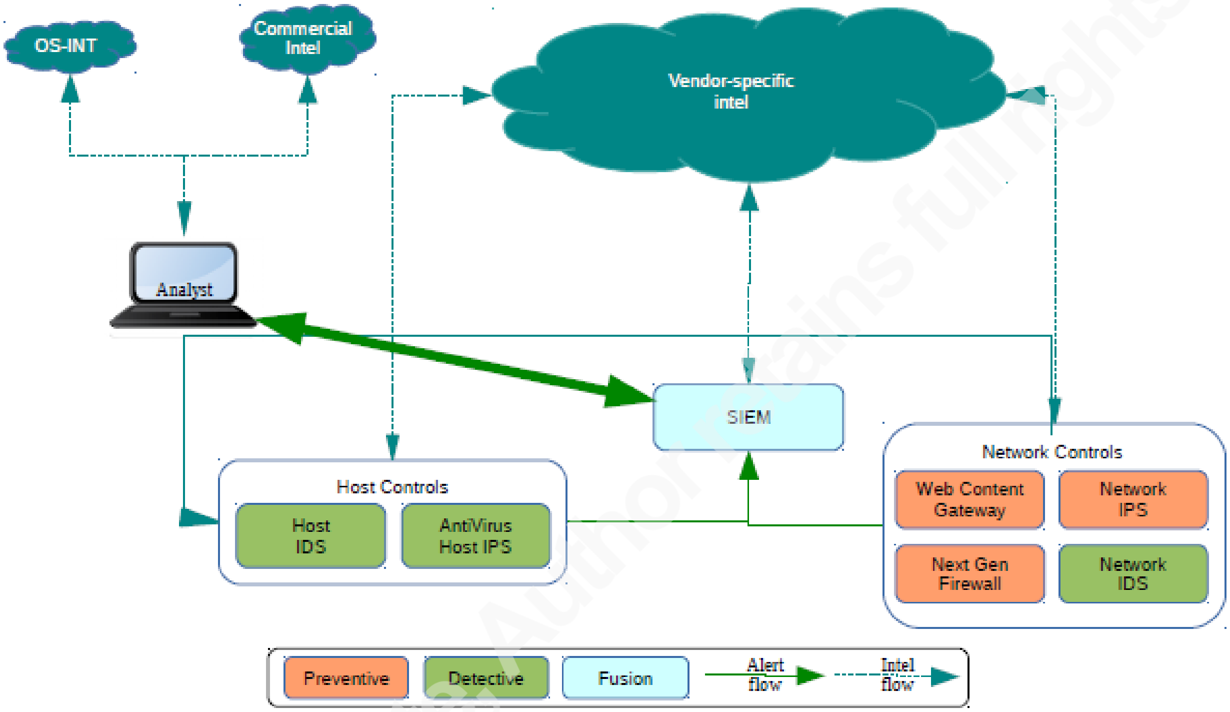


Figure 2: Augmenting default feeds with anecdotal data

Augmenting Vendor-centric combined with anecdotal intelligence will become a good starting point for the interplay between the Threat Intelligence function and the Incident Response function. As a response to various breaches, the organization will start implementing alerts in a SIEM, custom domain lists in the web proxy, and IP blacklists in the firewall to try to stop or alert for similar activity in the future.

While it is a good augmenter to basic vendor-provided method and allows blocking and alerting based on intelligence about actors targeting the specific organization, this approach under-performs with documenting and keeping up with the observables' life cycle. It is also time intensive for the defensive organization and has issues scaling with the number of observables. Another scale problem that will start becoming more evident with the increase in the number of tracked observables is the false positive rate. Setting up Snort signatures to alert on any traffic to .ro domains will generate so many alerts for legitimate traffic that it will overwhelm the staff responsible for the log monitoring to the degree that they will ignore interesting domains due to the alert fatigue.

This issue should be seen as a great springboard for setting up a continual improvement program for the alerts. As the number of observables tracked increases, the need for a lifecycle and a confidence level becomes more evident. One option for a stopgap is to use the Threat Intelligence observables (domains and IP addresses) as a context enhancer in the SIEM rather than as an alert generator in the detective mechanisms.

A third step in the maturity of the Threat Intelligence delivery model is to have a central collection and automation point for the dissemination of the observables. For example, the defenders can have a modified version of MLSec's Combine that collects indicators from various OS-INT feeds, enriches them, and dumps out the CSV for manual ingestion. The organization should script against the APIs and integrations available in the various defensive technologies in order to maximize the benefit from this intelligence. For the more targeted indicators, the organization would still need to plan on updating lists automatically even if it is a text file that Combine merges with the rest of the Threat Intelligence.

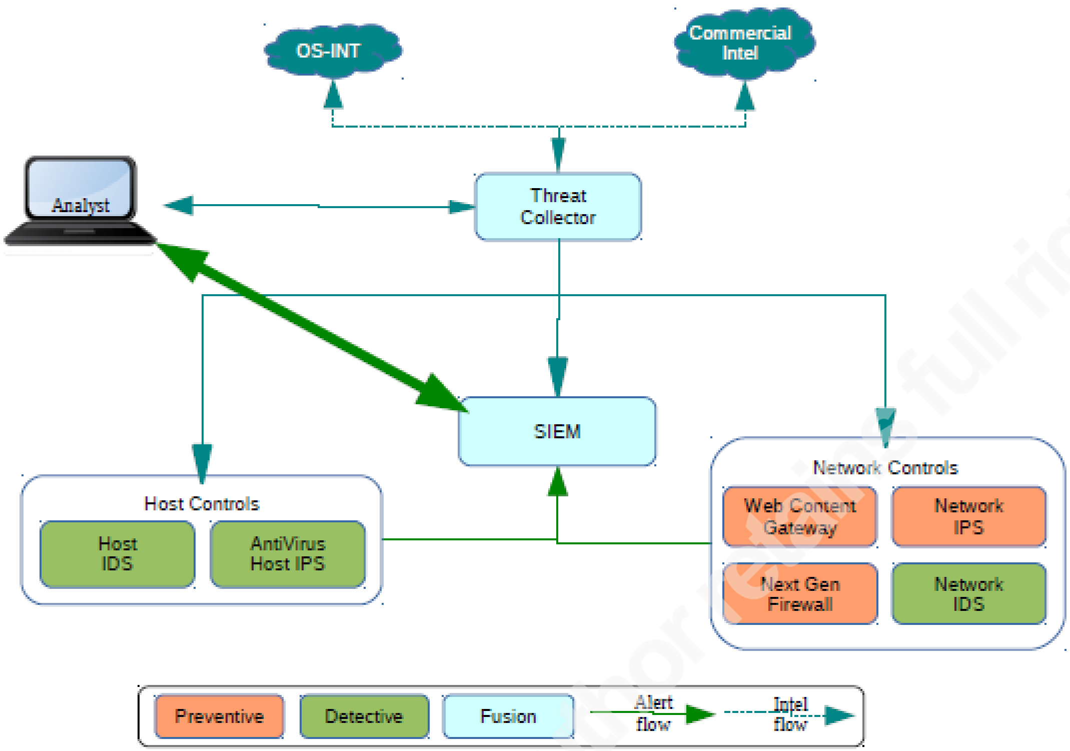


Figure 3: Central collection of Intelligence

This should provide the capabilities to load and analyze a wider range of observables in a more organized fashion as well as provide a rudimentary confidence schema. The problems related to the number of observables noticed in the previous approach will intensify, encouraging a creative solution. Here is where the defenders can start being more discerning with the observables. For example, if outputting to CSV, the defender can rank the intelligence providers and only import the observables from the better sources in any blacklists. The rest of the indicators might still be useful when imported into a watch list in the SIEM where they would increase the alert level of a triggered event If there is an event for an executable download and the foreign IP happens to be connected to malware in the alienvault blacklist, we should raise the alert level, but if that IP is one of Amazon's public IPs, alerting on all traffic to it will get tiresome. By only importing highly ranked sources, intelligence will help defenders spend valuable Security Operations Center (SOC) time analyzing alerts that have already passed some contextual test.

As the Incident Response, SOC, and Threat Intelligence functions start to integrate more at this stage of the maturity, a feedback loop can be created on the quality of the alerts coming from the SIEM, the quality of the observables gleaned by the Threat Intelligence function, and the quality of the observables gathered by the Incident Response function.

A mature model of ingesting and disseminating Threat Intelligence would include a number of open source and commercial intelligence feeds that are ingested automatically by a collector. After the initial collection, the observables will be pushed to a Threat Library that tracks both the observables and the contextual information (both Tactical and Strategic Threat Intelligence). The Threat Library would then update the controls (detective and preventive) with the correct list of observables (according to the confidence matrix).

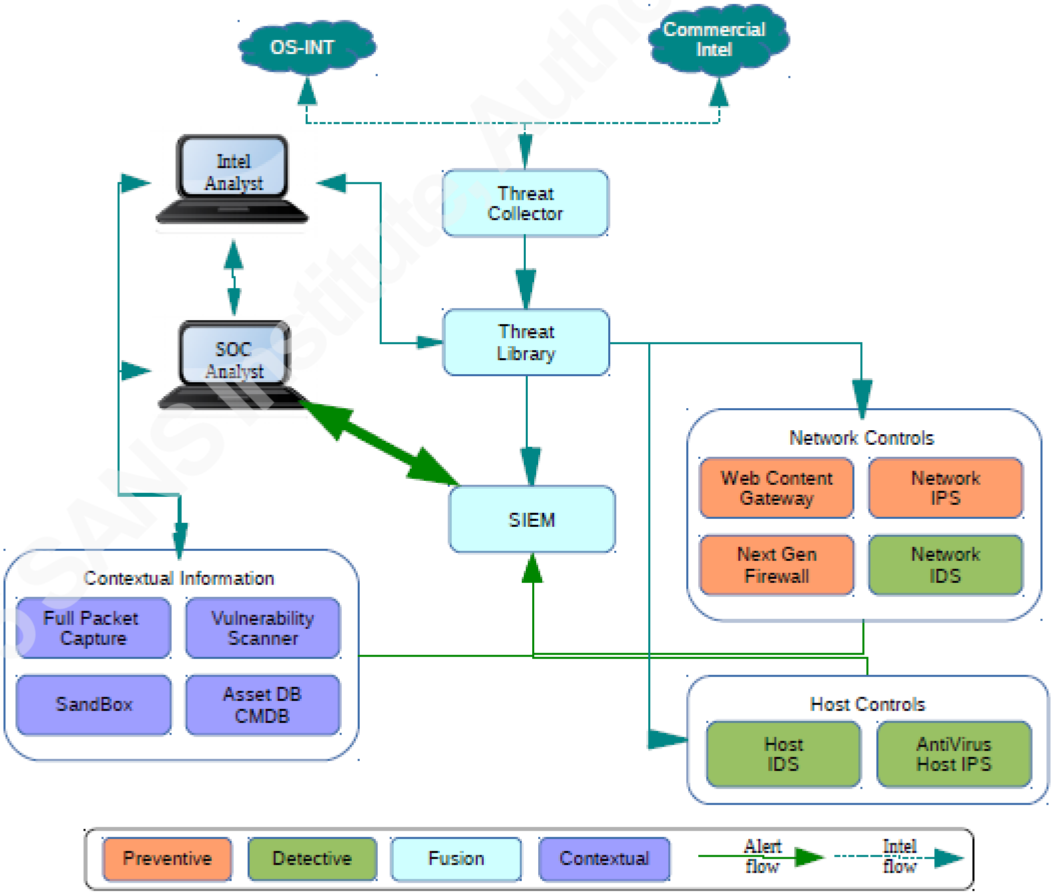


Figure 4: Central Collection and Storage

The Threat Library would manage the observables' life cycle, promoting, demoting, and aging them out based on the new information received about them. The Threat Library will also allow the defenders to manually change the impact and confidence context for indicators, properly document observables, enrich the context of the observables, and share the observables with other organizations.

As the Incident Response, SOC, and Threat Analysis functions are further separated, the communication between teams becomes very important, and especially for the feedback loop that controls the quality of the intelligence and alerts. Contextual information also becomes very important, for both internal and external contexts. By this level of maturity, the organization should not only track basic observables (IP addresses, domains, FQDNs, and hashes) but also TTPs of the adversary and adversary groups.

The organization should also move away from the alert-based, reactive posture into a more proactive, hunting posture. Context tools will be very helpful here be it full packet capture, verbose IDS logs, vulnerability information, asset information, or a SIEM to present some of that data in a very hunter-friendly way. The defenders should be able to distill some of the hunting results into intelligence to be used in the future or into information about compromised hosts.

At this point, a better integration between the SIEM / ESM platform and the Threat Library could benefit an organization by moving towards achieving enhanced automation. The SIEM should be able to automatically extract newer observables from alerts deemed fairly accurate and update the Threat Library.

The intelligence and context infrastructure should not only allow the analysts to make quicker, better decisions about potential compromises but should also allow the defenders to start automating some tasks. It is reasonable to think that, based on event types, the SIEM should be able to start some portions of the Incident Response process, whether it is to create a ticket in the Incident Response tracking systems, issue a cleanup request via the help desk tracker, initiate a deeper scan of the victim computer to try to catch other components of the infection, or simply try to collect more information for the incident responder. In some more extreme cases, it might be advantageous to isolate a host from the trusted network automatically upon detections of destructive malware (like something in Crypto Locker's family).

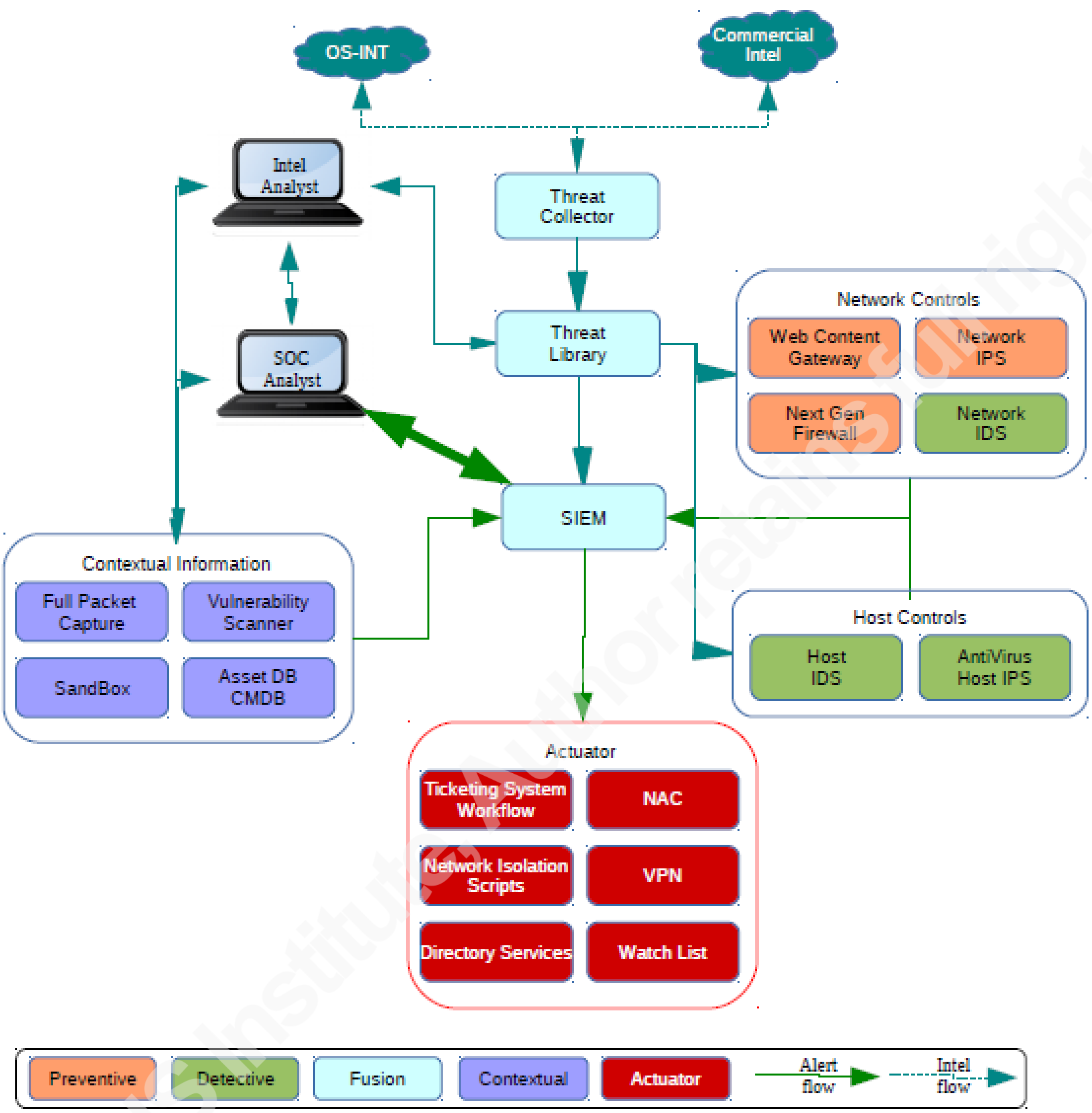


Figure 5: Central Intelligence Gathering, Storage, and context plus automated actions

The major risk associated with using automation for some of the basic Incident Response steps is that a false positive might lead to network disruption. Avoid this is akin to optimizing the SIEM rules: the defenders can start with a few critical attack scenarios and create the logic to stop them, the defenders can focus on high-confidence, high signal-to-noise ratio alerts, or start with less disruptive scripts

The goal of the automation is to make the hunting and Incident Response processes easier and more streamlined so that the analysts can spend more time on important, value-adding tasks instead of the menial parts of the process. Another great benefit of automation is the introduction of consistency to the process and encouragement of innovation and creativity in dealing with other inefficiencies.

In the current commercial space, the standard configuration consists of partial integrations between the Threat Library, the SIEM, and some defensive controls. These integrations are usually centered on one or more lists of basic indicators that are fed into the SIEM. The SIEM generates alerts based on those indicators, encouraging the analysts to research indicator occurrence. This approach is a great start since it provides a great basic step for integration, but too much focus on these basic building blocks might cause the defenders to lose sight of the fact that the further up the Pyramid of Pain the defenders can go, the more difficult successful attacks will become.

Threat Intelligence should become the common information bridge between security controls. In an optimistic future state, the SIEM and the Actuator should be able to take the information gleaned from the alerts and feed it into a feedback loop back into the Threat Library. For example, a Snort alert for Asprox Command and Control traffic should both tell the SIEM what computer has been compromised and is in need of reimaging as well as what foreign IP is connected to the Asprox botnet and which logs associated with it might need a second opinion. The SIEM should then be able to record the foreign IP as malicious and disseminate it to other controls.

Two companies are taking a very similar approach in using Threat Intelligence to enhance their incident response process.

First, Netflix is working on a very interesting project that bypasses the Threat Intelligence library and SIEM concepts and instead focuses on automating the workflow for certain alerts based on a confidence score of the alert, an importance score for the user and computer as well as some fusion capabilities for the observables using Threat Intelligence sources. Their system automates some of the identification, containment, and eradication functions, reducing the time incident resolution. Focusing on immediate context rather than long-term historical context for alerts is an efficient way to eradicate commodity malware (http://www.rsaconference.com/writable/presentations/file_upload/tech-f03a-malware-defense-integration-and-automation-v4.pdf).

Second, GitHub is taking advantage of a chatbot for IR note transcription, Threat

Intelligence context, and rudimentary actions based on commands to the bot. Some of the best features of the chat bot are its availability across platforms and its integration with some defensive measures (https://speakerdeck.com/sroberts/building-your-own-dfir-sidekick-threads-edition).

Conclusion

Properly using Threat Intelligence might help defend against the advanced attacker as signatures by themselves are proving increasingly less useful. By being able to know a bit more about the adversary and to codify that knowledge into some observables and indicators of compromise, defenders can render some of the attackers' infrastructure useless and therefore increase the costs for the attackers.

There are various open source and commercial solutions for storing Threat Intelligence and several formats for sharing it. Availability of tools and integration with existing systems is driving the market to a common set of features, focusing more and more on both detection and Incident Response as the quality of the observables makes is difficult to use them in pure blacklists.

The market for Threat Intelligence is still developing, accounting for the difficulties in assessing the value of a Threat Feed. These difficulties are evident in the great oscillations in price between threat feeds. As the market matures, the prices should normalize and better reflect the quality of the observables offered.

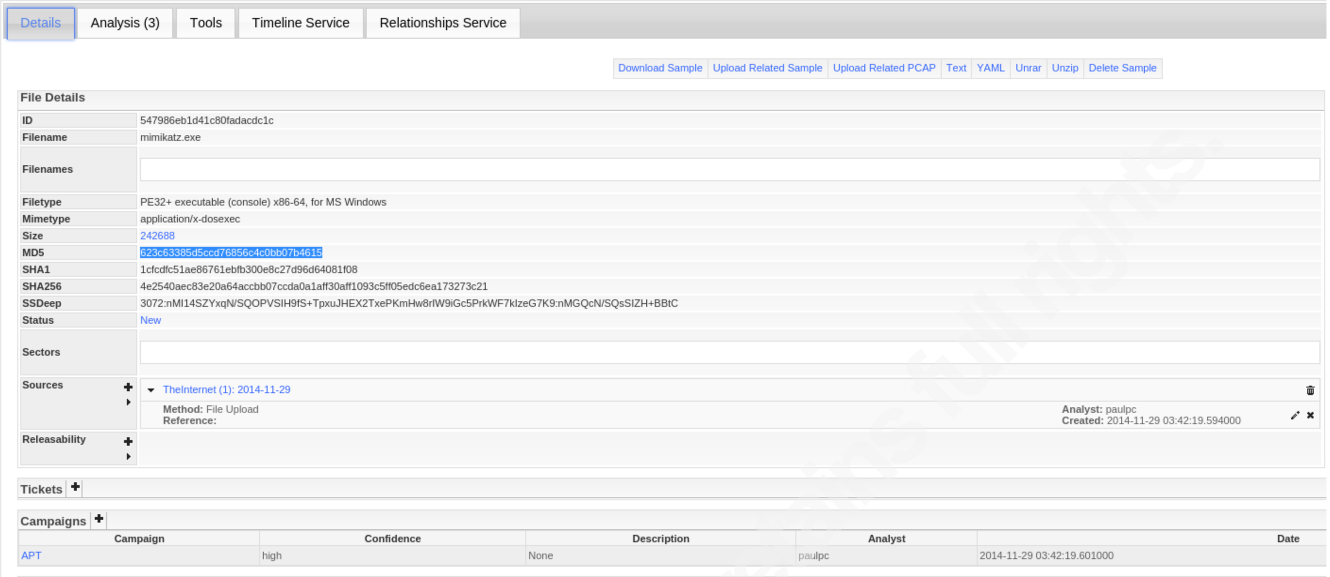
Appendix – Example of Using Threat Intelligence

The easiest way to get started with Threat Intelligence to download some of the Open Source Intelligence (OS-INT) feeds available on the Internet. Combine comes with a few feeds in the inbound and outbound files. Lenny Zeltser blogged an interesting list of feeds (Zeltser Blocklists). By using CIF, or CRITs and Combine, the defender can start creating a collection of observables.

Nyx (https://github.com/paulpc/nyx) is an attempt to create some automation and take this from pure theory to something that can be implemented. This should be seen as a Proof of Concept code more than a plug-and-play program. The focus was on integrating some of the more widely used technologies, including IDS, NG-Firewall, Web Proxy, and SIEM.

The basic observables (IP, FQDN, Hashes) are supported almost across the board in open source and commercial products since they are often seen as de facto Threat Intelligence by security vendors. If one uses the SIEM as the artifact storage device, this should prove to be a simple deployment, consisting of a few watch lists of basic observables. Unfortunately, this approach will limit the defender's capabilities to using intelligence as an enhancement to signatures.

The Threat Library should become the system of reference for the observables as it will allow the defender to track not only the basic observable (623c63385d5ccd76856c4c0bb07b4615) but more data to enrich it as well, including metadata on the file, notes, campaigns using the file, source of the file. CRITs is particularly useful at doing some static analysis on the file. All this information should come in handy when implementing an automated dissemination mechanism when we can use multiple vectors to pick how to best alert on or block a specific piece of information.



File Details

ID	547986eb1d41c80fadacdc1c
Filename	mimikatz.exe
Filenames	
Filetype	PE32+ executable (console) x86-64, for MS Windows
Mimetype	application/x-dosexec
Size	242688
MD5	623c63385d5ccd76856c4c0bb07b4615
SHA1	1cfcdfc51ae86761ebfb300e8c27d96d64081f08
SHA256	4e2540aec83e20a64accbb07ccda0a1aff30aff1093c5ff05edc6ea173273c21
SSDeep	3072:nMI14SZYxqN/SQOPVSIH9fS+TpxuJHEX2TxePKmHw8rIW9iGc5PrkWF7klzeG7K9:nMGQcN/SQsSIZH+BBtC
Status	New
Sectors	
Sources	TheInternet (1): 2014-11-29 Method: File Upload Reference: Analyst: paulpc Created: 2014-11-29 03:42:19.594000
Releasability	

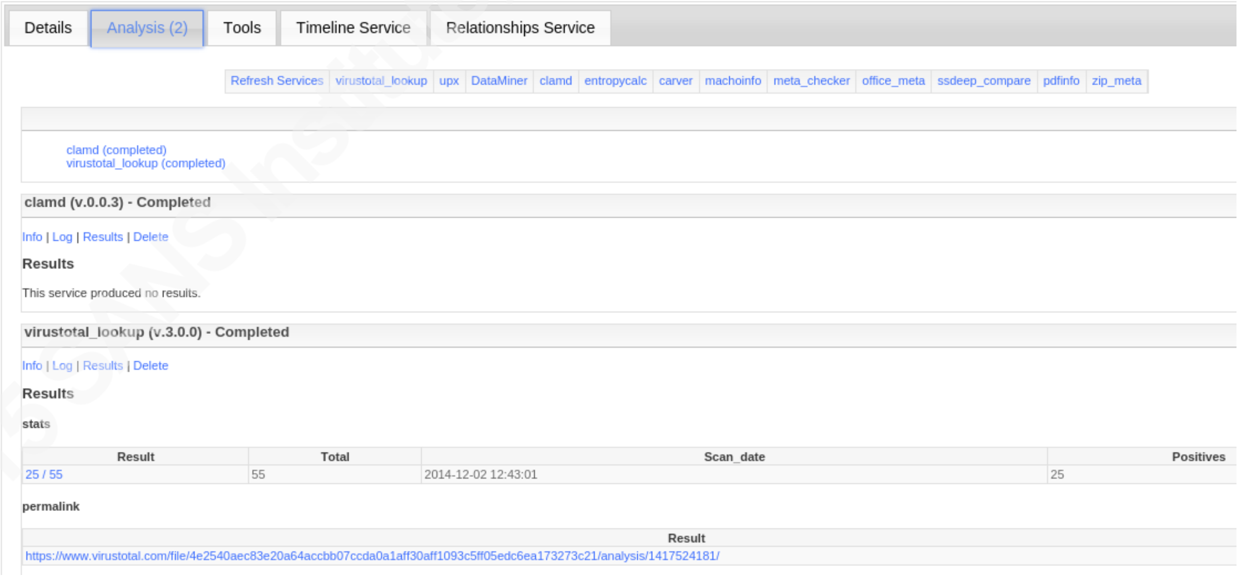
Tickets

Campaigns

Campaign	Confidence	Description	Analyst	Date
APT	high	None	paulpc	2014-11-29 03:42:19.601000

Figure 6: CRITs observable details

Threat Intelligence Libraries can offer some analytic help. CRITs has a wide range of services that facilitate a cursory static analysis of the observables by reaching out to services like VirusTotal, by facilitating strings and XOR analysis, or by the relationship services.



clamd (completed)
virustotal_lookup (completed)

clamd (v.0.0.3) - Completed

Info | Log | Results | Delete

Results

This service produced no results.

virustotal_lookup (v.3.0.0) - Completed

Info | Log | Results | Delete

Results

stats

Result	Total	Scan_date	Positives
25 / 55	55	2014-12-02 12:43:01	25

permalink

Result
https://www.virustotal.com/file/4e2540aec83e20a64accbb07ccda0a1aff30aff1093c5ff05edc6ea173273c21/analysis/1417524181/

Figure 7: CRITs Analysis Services

The relationship service provides the ability to tie together observables in order to attempt a high-level narrative of the adversaries' tactics, techniques, and procedures, effectively moving the controls towards the higher steps on the Pyramid of Pain. This also helps in the fusion function by distilling the indicators to achieve a better quality and by facilitating finding similar observables.

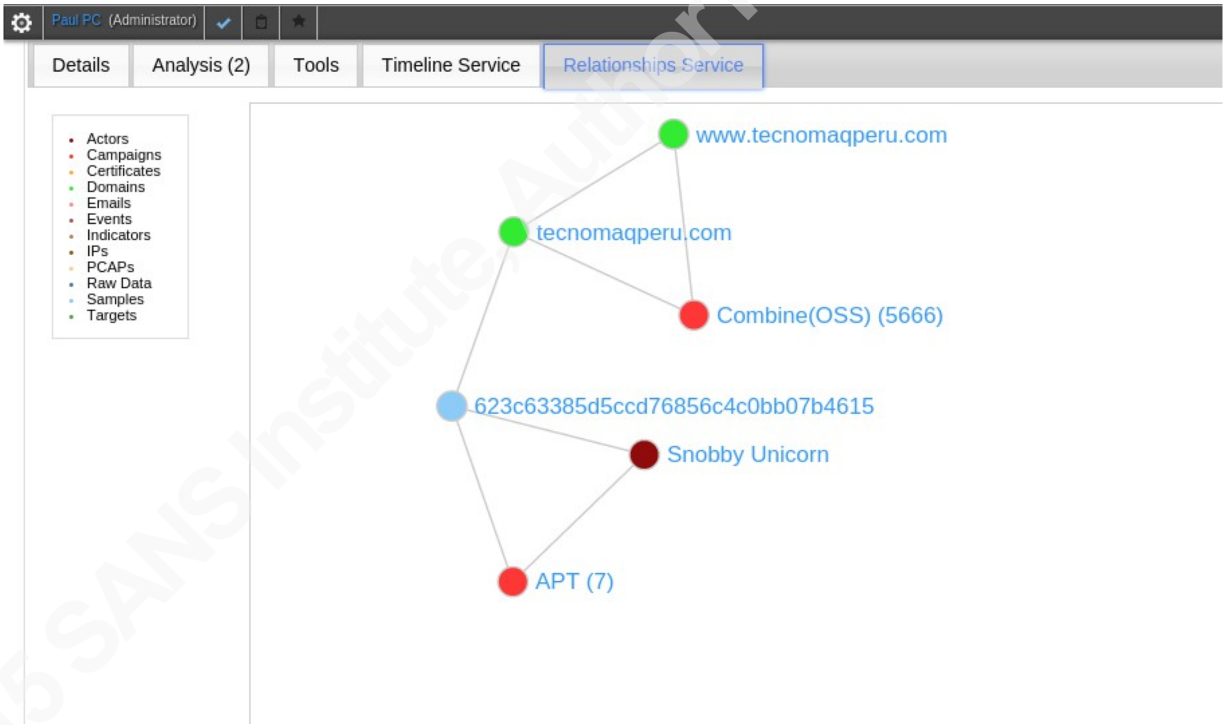


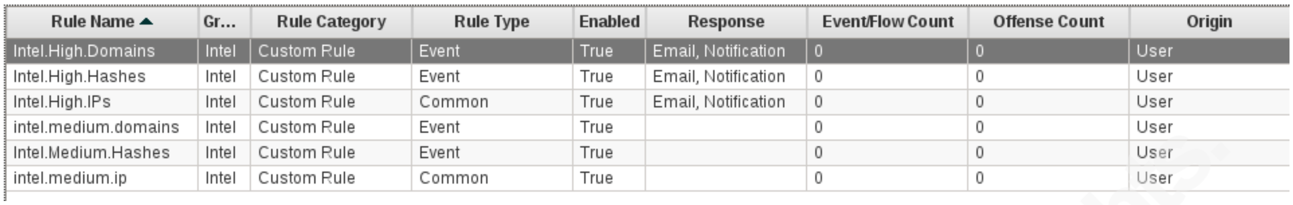
Figure 8: Relationships in CRITs

Because a SIEM collects logs from the security devices across the enterprise, it seems to be the best place to look for Indicators of Compromise. It is also the perfect place to look for past activity connected to the observables. Where it falls short is its limitation to low-level matching for the observables (MD5 hashes versus SSDEEP hashes) and the limited default context (adversary, campaign, and victim tracking require customization).

Fortunately, the context is easy to improve by taking advantage of some right-click context options. For example, searching IP addresses in VirusTotal, TrustedSource, Theath Library, or the ISC sites is rather trivial. To enhance this, it makes the most sense to employ Reference Sets (watch lists) as a local repository for the observables. Most of these observables will have a fairly low confidence (trivial to medium), and therefore might make poor alerts due to the high false positive rate, but they might make a great alert relevance enhancer. Rather than create a new alert based on these low-fidelity observables, the defenders can use them to raise the importance of the alerts that are currently triggering from the signatures implemented (e.g. a malware deleted alert from the host antivirus is a common occurrence, but it is worth a second look if the MD5 designates it as fgdump, a tool used by some of the adversaries). Other indicators might require immediate analysis triggering based on high-fidelity observables. For this purpose, the following reference sets should be a base for different alert actions:

- Intel.Attacker.Emails
- Intel.High.Domains
- Intel.High.Hashes
- Intel.High.Ips
- Intel.Medium.Domains
- Intel.Medium.Hashes
- Intel.Medium.Ips
- Intel.Target.Emails
- Intel.Targets

The medium sets are used for enhancing the current alerts based on observables being part of the alert. The high sets are triggering an alert automatically. The email alerts are centered on both targets of advanced attacks as well as some of the known attacker infrastructure.



Rule Name ▲	Gr...	Rule Category	Rule Type	Enabled	Response	Event/Flow Count	Offense Count	Origin
Intel.High.Domains	Intel	Custom Rule	Event	True	Email, Notification	0	0	User
Intel.High.Hashes	Intel	Custom Rule	Event	True	Email, Notification	0	0	User
Intel.High.IPs	Intel	Custom Rule	Common	True	Email, Notification	0	0	User
intel.medium.domains	Intel	Custom Rule	Event	True		0	0	User
Intel.Medium.Hashes	Intel	Custom Rule	Event	True		0	0	User
intel.medium.ip	Intel	Custom Rule	Common	True		0	0	User

Figure 9: QRadar intel-based rules

The SIEM's efficacy is contingent on the quality of the alert data. It is important to offload the appropriate controls to the system that can best handle them. For example, an array of Bro sensors monitoring traffic going in and out of the network might be more efficient than having the SIEM look for the all the intel-related IP addresses in every Firewall, Web Proxy, IDS, DNS, and End Point log entry. Therefore, one of the assumptions of this model is a 'monitoring-in-depth' approach, with a minimum of network monitoring with IDS of all outbound and inbound traffic (at the border), Web Proxy, a Next Generation Firewall, and some End Point visibility. The more tools that output useful data, the easier the Incident Response process, and the better the Threat Intelligence fusion.

The focus of this paper is on integrating the following commercial and open source tools in order to facilitate the dissemination of threat intelligence:

Product	**Role**	**Observables**
Combine	Collect OS-INT	IP, FQDN
CRITs	Threat Library – store Threat Intelligence observables	IP, FQDN, file metadata, targets, emails
QRadar	SIEM – log aggregation, alert correlation	IP, FQDN, hashes, filenames, email addresses, userid,
Palo Alto	Next-Generation Firewall	FQDN, IP, userid
BRO IDS	Intrusion Detection, Network Security Monitoring	IP, FQDN, MD5, User Agent String
Generic Web proxy	Web content gateway	FQDN, IP, userid
Generic Sandbox	Analyze binaries and output results to SIEM	MD5, filename
Host IDS	Host intrusion detection – record binaries ran on end-points and report to SIEM	MD5, filename, IP, FQDN
Generic Email Gateway	Stop SPAM, send email records to SIEM	Email addresses, subject,

Table 4: Security Tools Intel Integrations

Some of the tools are specific (QRadar as the SIEM, Palo Alto as the Next Generation Firewall, CRITs as the Threat Library, and Bro as the IDS) since the scripting depended on their availability and APIs. That is not necessarily an appreciation of the quality of the tools but a coincidence of availability and API capabilities. The methods highlighted here should apply to any other tools that provide a decent API interface.

The flow of the observables should look something like this:

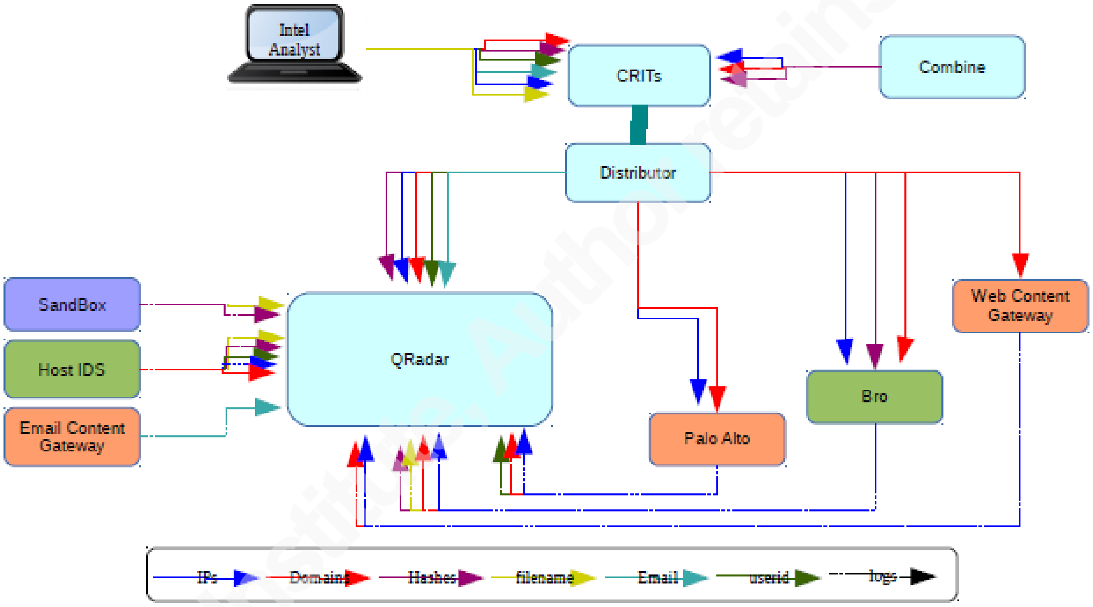


Figure 10: IOC flow in the monitoring ecosystem

For the purpose of this exercise, the intelligence sources are OS-INT sources that are set up by default in Combine. After Combine collects the artifacts, it uploads them into CRITs using the API, populating the source field with the origin of the indicators, the campaign with Combine, and a confidence level of 'medium'.

The observables will be read by the distribution engine, Nyx, from CRITs (along with any enhancements made by the threat analysts) and disseminated to the detective and preventive technologies. Those systems will then feed the alerts back into the SIEM, which will alert on the high-fidelity observables and allow the defenders to hunt though the enriched logs.

Before disseminating the observables, Nyx tries to determine the confidence associated with the observable by looking at the associated campaign. It will allow a couple of methods for upgrading the confidence:

- If an indicator appears to be connected to multiple sources, the likelihood of it being a quality indicator should go up under the assumption that multiple independent parties have discovered it.
- The confidence of the indicator should be at least as high as the highest campaign confidence.

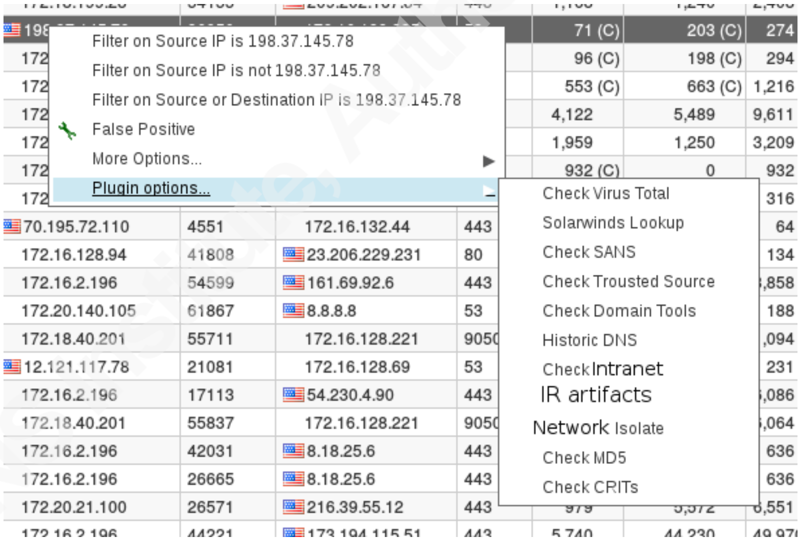
Nyx currently looks at five categories of Threat Intelligence from CRITs: IP addresses, domains, samples, emails, and targets. From these high-level categories, Nyx will load the low-level observables: (IP address, domain name, MD5 hash, file name, email address, userID) and disseminate them to the appropriate systems:

- **Bro**: The IP addresses, domains, MD5 hashes, and file names will be placed in a text file and made available on a web server. Crontab starts a script on the Bro Manager to download the text file. A Bro script will load the file in the Threat Intelligence Library.
- **Palo Alto**: Nyx will send the Domains and IPs to the Palo Alto API. The domains would be put in two custom categories, one focused on blocking the high-fidelity observables, the other one for alerting. Only the high-confidence IP addresses will be uploaded to Palo Alto since it is used more as a blocking tool.
- **Web Proxy**: The high-fidelity domains will be placed in a text file on the web server. Most web filters are able to load a flat file and use the domains into a custom blocking category.
- **QRadar**: Nyx will send IP addresses, domains, MD5s, email addresses, and userIDs to the QRadar API to be stored in reference sets based on the campaign confidence.

After the intelligence is processed by the detective and preventive systems, they send alerts to the SIEM. QRadar alerts for the high-confidence and blocking alerts. Based on the lower confidence reference sets, QRadar increases the relevance for events that contain pieces of Threat Intelligence, helping them bubble up to the top to increase the likelihood they will receive attention during the hunting activities.

Threat Intelligence enhances the hunting experience with both OS-INT sources (SANS Trusted Source, Domain Tools, VirusTotal, historic DNS), internal context (IP address management, intranet) as well as search CRITs for the IP address.

The right-click menu can also be used to deploy scripts that collect DFIR artifacts. This should decrease the manual tasks in the Identification phase, create a consistent, repeatable methodology, and facilitate an easier way to do triage. Once a severe incident has been declared, the right-click menu offers a convenient way to isolate infected machines from the network.



198				71 (C)	203 (C)	274
172				96 (C)	198 (C)	294
172				553 (C)	663 (C)	1,216
172				4,122	5,489	9,611
172				1,959	1,250	3,209
172				932 (C)	0	932
172						316
70.195.72.110	4551	172.16.132.44	443			64
172.16.128.94	41808	23.206.229.231	80			134
172.16.2.196	54599	161.69.92.6	443			,858
172.20.140.105	61867	8.8.8.8	53			188
172.18.40.201	55711	172.16.128.221	9050			,094
12.121.117.78	21081	172.16.128.69	53			231
172.16.2.196	17113	54.230.4.90	443			,086
172.18.40.201	55837	172.16.128.221	9050			,064
172.16.2.196	42031	8.18.25.6	443			636
172.16.2.196	26665	8.18.25.6	443			636
172.20.21.100	26571	216.39.55.12	443	979	5,572	6,551
172.16.2.196	44221	173.194.115.51	443	5,740	44,230	49,97

Figure 11: QRadar Context Menu

A great enhancement to the already existing scripts was creating automated actions based on the high-fidelity alerts and automatically launching the Incident Response scripts. We are working on the ability to restrict access to critical data for users or IPs based on alerts triggered on that machine. A computer with anti-malware detections for a Banking Trojan should not be able to access the company financial or accounting applications. A computer displaying signs of CryptoLocker should have restricted access to network drives.

While this approach is biased towards the specific tools mentioned, the underlying methodology should make it easy to extend to other tools with an API. If CIF was to be the library of choice, it has an automatic way of exporting indicators to Snort signatures that may be directly imported into IDS sensors at the border. If Splunk was to be used instead of QRadar, the Palo Alto folks wrote some impressive integrations (https://github.com/PaloAltoNetworks-BD/SplunkforPaloAltoNetworks). While QRadar provided a good platform for right-click integrations, some Threat Intelligence vendors are providing a similar functionality via browser extensions. Some notable examples are CIF, ThreatStream, and VirusTotal.

The biggest challenge will continue to be moving away from the basic disposable observables (IP addresses, domains, hashes) and towards more high-level intelligence (automated tool identification or attribution-based detection). Only when we force our adversaries to constantly reinvent their techniques and recreate their infrastructure will we have a chance to lose fewer cyber battles.

Further Research

A next step for this is to ingest more observables. Specifically, CRITs allows another class of observables called Indicator. It allows for more types of observables and a more detailed confidence and impact rating. If the downstream systems will be able to better ingest and utilize the observable, the level of detail would help by better expressing attacker TTPs.

References:

Bianco, David. Enterprise Security Monitoring. Retrieved 12/1/2014 from Speaker Deck. https://speakerdeck.com/davidjbianco/enterprise-security-monitoring-comprehensive-intel-driven-detection. Enhanced with the video. https://www.youtube.com/watch?v=SVKcFhyGqcY

Bianco, David. On Misuse of Indicators. Retrieved 12/1/2014 from blogspot.com. http://detect-respond.blogspot.com/2013/07/on-misuse-of-indicators.html

Bianco, David. The Pyramid of Pain. Retrieved 12/1/2014 from blogspot.com. http://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html

Bianco, David. Use of Term Intelligence at RSA. Retrieved 12/1/2014 from Blogspot. http://detect-respond.blogspot.com/2014/03/use-of-term-intelligence-at-rsa.html

Bianco, David, What Do You Get When You Cross a Pyramid With A Chain? Retrieved 12/1/2014 from Blogspot. http://detect-respond.blogspot.com/2013/03/what-do-you-get-when-you-cross-pyramid.html

Brady, Sean. Building a Threat Intelligence GamePlan. Retrieved 12/1/2014 from RSA GlobalSummit. http://globalsummit.rsa.com/wp-content/uploads/2015/09/Threat-Intelligence-Automation-And-Sharing-In-Security-Analytics-Environments.pdf

Chuvakin, Anton. How to Collect, Refine, Utilize and Create Threat Intelligence, Retrieved 10/9/ 2014 from Gartner. http://www.gartner.com/document/2738618

Chuvakin, Anton. Threat Assessment in the Age of the APT, Retrieved 10/9/2014 from Gartner. http://www.gartner.com/document/2738617

Cyber Squared. Threat Intelligence Platforms. Retrieved 12/1/2-14 from Threat Connect. http://www.threatconnect.com/guide-to-threat-intelligence-platform/

Fry, Rob. Malware Defense and Automation: Fully Integrated Defense Operation (F.I.D.O.). Retrieved on 12/1/2014 from RSA Conference. http://www.rsaconference.com/writable/presentations/file_upload/tech-f03a-malware-defense-integration-and-automation-v4.pdf

Grobauer, B. & Berger, S. & Göbel, J. & Schreck, T. & Wallinger, J. The MANTIS Framework Cyber-Threat Intelligence Mgmt. for CERTs. Retrieved 12/1/2014 from FIRST. http://www.first.org/resources/papers/conference2014/first_2014_-

_grobauer-_bernd_-_mantis_framework_20140606.pdf

Holland, Rick. Use Actionable Threat Intelligence to Protect Your Digital Business. Retrieved from Forrester. https://www.forrester.com/Use+Actionable+Threat+Intelligence+To+Protect+Your+Digital+Business/fulltext/-/E-RES118032

Hutchins, Eric M. & Cloppert, Michael J. & Amin, Rohan M., Ph.D. Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. Retrieved 12/1/2014 from Lockheed Martin. http://www.lockheedmartin.com/content/dam/lockheed/data/corporate/documents/LM-White-Paper-Intel-Driven-Defense.pdf

Kornblum, Jesse, Identifying almost identical files using context triggered piecewise hashing. Retrieved 12/1/2014 from Digital Investigations 3S. http://dfrws.org/2006/proceedings/12-Kornblum.pdf

Los, Rafal & Robinson, James & Brooks, Rob & Brown, Woodrow. Solution Primer Threat Intelligence. Retrieved 12/1/2014 from Accuvant. http://files.accuvant.com/web/file/d96f8c4996ee4571999bcf513126399c/Threat%20Intelligence_Solution%20Primer.pdf

Mandiant Inc. Tracking Malware with Import Hashing. Retrieved 12/1/2014 from Mandiant (FireEye). https://www.mandiant.com/blog/tracking-malware-import-hashing/

McMillan, Rob. Definition: Threat Intelligence. Retrieved 10/28/2014 from Gartner. https://www.gartner.com/doc/2487216?ref=SiteSearch&sthkw=G00249251

NCI Agency. Malware Information Sharing Platform. Retrieved 12/1/2014 from NATO. http://www.ncia.nato.int/Documents/Agency%20publications/Malware%20Information%20Sharing%20Platform%20(MISP).pdf

Pinto, Alex & Maxwell, Kyle: Measuring the IQ of your Threat Intelligence Feeds. Defcon 2014. https://www.youtube.com/watch?v=yG6QlHOAWiE

Roberts, Scott. A Basic Guide to Incident Response. Retrieved 12/1/2014 from github.io. http://sroberts.github.io/2014/05/07/a-basic-guide-to-advanced-incident-response/

Roberts, Scott. Building Your Own DFIR Sidekick. Retrieved 12/1/2-14 from speakerdeck.com. https://speakerdeck.com/sroberts/building-your-own-dfir-sidekick-threads-edition

Vorstack. Developing a Threat Intelligence Game Plan. Retrieved from Vorstack. http://www.isightpartners.com/wp-content/uploads/2014/07/iSIGHT_Partners_What_Is_20-20_Clarity_Brief1.pdf

1.1.5. Code:

Pinto, Alex & Maxwell, Kyle. combine: https://github.com/mlsecproject/combine

Goffin, Mike & Shields, Wesley. CRITs: https://github.com/crits

Grobauer, Brend. Django-mantis: https://github.com/siemens/django-mantis

Poputa-Clean, Paul. nyx: https://github.com/paulpc/nyx

1.1.6. Appendix References:

Manahan, Peter. QRadar API samples: https://github.com/ibm-security-intelligence/api-samples/tree/7.2.3

Palo Alto Networks. PAN-OS and Panorama XML API Reference Guide 6.0. Retrieved 12/2/2014 from Palo Alto Networks. https://live.paloaltonetworks.com/docs/DOC-6607

Schipp, Jon. Intelligence Data and Bro. Retrieved 12/2/2014 from Blogspot. http://blog.bro.org/2014/01/intelligence-data-and-bro_4980.html

Torres-Gil, Brian. Splunk for Palo Alto Networks App. Retrieved 12/2/2014 from GitHub https://github.com/PaloAltoNetworks-BD/SplunkforPaloAltoNetworks

US-CERT, Traffic Light Protocol (TLP) Matrix and Frequently Asked Questions, Retrieved 1/6/2015 from https://www.us-cert.gov/tlp

Zeltser, Lenny. Blocklists of Suspected Malicious IPs and URLs. Retrieved 12/1/2014 from zeltser.com. http://zeltser.com/combating-malicious-software/malicious-ip-blocklists.html

1.1.7. Threat Intelligence Languages:

- openIOC: http://openioc.org/
- STIX: https://stix.mitre.org/
- cybox: http://measurablesecurity.mitre.org/docs/cybox-intro-handout.pdf
- IODEF: http://xml.coverpages.org/iodef.html

  - http://www.ietf.org/rfc/rfc5070.txt

1.1.8. Commercial Offerings:

- Tactical Intelligence
  - Norse: https://www.norse-corp.com/
  - ThreatGrid: http://www.threatgrid.com/
  - LookingGlass: http://www.lgscout.com/
- Tactical and Strategic Intelligence
  - iSight: http://www.isightpartners.com/
  - Mandiant: https://www.mandiant.com/
  - CrowdStrike: http://www.crowdstrike.com/
  - Verisign iDefense: https://www.verisigninc.com/en_US/cyber-security/security-intelligence/threat-intelligence/index.xhtml
  - SenseCy: https://www.sensecy.com/
- Threat Libraries
  - ThreatStream: http://threatstream.com/
  - ThreatQuotient: http://www.threatquotient.com/
  - ThreatConnect: http://www.threatconnect.com/
- Threat Intel Automation
  - Vorstack: https://vorstack.com/
  - McAfee TIE: http://www.mcafee.com/us/products/threat-intelligence-exchange.aspx