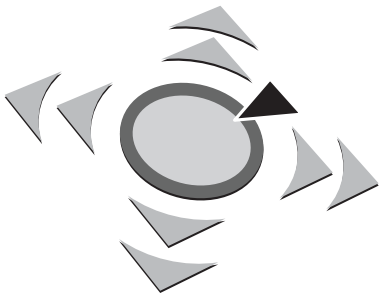
10. GI FG SIDAR Graduierten-Workshop über Reaktive Sicherheit

SPRING

Matthias Wübbeling (Hrsg.)

02.-03. Juli 2015, Neubiberg



SIDAR-Report SR-2015-01
ISSN 2190-846X

Vorwort

SPRING ist eine wissenschaftliche Veranstaltung im Bereich der Reaktiven Sicherheit, die Nachwuchswissenschaftlern die Möglichkeit bietet, Ergebnisse eigener Arbeiten zu präsentieren und dabei Kontakte über die eigene Universität hinaus zu knüpfen. SPRING ist eine zentrale Aktivität der GI-Fachgruppe SIDAR, die von der organisatorischen Fachgruppenarbeit getrennt stattfindet. Die Veranstaltung dauert inklusive An- und Abreise zwei Tage und es werden keine Gebühren für die Teilnahme erhoben. SPRING findet ein- bis zweimal jährlich statt. Die Einladungen werden über die Mailingliste der Fachgruppe bekanntgegeben. Interessierte werden gebeten, sich dort einzutragen (`http://www.gi-fg-sidar.de/list.html`). Für Belange der Veranstaltung SPRING ist Ulrich Flegel Ansprechpartner innerhalb der Fachgruppe SIDAR. Inzwischen blickt die SPRING auf zehn erfolgreiche Veranstaltungen zurück; angefangen mit der Premiere in Berlin über Veranstaltungen in Dortmund, Mannheim, Stuttgart, Bonn, München, Bochum und Neubiberg. Die Beiträge in diesem Jahr decken ein breites Spektrum ab, von noch laufenden Projekten, die ggf. erstmals einem breiteren Publikum vorgestellt werden, bis zu abgeschlossenen Forschungsarbeiten, die zeitnah auch auf Konferenzen präsentiert wurden bzw. werden sollen oder einen Schwerpunkt der eigenen Diplomarbeit oder Dissertation bilden. Die Abstracts sind in diesem technischen Bericht zusammengefasst. Der Bericht ist elektronisch, zitierfähig und recherchierbar veröffentlicht. Der Bericht ist über das Internet-Portal der Fachgruppe SIDAR (`http://www.gi-fg-sidar.de/`) verfügbar. Der Bericht und die zugehörigen Präsentationsfolien sind über die Veranstaltungs-Webseite (https://itsec.cs.uni-bonn.de/spring2015/) verfügbar. In dieser Ausgabe finden sich Beiträge zu den folgenden Themen: Intrusion Detection, Threat Intelligence, Malware, Phishing, IT-Sicherheitsbewusstsein und Incident Response.

Ich danke dem Gastgeber, der Infineon Technologies AG, sowie allen Autoren und Teilnehmern der 10. SPRING.

Neubiberg/Bonn, Juli 2015 Matthias Wübbeling

Contents

Diesen Bericht zitieren als:
Matthias Wübbeling, editor. Proceedings of the 10. GI SIG SIDAR Graduate Workshop on Reactive Security (SPRING). Technical Report SR-2015-01, GI FG SIDAR, ISSN 2190-846X, Neubiberg/Bonn, Juli 2015,

Einzelne Beiträge zitieren als:
Autor. Titel. In Matthias Wübbeling, editor, Proceedings of the 10. GI SIG SIDAR Graduate Workshop on Reactive Security (SPRING). Technical Report SR-2015-01, page xx. GI FG SIDAR, ISSN 2190-846X, Neubiberg/Bonn, Juli 2015.

Behavior-Driven Development in Malware Analysis: Can it Improve the Malware Analysis Process?

Thomas Barabosch
Fraunhofer FKIE
Friedrich-Ebert-Allee 144
53113 Bonn, Germany
thomas.barabosch{at}fkie.fraunhofer.de

Malware analysis is the process of understanding the behavior of a malware. The reimplementation of malicious behavior is a fundamental part of the malware analysis process. Such behaviors include the domain generation algorithm (DGA) to predict future rendezvous points or the communication protocol to track the malware's command and control infrastructure. The malware sample at hand already implements the system specifications, albeit heavily obfuscated. In a perfect world, an analyst would hypothesize and would corroborate hypotheses continuously in order to derive the malware's system specifications. As of now analysts merely translate assembler code into an arbitrary high-level language without ensuring the translated code's correctness or its readability.

When reimplementing a behavior, or scientifically speaking corroborating an hypothesis, the analyst faces several obstacles. These obstacles include finding errors in a reimplementation that reimplements faulty code or keeping the code base slim and understandable for its integration in other software systems. These obstacles ask for a different way of how malware analysts face the reimplementation task and the malware analysis process as a whole. Modern malware analysis processes [1] already benefit from modern product development processes like SCRUM. But so far the malware analysis community has not widely embraced these processes. We therefore believe that it is important to think out of the box and to apply modern software development processes to malware analysis as well.

Behavior-Driven Development (BDD) is a software development process that emerged from Test-Driven Development (TDD) in the late 2000s. As in TDD, a programmer runs through a short development cycle in BDD. This cycle consists of three stages. Firstly, the programmer specifies a behavior as a test. Secondly, he writes the minimum of code to implement this behavior and to pass the test. Thirdly, he refactors the code to simplify it and also to meet code standards. These short development cycles ensure a high confidence in the correctness of the code, a slim code base and less debugging of the code.

In this talk, we discuss how BDD can enhance the malware analysis process. We give answers to questions such as

- How can we integrate BDD into the malware analysis process?
- What are the advantages and drawbacks of BDD malware analysis?
- What are the limitations of BDD malware analysis?

We conclude our talk with a case study on Nymaim. We show how we analyzed Nymaim's DGA by applying the principles of BDD to malware analysis.

References

[1] Daniel Plohmann, Sebastian Eschweiler and Elmar Gerhards-Padilla: *Patterns of a Cooperative Malware Analysis Workflow*
5th International Conference on Cyber Conflict, Tallinn (Estonia), 2013

Apate Interpreter - A Kernel Hook Rule Engine

Christoph Pohl*†, Hans-Joachim Hof* and Michael Meier†

* University of Applied Sciences Munich
Muse - Munich IT-Security Research Group
pohl2,hof{at}hm.edu

†Fraunhofer FKIE
Universität Bonn
mm{at}cs.uni-bonn.de

Honeypots are used in IT Security to detect and gather information about ongoing intrusions, e.g. by documenting the approach of an attacker. Honeypots do so by presenting an interactive system that seems just like a valid application to an attacker. This paper presents a part of APATE, a Linux Kernel Module (LKM) that is able to log, block and manipulate system calls based on preconfigurable conditions like Process ID (PID), User Id (UID), and many more. APATE can be used to build and harden High Interaction Honeypots. APATE can be configured using an integrated high level language. This language gets compiled to a fast intermediate language, which is processed by the APATE interpreter. This interpreter is able to hook, manipulate and log most of Linux Kernel functions. This research presents the language, the interpreter and its performance evaluation.

How IT Security Awareness Should be Tested

Arnold Sykosch*† and Matthias Wübbeling*†

* University of Bonn
Working Group IT Security

†Fraunhofer FKIE
Cyber Security Department

The first thing that has to be recognized it the fact, that no technical system exists for a self serving purpose. Every technology serves the user. That is also true for any part of a modern IT infrastructure. While the user follows his task, he has a certain understanding of the situation at hand, he is aware. Awareness might be described as „the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future“[1].

IT security awareness is situation awareness limited to elements directly or indirectly related to IT security. These elements are specifically security related when they are a result of or manipulated by an ongoing cyber attack. We call these objects, items, events or effects that are artificially constructed, observable and part of or caused by an attack *artifacts*. These artifacts are the interface between the user and and the actor of a cyber attack.

To measure a user's IT security awareness the user is exposed to a simulated cyber attack during his daily business and the users reaction to exposure is recorded. These reactions may be detrimental to an ongoing attack e.g. the user reports a phishing attempt or beneficial for the attacker e.g. the user clicks the link provided in the phishing mail. The user recognizes the artifacts as elements in his situation and reacts to them.

This approach delivers results that are influenced by the users preference to accomplish his primary goal without any possible distraction and unbiased by effects that accompany methods within a confined environment, e.g. questionnaires, interviews or tests in a laboratory, from effects like social desirability bias [2].

This approach is followed when phishing tests are conducted [3]. But situations where the user is not primarily involved are usually not simulated. The project ITS.APT [4] closes this gap.

References

[1] Mica R. Endsley: *Toward a Theory of Situation Awareness in Dynamic Systems* - Human Factors: The Journal of the Human Factors and Ergonomics Society, 1995.

[2] Lee J. Cronbach: *Response Sets and Test Validity* - Educational and Psychological Measurement, 6(4), 475-494, 1946.

[3] Dhamija, Rachna, J. Doug Tygar, and Marti Hearst: *Why phishing works* - Proceedings of the SIGCHI conference on Human Factors in computing systems. ACM, 2006.

[4] `https://itsec.cs.uni-bonn.de/itsapt` (German only)

A graph model for incident analysis

Christian Kollee
Fraunhofer FKIE
53343 Wachtberg, Germany
christian.kollee{at}fkie.fraunhofer.de

Todays society depends on the reliability and security of computer networks. News showed that the security of those systems is not in a good shape. Prominent examples like Sony, the U. S. Office of Personal Management or the Bundestag back the assumption that prevention eventually fails [1]. Especially a well-prepared attacker will circumvent preventive measures with ease using sophisticated attacks. Therefore, detection and analysis abilities are required to react to such attacks. This gets even more necessary as the average time from compromise to detection is 205 days according to the latest Mandiant M-Trends™ [2].

The biggest challenge is to detect all attacks and at the same time reducing the number of false positives to avoid that the analyst is flooded by alert messages. But in most cases the analyst will be required to process too many events. Therefore, the main objective has to be to provide the human with enough information to decide quickly if an event is a real incident or not. Security information and event management systems (SIEMs) are build for that purpose. They primarily consume syslog and netflow information and do not integrate well with the diverse data sources needed by security analysts. Providing the analyst with a model of these different data sources and the relationships between them could enhance the incident analysis.

In this talk, we present our first approach for such a model using property graphs. Property graphs are a general-purpose tool to model entities and their relationships which is particular suitable for modeling (computer & communication) networks. In contrast to other approaches, like attack trees or the kill chain models [3], we do not aim to model the unknown and unpredictable attacker but the structure of the defendable system and its current state, consisting of all communication and occuring events. Using graph query languages, like SPARQL [4] or Cypher [5], instances of such a model can be queried to support the analysts understanding of the defendable system and its current state which helps them to assess possible incidents.

References

[1] Chris Sanders, Jason Smith, David J. Bianco, and Liam Randall. *Applied Network Security Monitoring: Collection, Detection, and Analysis*. Syngress, 2013.

[2] Mandiant. M-Trends™ 2015: A view from the front lines. Technical report, 2015.

[3] Eric M. Hutchins, Michael J. Cloppert, and Rohan M. Amin. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. Technical report, Lockheed Martin Corporation, 2011.

[4] S. Harris and A. Seaborne. SPARQL 1.1 query language. Technical report, W3C, 2013.

[5] Ian Robinson, Jim Webber, and Emil Eifrem. *Graph Databases – New Opportunities for Connected Data*. O'Reilly Media, 2015.

Sicherheitsanalyse der TLS-Konfiguration von SMTP-Installationen

Thomas Maier
Hochschule München
D-80335 München
tmaier{at}fs.cs.hm.edu

Um eine Aussage über die Sicherheit von TLS bei SMTP zu treffen, wurden 50.000 Mail-Server-Konfigurationen aktiv gescannt und analysiert. Bei der Datenerhebung wurde eine Vielzahl von TLS-Verbindungen zu Mail-Servern aufgebaut, um zu prüfen, wie gut die Sicherheitsziele Authentizität, Integrität und Vertraulichkeit im Mail-Verkehr umgesetzt werden. Zusätzlich wurde jedes Zielsystem auf die Heartbleed-Schwachstelle geprüft. Der Vortrag beschreibt die Datenerhebung, die Analyse und dadurch gewonnene Erkenntnisse.

Wie sicher TLS bei der Übertragung von Mails per SMTP implementiert wurde, ist nicht bekannt. Daher wurde in dieser Arbeit die TLS-Konfiguration von SMTP-Servern geprüft, um eine repräsentative Aussage über den aktuellen Stand der Sicherheit in Hinsicht auf den Transport von Mails zu treffen. Um diese Analyse zu ermöglichen, wurde im ersten Schritt ein Werkzeug entwickelt, mit dem anschließend TLS-Konfigurationen im Internet gesammelt werden. Im zweiten Schritt wurden die bei der Datenerhebung gesammelten Daten analysiert und ausgewertet.

Zur Datensammlung wurden alle deutschen IP Ranges nach den Ports 25, 465 und 587 durchsucht. Anschließend wird die Software SSLyze [1] verwendet, um zu testen, ob TLS vom Mail-Server unterstützt wird. Um die genaue TLS-Konfiguration festzustellen wird SSLyze verwendet, um die Eigenschaften des Zertifikats auslesen, die Heartbleed-Schwachstelle testen, verwendete Cipher Suites überprüfen und eine DANE-/DNSSEC-Validierung durchführen [2, 3].

Um diese Daten zu analysieren, wurde eine Evaluation im Vergleich zu Empfehlungen des *Bundesamts für Sicherheit in der Informationstechnik* [4] und des *Open Web Application Security Project* [5] durchgeführt.

Literatur

[1] Alban Diquet (Entwickler): *SSLyze - Fast and full-featured SSL scanner*
`https://github.com/nabla-c0d3/sslyze`

[2] Internet Engineering Task Force (IETF): *RFC 6698 - The DNS-Based Authentication of Named Entities (DANE)/Transport Layer Security (TLS) Protocol: TLSA*
`https://tools.ietf.org/html/rfc6698`

[3] Internet Engineering Task Force (IETF): *RFC 4034 - Resource Records for the DNS Security Extensions*
`https://tools.ietf.org/html/rfc4034`

[4] Bundesamt für Sicherheit in der Informationstechnik (BSI): *Technische Richtlinie TR-02102-2 - Kryptographische Verfahren: Empfehlungen und Schlüssellängen - Teil 2 - Verwendung von Transport Layer Security (TLS)*
`https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102-2_pdf.pdf?__blob=publicationFile`

[5] Open Web Application Security Project (OWASP): *Transport Layer Protection Cheat Sheet*
`https://www.owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet`

Real-time DDoS Defense: A collaborative Approach at Internet Scale

Jessica Steinberger*†, Anna Sperotto†, Aiko Pras† and Harald Baier*

* da/sec - Biometrics and Internet Security Research Group
University of Applied Sciences Darmstadt
Darmstadt, Germany
Email:{Jessica.Steinberger, Harald.Baier}@h-da.de

† Design and Analysis of Communication Systems (DACS)
University of Twente
Enschede, The Netherlands
Email:{J.Steinberger, A.Sperotto, A.Pras}@utwente.nl

Over recent years, network-based attacks have become one of the top causes of network infrastructure and service outages. The reason is that attacks are getting larger, more sophisticated (e.g. multi-vector attacks) and frequent. At the same time it has never been easier to execute DDoS attacks, e.g., Booter services offer paying customers without any technical knowledge the possibility to perform DDoS attacks as a service.

Currently, mitigation of DDoS is performed using traditional security solutions such as firewall and Intrusion Prevention System devices [2]. These solutions are often not able to handle the large amount of traffic reaching the target network. One reason is that the monitoring equipment located at the victim side might exhaust its own resources as a side-effect of the attack. Further if a DDoS attack already reached the perimeter of the target network, it is often too late to start mitigation procedures, because the network link is already saturated. Therefore detection and mitigation need to be located closer to the source of these kind of attacks.

In accordance with van Eeten et al.[1], we believe it to be a promising approach to move mitigation from the target network to the network of Internet Service Providers. Additionally, ISPs should collaborate to share and exchange information in context of network security, in order to support proactive detection, real-time and automatic mitigation of current anomalies [3].

In this talk, we present the results of a survey that aims to gain insight into processes, structures and capabilities of ISPs to mitigate and respond to network-based attacks. Further, we proposes a framework for flow-based real-time and automatic mitigation of DDoS attacks in ISP networks. The framework collects and processes network flow-based data e.g. NetFlow/IPFIX from the network edge router of an ISP network. The collected data is used to perform anomaly detection, data fusion and classification. In case of an detected anomaly within the flow-based data a security event is raised. Based on this security event a collaborative process is initiated. The framework collaborates with third parties by gathering and processing security information e.g. from other ISPs, customers or available data e.g. Blacklists, Open DNS resolvers etc.).

References

[1] van Eeten, M., Bauer, J., Asghari, H., Tabatabaie, S.: The Role of Internet Service Providers in Botnet Mitigation: An Empirical Analysis Based on Spam Data. In: OECD Science, Technology and Industry Working Papers. Volume 2010/05. Freeman, Rebecca and Auriol, Laudeline and Misu, Max (2013)

[2] Steinberger, J., Schehlmann, L., Abt, S., Baier, H.: Anomaly Detection and Mitigation at Internet Scale: A Survey In Emerging Management Mechanisms for the Future Internet Springer Berlin Heidelberg, 2013, vol. 7943.

[3] Steinberger, J., Sperotto, A., Golling, M., Baier, H.: How to Exchange Security Events? Overview and Evaluation of Formats and Protocols. In: IFIP/IEEE International Symposium on Integrated Network Management (IM2015), 2015. (May 2015) 261–269

What botnet characteristics can be used for distributed and collaborative network based botnet detectection?

Christian Dietz*+, Anna Sperotto+, Gabi Dreo*, Aiko Pras+

* Universität der Bundeswehr München, Germany
christian.dietz{at}unibw.de; gabi.dreo{at}unibw.de
+ University of Twente, Netherlands
a.sperotto{at}utwente.nl; a.pras{at}utwente.nl

In the recent years, our society has become highly dependant on the Internet. In the future, an increasing number of critical systems and infrastructure, like for example power plants, cars and industrial control systems, will based on ICT. Due to the assets moved through the Internet, it is constantly under attack. Botnets are used for many of such attacks, which reach from banking fraud over distributed denial of service (DDoS) attacks to highly sophisticated multi-vector attacks. However, running a botnet implies to maintain a complex infrastructure, especially as most botnets, today, make use of P2P command and control (C&C) infrastructures. This influences the behaviour of bots and is potentially observable on the network. In the past years, many detection approaches have already been proposed. The botnet phenomenon, however, seems to be more present and challenging than ever before.

We believe that better understanding the characteristics of botnets as well as the limiting influences that are present in real-live environments is essential to improve botnet detection. Botnets are usually spread across the borders of multiple networks and involve large-scale network traffic analysis. Therefore, our approach is to focus on distributed and collaborative detection. The goal of this PhD project is to investigate what botnet characteristics can be observed and how the network specific influences ("noise") can be eliminated to make them exchangeable between different network environments. To achieve this goal, we formulate the following research questions:

- What characteristics of P2P botnets are observable on the network?
- How robust are these characteristics against noise or incomplete data?
- How can such characteristics be made exchangeable between different network environments?

In the first three months of this PhD research project, we performed a study on a sinkholed botnet, to find characteristics that could be used for detection. Early results of our study show that characteristics, like regular temporal behaviour of the bots can be observed. However, our analysis also revealed subtle limitations that are likely to appear in real-live environments. In particular, we observed that these temporal characteristics contained more noise than we expected. Such noise is unlikely to appear in simulated or laboratory environments, but might be even stronger in large-scale operational networks. We believe that multiple infections behind a NAT and/or clock-skews could cause some of that noise. Further research is needed to identify the root cause of this noise. Moreover, we found a hidden relation between the operating system (OS) version and the source port usage of the bots, which can be problematic in some machine learning based detection approaches that make use of this feature.

In a next step, we plan to approach the detection problem in a distributed and collaborative way. This is essential because botnets are a global phenomenon and thus, are not limited to one specific (sub-)network. Our research, on both, the botnet characteristics and the network noise, is enabling the creation and exchange of behaviour signatures for enhanced detection approaches.

Phishingerkennung mittels visuellem Ähnlichkeitsvergleich

Felix Hill
Ruhr-Universität Bochum
D-44801 Bochum
felix.hill{at}rub.de

Ziel dieser Arbeit soll es sein, phishing Webseiten anhand ihres Aussehens mittels Screenshots zu identifizieren und zuzuordnen. Hierbei werden Screenshots von unbekannten Webseiten mit Screenshots von bereits klassifizierten Webseiten verglichen.

Um einen Nutzer und seine persönlichen Daten zu schützen ist die Erkennung von Phishing Webseiten im Internet ein wichtiger Schritt. Diese Arbeit zielt auf die bildbasierte Erkennung von Webseiten ab, so wird direkt auf den Daten gearbeitet, die der Nutzer im Internet sieht, da andere Anhaltspunkte wie die URL oder der Seitenquelltext gezielt manipuliert werden können.

Es soll untersucht werden, welche Erkennungsraten möglich sind, ohne dabei die false-positive Rate von einer Falschklassifikation auf 1000 richtige zu überschreiten. Um dies zu evaluieren werden Erkennungsmuster auf Live-Datenströmen angewendet und evaluiert.

Die Arbeit wird in drei verschiedene Phasen unterteilt:

Die erste Phase befasst sich mit der Evaluation verschiedener Bilddistanzalgorithmen. Unterschiedliche existierende Algorithmen wurden anhand ihres ursprünglichen Anwendungsgebietes bewertet. Lässt sich dieses Gebiet auf den Aufbau von Webseiten übertragen, so wurde dieser Algorithmus näher in Phase zwei betrachtet. 10 der 13 betrachteten Algorithmen sind für diesen Anwendungsfall interessant und wurden weiter untersucht.

Die zweite Phase zielt auf die Implementierung und Tests der vielversprechend erscheinenden Algorithmen ab. Es wurden Phishingscreenshots als Testset und Bilder einiger entsprechend echten Webseiten als Signaturset verwendet. Dabei wurde darauf geachtet, dass die phishing Seiten für das menschliche Auge dem Original ähneln. Anhand dieser Vorgaben wurden Algorithmen ausgewählt und weiter betrachtet.

In der dritten Phase werden die Algorithmen und deren Kombinationen, weiter untersucht. Es wurde ein Schwellwert gesucht bei dem, abhängig vom jeweiligen Algorithmus, sicher gesagt werden kann, dass zwei Bilder die gleiche Webseite zeigen. Der Schwellwert muss jedoch keine statische Distanz sein, es wurden weitere Merkmale der Screenshots gefunden, welche in die Berechnung einflossen.

In dem aktuellen Entwickelungsstadium ist es bereits möglich mit dem bisher erfolgreichsten Algorithmus auf einem Signaturset von 1500 Bildern 83% der zu einem Signaturbild ähnlichen Screenshots in einem Testset zu erkennen und korrekt zuzuordnen, ohne dabei eine falsche Zuordnung vorzunehmen. Auf einem Livestrom verdächtiger Screenshots lassen sich in Echtzeit mit dem trainierten Algorithmus automatisch 6% der Bilder zuordnen, wobei die Anzahl der falschen Klassifikationen unter 0,1% liegt.

Als Ausblick kann das entwickelte Verfahren genutzen werden, um die Phishing Erkennung zu beschleunigen und Trends in einem Strom eintreffender Bilder zu finden. Mit Hilfe der durch den Schwellwert normalisierten Distanzen können die Bilder eines Signatursets gruppiert werden. Gibt es Gruppen von Bildern, die die gleiche Webseite abbilden, so werden Duplikate aus dem Set entfernt. Das Verfahren kann ebenfalls genutzt werden, um unbekannten Trends in einem Eingangsstrom aus Bildern zu erkennen. Taucht in kurzer Zeit eine unbekannte Webseite mehrfach auf, so gibt es eine vielbesuchte Webseite, die noch keine Signatur besitzt.

Literatur

[1] Sadia Afroz and Rachel Greenstadt. Phishzoo: Detecting phishing websites by looking at them. In *Proceedings of the 5th IEEE International Conference on Semantic Computing (ICSC 2011), Palo Alto, CA, USA, September 18-21, 2011*, pages 368–375, 2011.

[2] Eiji Kasutani and Akio Yamada. The MPEG-7 color layout descriptor: a compact image feature description for high-speed image/video segment retrieval. In *ICIP (1)*, pages 674–677, 2001.

Intelligence Driven Intrusion Detection

Matthias Wübbeling*† and Arnold Sykosch*†

* University of Bonn
Working Group IT Security

†Fraunhofer FKIE
Cyber Security Department

Knowing a threat before being confronted with it, is an invaluable advantage for the defender. This provides one with the opportunity to prepare for emerging threats. Therefore, communities are established, to share threat intelligence with each other. Thus, participants learn from each other and may increase their own protection. Threat intelligence is often structured in threat reports. These reports are usually expressed in a format called STIX (*Structured Threat Information eXpression*) [3].

STIX data is meant to describe a threat's meta information like the *threat actor*, the *campaign* he is running and the *tools, tactics and procedures* (TTP) he is employing to achieve his goals. Along with this TTP, a threat report might also describe *indicators*. These indicators "convey specific Observable patterns combined with contextual information intended to represent artifacts and/or behaviors of interest within a cyber security context."[1]

These Observables are defined in a format called *Cyber Observable eXpression* (CybOX). CybOX objects contain a comprehensive description of the object itself by characterizing its attributes. A typical example is the description of a file by its name, location, checksum, type, etc. These objects are the link between a threat description and the infrastructure in which it may be observed. [2]

The description of an object, may be interpreted as its signature. Ideally the object is identified before it reaches a host. Most networks employ some kind of network-based intrusion detection system (NIDS). NIDSs are developed specifically for this purpose, the identification of objects or actions within a network by a provided signature. We therefore created a mapping between CybOX formatted objects and NIDS signatures.

Shortcomings exist, when it comes to network based detection. Not every object is visible to the detection mechanisms. Traffic might be encrypted or the object might be created on the host and never transmitted over the network. There might also be situations where the reconstruction of an object from its representation during transmission is computational to expensive, e.g. when a file has to be identified by its MD5 checksum. Therefore, a host based approach complements the detection. Since the only link between the infrastructures' objects and STIX data is the CybOX object, an alert should reflect the id of a CybOX object.

By the time of writing, a prototype is under development.

References

[1] MITRE Corporation: *Structured Threat Information eXpression Documentation* - `https://stixproject.github.io`, Retrieved: April 1st, 2015.

[2] MITRE Corporation: *Cyber Observable eXpression Documentation* - `https://cyboxproject.github.io`, Retrieved: April 1st, 2015.

[3] Barnum, S.: *Standardizing cyber threat intelligence information with the Structured Threat Information eXpression (STIX)* - MITRE Corporation, 2012.