PAPER

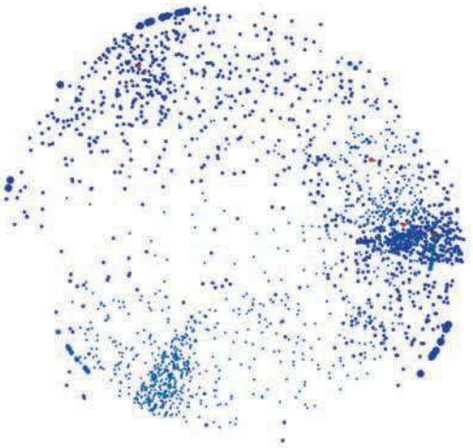
Near-real-time Anomaly Detection in Encrypted Traffic using Machine Learning Techniques

COLLECT

Data fuels the Aramis engine.

Passive network probes are strategically positioned at various nodes within the network, depending on the throughput of the monitored flows and the amount of data transferred. Each sensor collects information from the network segment in which it is installed, analyses it in real time and sends the first results to the local server.

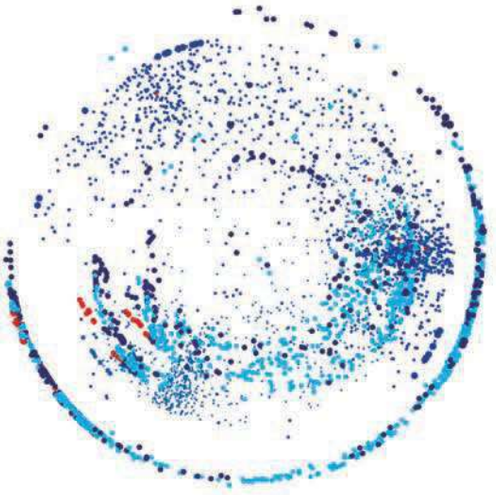
Additional: each sensor can be deployed with a "honey pot" for deception purposes.



ENRICH

Cyber Intelligence is the nitrous.

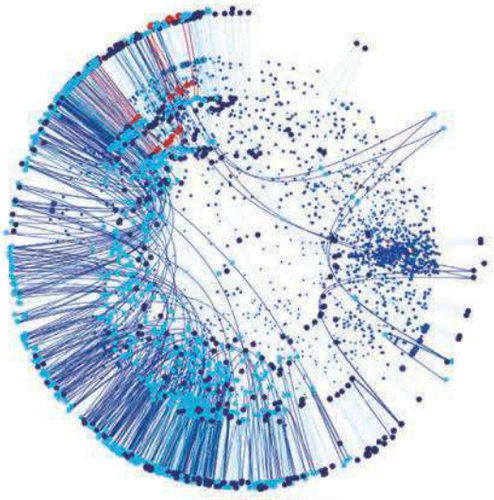
On the local server the data received from the probes are enriched with information from sources such as OSINT and Threat Intelligence as well as using information specific for the customer environment.



CORRELATE

Empower and orchestrate the mixture.

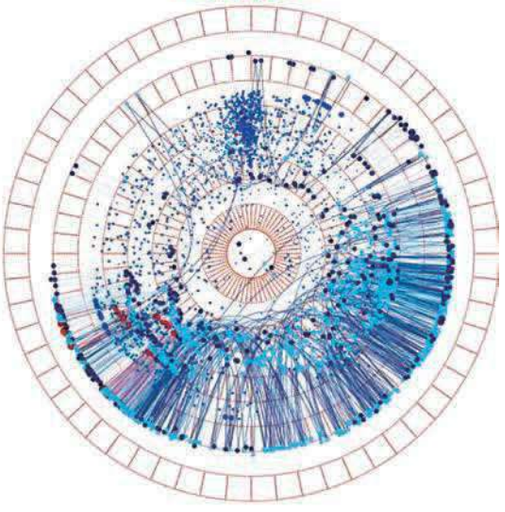
aramis constantly performs two kind of analysis on the collected data:
- Continuous Modulation of its analytics based on the dynamic variation of the measured risks.
- Analysis through the AI Engine of the behavior of each single network
node, in order to detect any possible anomaly.



VISUALIZE

Identify and act — on time.

The information is represented in the dashboards with an effective "cognitive visualization" approach allowing to promptly highlight any minimum deviations from repetitive patterns. These graphics, thanks to their zoom and drill down capabilities, afford analysts with
a powerful tool for the identification and analysis of alarms.



ITASEC

ITALIAN CONFERENCE ON CYBERSECURITY

Roma, Italy
June 20th – 23th, 2022

What is ITASEC?

The **Italian Conference on Cyber Security** is the most relevant conference dedicated to cyber security at the national level and it is organized annually by **CINI**, the **Italian National Cyber Security Lab**. The program is rich with **scientific workshops** and **tutorials**, ad hoc sessions dedicated to **academic paper** presentations, **vendor spaces** and **vision speeches** provided by sponsor companies. The main cyber security related themes include Blockchain, Cryptology, Data Security and Privacy, Security Management and Governance, Operational Incident Handling and Digital Forensics, AI and Security.

Our contribution

aizoOn presents a **scientific paper** titled *"Near-real-time Anomaly Detect-ion in Encrypted Traffic using Machine Learning Techniques"*, a joint work between the Aramis team and the aizoOn SOC.

In the last decade, the adoption of HTTPS for securing Internet communications increased by up to 90%. Threat actors adapted to this transition to HTTPS by writing more sophisticated malware that encrypt their communications with command-and-control centers. On the other hand, network security appliances are limited by the impossibility of inspecting packet payloads for deeper investigations. In this paper, we propose a cybersecurity analytics which monitors encrypted network flows and extracts features to detect possible occurring attacks and anomalies, by combining machine learning with a statistical approach. The analytics is embedded in a network security monitoring platform, named **aramis**, which provides cybersecurity analysts with a comprehensive overview of the monitored network and its traffic to support them in the identification of potentially malicious activities taking place. The detection capabilities of the proposed analytics have been tested both on a benign and a malicious dataset. This latter has been assembled by our security analysts and includes packet captures of samples and tools, respectively, developed and used by worldwide leading threat actors. Results show 96.6% accuracy on the malicious dataset, with a false positive rate approximatively equal to 0.001% when the analytics monitors legitimate encrypted network traffic.

CISDA IEEE

INTERNATIONAL IEEE SYMPOSIUM ON COMPUTATIONAL INTELLIGENCE FOR SECURITY AND DEFENSE APPLICATIONS (IEEE CISDA)

Virtual Event

December 4th-7th, 2021

IEEE SSCI 2021

What is CISDA?

CISDA is a symposium that brings together industry practitioners and researchers, together with academicians, with the aim of **presenting current and ongoing efforts in computational intelligence to detect and adapt to emerging threats**. This year, CISDA presentations will tackle the analysis of the main **challenges to solve problems in the fields of security and defense** with the application of novel techniques, ranging from **neural networks** to **evolutionary computation** and **swarm intelligence**. The symposium is held in conjunction with other symposia, annually organized during the IEEE Symposium Series on Computational Intelligence (IEEE SSCI), in the effort to **promote and stimulate discussion on the latest theory, algorithms, applications and emerging topics on computational intelligence**.

Our contribution

aizoOn presented a **scientific paper** titled *"Near-real-time Anomaly Detect-ion in Encrypted Traffic using Machine Learning Techniques"*, a joint work between the Aramis team and the aizoOn SOC.

In the last decade, the adoption of HTTPS for securing Internet communications increased by up to 90%. Threat actors adapted to this transition to HTTPS by writing more sophisticated malware that encrypt their communications with command-and-control centers. On the other hand, network security appliances are limited by the impossibility of inspecting packet payloads for deeper investigations. In this paper, we propose a cybersecurity analytics which monitors encrypted network flows and extracts features to detect possible occurring attacks and anomalies, by combining machine learning with a statistical approach. The analytics is embedded in a network security monitoring platform, named **aramis**, which provides cybersecurity analysts with a comprehensive overview of the monitored network and its traffic to support them in the identification of potentially malicious activities taking place. The detection capabilities of the proposed analytics have been tested both on a benign and a malicious dataset. This latter has been assembled by our security analysts and includes packet captures of samples and tools, respectively, developed and used by worldwide leading threat actors. Results show 96.6% accuracy on the malicious dataset, with a false positive rate approximatively equal to 0.001% when the analytics monitors legitimate encrypted network traffic.

Near-real-time Anomaly Detection in Encrypted Traffic using Machine Learning Techniques

Daniele Ucci, Filippo Sobrero, Federica Bisio
Data Analytics Team
Cyber Security Division
aizoOn Technology Consulting
Turin, Italy
{name}.{surname}@aizoongroup.com

Matteo Zorzino
Intelligent Security Operation Center
Cyber Security Division
aizoOn Technology Consulting
Turin, Italy
{name}.{surname}@aizoongroup.com

***Abstract*—In the last decade, the adoption of HTTPS for securing Internet communications increased by up to 90%. Threat actors adapted to this transition to HTTPS by writing more sophisticated malware that encrypt their communications with command-and-control centers. On the other hand, network security appliances are limited by the impossibility of inspecting packet payloads for deeper investigations. In this paper, we propose a cybersecurity analytics which monitors encrypted network flows and extracts features to detect possible occurring attacks and anomalies, by combining machine learning with a statistical approach. The analytics is embedded in a network security monitoring platform, named aramis®, which provides cybersecurity analysts with a comprehensive overview of the monitored network and its traffic to support them in the identification of potentially malicious activities taking place. The detection capabilities of the proposed analytics have been tested both on a benign and a malicious dataset. This latter has been assembled by our security analysts and includes packet captures of samples and tools, respectively, developed and used by worldwide leading threat actors. Results show 96.6% accuracy on the malicious dataset, with a false positive rate approximatively equal to 0.001% when the analytics monitors legitimate encrypted network traffic.**

***Index Terms*—encrypted malware communications, passive network analysis, anomaly detection, machine learning, SSL, JA3**

I. Introduction

Nowadays the vast majority of Internet traffic is encrypted thanks to a cross-industry effort involving companies both from private and public sector. This effort started in the '90s but, only in recent years, the percentage of HTTPS encrypted network traffic has experienced a significant increase [1], up to achieving a percentage ranging between 80% and 90% [1]–[5]. Clearly, encrypted communication adoption varies from country to country and may increase quickly in some regions with respect to others [5].

The implications are twofold: on the one hand, threat actors adapted to the transition from HTTP to HTTPS, at a higher economic cost, performing more sophisticated and concealed attacks; on the other hand, network security appliances are limited by the impossibility of inspecting packet payloads for deeper investigations. The combination of these two factors, in 2020, enabled threat actors to perform malware campaigns relying on HTTPS for delivering malware, contacting command-and-control activity, and exfiltrating data [6]. In particular, just in 2020, 67% of malware has been delivered via encrypted HTTPS connections [7]. In addition, data exfiltration and sensitive information stealing have always represented a challenging threat for companies [8]–[10], primarily from a financial point of view [6]. With the mainstream adoption of secure communications (also used by attackers), specific countermeasures need to be taken into account.

Both academia and industry have proposed different solutions to cope with encrypted traffic, as discussed more in detail in Section II of this paper. However, a key point that differentiates the various approaches is their level of intrusiveness: some approaches work directly with encrypted traffic, while others decrypt and re-encrypt data to be inspected. The first ones do not decipher encrypted communications, but consider exchanged data and metadata. For this reason, these approaches are not able to detect compliance and policy violations or possible security breaches by examining traffic payloads. Conversely, there exist approaches implemented in security products, like [11], which decrypt secure communications and allow to analyze payloads. However, decryption and encryption processes insert a significant computational overhead that negatively impacts the performance of these security products [4]. As discussed later in the paper, the protocols on which HTTPS relies on provide many negotiable cipher suites that are not necessarily supported by a specific security product [4]: according to [6], 60% of organizations is not prepared to decrypt HTTPS traffic efficiently.

In this context, we propose an advanced cybersecurity analytics (ACA) which analyzes HTTPS exchanged protocol messages and extract data and metadata to detect possible occurring attacks and anomalies. More in detail, the ACA extracts metadata contained in the fields of X.509 certificates and SSL/TLS metadata and has been designed to detect anomalies taking place during a SSL/TLS handshake between a client and an external server. The analytics combines an unsupervised machine learning technique with a statistical approach: after characterizing the SSL/TLS flow with selected features, a machine learning module isolates anomalous connections and an anomaly score is calculated in order to alert security analysts about potential malicious communications.

The proposed algorithm is embedded in aramis® (Aizoon Research for Advanced Malware Identification System), a commercial network security monitoring platform able to collect, process, and elaborate network flows in near-real time in order to detect and investigate potential malicious or anomalous activities. Network data are processed to detect potentially malicious activities and, in case of successful detection, two different kinds of notifications can be issued to SOC analysts: the first one involves the observation in the network traffic of one, or more, indicators of compromise, while the second type of alerts comes from aramis®' ACAs. Each ACA is a combination of different statistical approaches and unsupervised machine learning algorithms. Starting from these alerts, analysts can rely on the platform's dashboards, that offer drill-down capabilities, to further investigate alert notifications by: correlating alerts produced by other analytics (e.g., detection of malicious payload downloads from compromised sites), or analyzing similar behaviors throughout the monitored network (e.g., machines sharing the same user agent or contacting the same command-and-control center).

The rest of the paper is organized as follows: Section II discusses related work, while Section III introduces basic notions that will be later used to detail the proposed approach (Section IV). The experimental evaluation is reported in Section V and Section VI presents a real-world case study. Finally, Section VII concludes the paper.

II. Related Work

The use of encryption poses significant challenges to network threat detection due to the inapplicability of traditional signature-matching techniques and the increasing number of malware authors taking advantage of it, as outlined in Section I.

The security community has therefore researched in two main directions: decryption of traffic flows [11], [12] and use of network-flow-based metadata [13]. Since decrypting network traffic and applying traditional signature based approaches to detect cyber threats is not always possible, not only due to privacy and legal concerns, but also for the introduced considerable overhead (as discussed in Section I), the combination of passive data extraction from a monitored network and subsequent application of machine learning techniques on SSL/TLS metadata has more and more become an appealing solution [6], [14]–[16]. As an example, [17] performed an analysis over millions of SSL/TLS encrypted flows and a study on 18 malware families by extracting meaningful features from data. With the widespread use of machine learning techniques, research focus has hence moved on the feature engineering tasks [18], [19]. Two different groups of features may be currently found in literature: statistical and sequential features. Statistical features contain but are not limited to flow-level metadata, packet length distributions, time distributions, byte distributions and SSL/TLS header information [17]. An example of deep learning framework combining statistical features can be found in [19]. Sequential features are obtained from the raw flow sequences by learning the generation probabilities of flows. By representing the traffic flow sequence via Markov transformation matrix, [20] clustered certificate lengths and first packet lengths to improve the classification performance under a second-order Markov model.

The approach we present and evaluate in the next sections passively extracts both statistical and sequential features from network flows to detect anomalies in a monitored network. Differently from [20], we leverage machine learning to recognize SSL/TLS handshakes deviating from the ones usually established in the network. Similar to [6] the proposed analytics establishes a baseline of usually secure connections, by using a different set of features.

III. Background

A. SSL and TLS protocols

SSL and TLS protocols allow two machines to authenticate and establish a session key, created to cryptographically protect the remainder of the session [21]. Authentication is performed by means of certificates, which are signed messages reporting the identity of either an individual, a host, or an organization. In the World Wide Web, certificates are typically signed by trusted nodes, called Certification Authorities (CA). The standard used for SSL/TLS protocols to define public key certificates' format is X.509, version 3 [22]. The advanced cybersecurity analytics we propose in this paper analyzes a subset of all the available fields in X.509 certificates, that are briefly described in the following. The signature field contains both the algorithm identifier and hash function used by the CA for signing the certificate (e.g., sha-1WithRSAEncryption). On the other hand, the validity field stores the time interval during which the CA ensures that it will keep information about the certified entity, specified in the subject name field. Each X.509 certificate contains, respectively, information about the subject public key and the issuer: the first specifies the public key itself and the algorithm applied for generating it (e.g., rsaEncryption), while the second reports the name of the CA that issued the public-key certificate.

B. One-class SVM

The original formulation of Support Vector Machines (SVMs) is related to the resolution of supervised tasks, but the one-class SVM has been shown to represent a suitable choice in the context of anomaly detection [23]. It is defined as a boundary-based anomaly detection method, which modifies the original SVM approach by extending it in order to deal with unlabeled data. Like traditional SVMs, one-class SVMs can also benefit of the so called kernel trick when extended to non-linearly transformed spaces, by defining an appropriate scalar product in the feature space.

C. Jenks' natural breaks optimization

This optimization method, applied to power-law distributions, divides input instances in classes by minimizing within-class variance, while maximizing between-class variance [24]. The goodness-of-variance-fit (gvf) value expresses the divergence between predicted classes and observed values. Jenks'

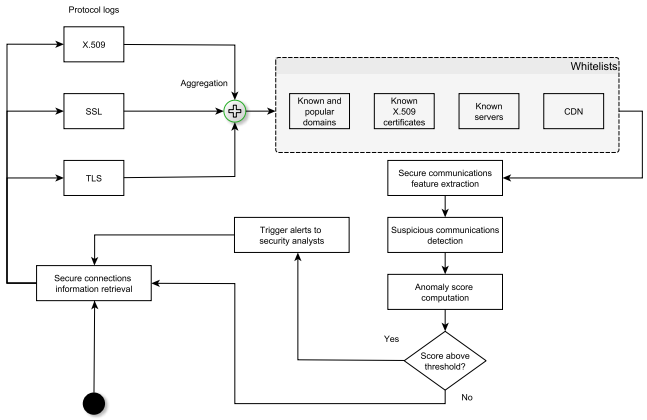


Fig. 1. SSL/TLS analytics overview.

natural breaks optimization consists in iteratively computing the gvf by moving one data value from the class with the largest deviations from the mean to the class with the lowest ones, until the sum of the within-class deviations reaches a minimum [25].

IV. SSL/TLS Analytics

The proposed approach aims at detecting possible anomalies occurring during a SSL/TLS handshake between a client, located inside the network monitored by the software platform outlined in Section I, and an external server. We recall that SSL/TLS protocols enable two machines to securely communicate over an unprotected network (e.g, Internet), as mentioned in Section III-A.

Detection of possible anomalies may be performed by simply analyzing the information exchanged during SSL/TLS handshakes, e.g., by examining the issuer and subject fields of a certificate. Another element that requires particular attention is represented by self-signed X.509 certificates: in this case, the issuer and the subject fields share the same CA value, and the private key employed by the CA to sign the certificate corresponds to the public key certified within the certificate itself [22]. The challenge is here represented by the fact that self-signed certificates can be included in certification paths and can be legitimately used by CAs to advertise information about their operations. However, it is an ever-growing common practice for malware to communicate with their command-and-control servers using a self-signed certificate.

Therefore, the SSL/TLS detection analytics examines information contained in X.509, SSL, and TLS exchanged protocol messages. As mentioned in Section I, aramis® is designed to collect data and metadata related to all the packets transmitted in the monitored network. After data collection, aggregation, and filtering, the SSL/TLS analytics extracts, for each SSL/TLS flow, features able to capture possible anomalies in the communication. Selected features are fed to a machine learning module, which detects suspicious connections, whose anomaly score is eventually computed and possible alerts are signaled to security analysts.


```
{
  "version" : "TLSv12",
  "server_name" : "teams.microsoft.com",
  "curve" : "secp384r1",
  "subject" : "CN=teams.microsoft.com",
  "issuer" : "CN=Microsoft RSA TLS CA 01,
              O=Microsoft Corporation,C=US",
  "server_cert_chain" : [
   {
    "md5" : "28211f1f8a50966b518ec39d3546d57d",
    "sha1" : "4a263f1f39dd526901987ecdb09e2d1297e2bc51",
    "x509" : {
        "version" : 3,
        "key_type" : "rsa",
        "key_alg" : "rsaEncryption",
        "key_length" : 2048,
        "sig_alg" : "sha256WithRSAEncryption",
        "not_valid_before" : 1606847889.0,
        "not_valid_after" : 1638383889.0,
        "subject" : "CN=teams.microsoft.com",
        "issuer" : "CN=Microsoft RSA TLS CA 01,
                    O=Microsoft Corporation,C=US",
      }
    }
  ],
  "ja3" : "7f805430de1e7d98b1de033adb58cf46",
  "ja3s" : "0f14538e1c9070becdad7739c67d6363",
  "cipher" : "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
  "machineDest" : "TEAMS.MICROSOFT.COM"
}
```


Fig. 2. Sample of a communication log including both TLS and X.509 information. For space constraints, in the server certification chain we kept only the end-user certificate.

A. General approach and feature extraction

Figure 1 reports the general structure of the proposed detection method: network traffic involving secure connections is monitored, collected and stored in a database. This knowledge base is periodically accessed in order to retrieve updated information about established encrypted outbound connections, i.e., data and metadata about X.509, SSL, and TLS exchanged protocol messages.

Data and metadata related to the same communication between a client and a server are then aggregated: indeed, a communication is eventually described by complementary information given by both SSL/TLS and X.509. Figure 2 shows an example of an aggregated log comprising both TLS and X.509 information, related to a secure connection established with the business communication platform Microsoft Teams[1]. For privacy reasons, we omitted from the log all the sensitive information as, for example, the IP addresses involved in the communication. It is important to note that each log contains a subset of the information briefly discussed in Section III-A.

The filtering phase allows to remove from input data information about connections to known and popular domains, servers, CDNs, and X.509 certificates trusted by the company where aramis® is deployed. It is worth noting that the platform itself enriches the whitelists database by taking trace of popular secure connections and highly visited servers,

[1]Microsoft Teams:https://teams.microsoft.com/

TABLE I
LIST OF NUMERIC FEATURES EXTRACTED FROM SSL/TLS FLOWS

Feature ID	*Numeric features F_n*
n_0	JA3 popularity (see IV-B for further details)
n_1	Server certificate chain popularity (see IV-B for further details)
n_2	Number of self-signed certificates normalized over a value indicating the maximum length of a certificate chain (e.g., 100)
n_3	Number of expired certificates normalized over a value indicating the maximum length of a certificate chain (e.g., 100)
n_4	Number of certificates reporting an anomalous validity (e.g., a validity less than 3 days) normalized over a value indicating the maximum length of a certificate chain (e.g., 100)
n_5	Number of certificates signed with a weak signing algorithms normalized over a value indicating the maximum length of a certificate chain (e.g., 100)

TABLE II
LIST OF BOOLEAN FEATURES EXTRACTED FROM SSL/TLS FLOWS

Feature ID	*Boolean features F_b*
b_0	The server certificate (or a certificate stored in the server certificate chain) is self-signed
b_1	The certificate signed by the server (or a certificate stored in the server certificate chain) is expired
b_2	The subject contained in the end-user certificate has an invalid top-level domain
b_3	The country listed in the end-user certificate is not valid
b_4	One of the certificates in the server certificate chain has an anomalous validity (e.g., a validity less than 3 days)
b_5	One of the certificates in the server certificate chain relies on a weak signing algorithm
b_6	The server name is not a sub-domain of the end-user subject's certificate
b_7, b_8, b_9	The server name, the subject, and the issuer of the end-user certificate might be randomly generated (see IV-C for further details)

through two different signatures: 'JA3' hashes and server certificate chains. A JA3 hash is defined as a fingerprint of a SSL/TLS flow generated by a client, built from the following handshake information: SSL/TLS protocol version, type of employed cypher, possible extension values [26], enumeration of the supported elliptic curves, and the point formats such curves can parse [27]; hexadecimal values representing this information are concatenated and then hashed trough an MD5 function. On the other hand, server certificate chains allow to identify communications with specific servers, encoded using MD5 hashes: each SHA1 hash identifying a specific certificate in the chain is concatenated with the other chain's SHA1s, which identify the other certificates in the chain. Concatenated SHA1s are then hashed using an MD5 function.

In order to create the feature space to be used by the machine learning algorithm and analytics' modules, the SSL/TLS analytics extracts, for each SSL/TLS flow, both numeric and boolean features which are listed in Table I and II. These chosen features are able to capture signals indicating possible anomalies in the certification chain sent by server to the client.

B. Popularity calculation

Regarding the JA3 and server certificate chain popularity (features n_0 and n_1 in Table I), term frequency metrics calculation is applied [28]. In particular, both the JA3 hashes and the received server certificate chains are generalized as terms $t_0, \ldots, t_i$, respectively generated and exchanged during SSL/TLS handshakes in a predefined time interval Δt. As an example, one can compute the popularity of a specific JA3 hash h, using one of the standard term-frequency tf definitions, commonly applied in information retrieval:

$$tf(h) = \frac{c_h}{\sum_{h' \in H} c_{h'}} \quad (1)$$

where c_h is the number of occurrences of h divided by the total number of hashes $|H|$ observed in Δt. Analogously, it is possible to compute the popularity of a server certification chain scc through $tf(scc)$. Popularity values are normalized and used to filter out either popular secure connections or highly visited servers.

C. Randomness calculation

Randomness of the server name, the subject, and the issuer of the end-user certificate (used to extract features b_7,b_8 and b_9 in Table II) is intended to measure how much their monograms and bigrams characters distributions are "close" to the ones associated to randomly generated strings, according, for example, to the English language characters distribution (note that also other languages can be configured). Such measure is obtained by employing the same approach of an open-source random string detector based on a 2-character Markov chain [29]: this system is trained on pairs of subsequent characters extracted from few megabytes of English text to let the Markov chain learn which is the probability distribution of the appearance of a character following a given one. Thus, the trained Markov chain fed with an input string produces an output probability p indicating how much the string follows the language distribution: the resulting probability grows as the similarity to English words gets higher.

D. Anomaly detection

Numeric features and a subset of boolean features – namely $F_{\text{SVM}} := \{F_n \cup b_7, b_8, b_9\}$ – are given in input to the one-class classifier described in Section III-B. In particular, we rely on a R library implementation of the one-class SVM that uses a radial basis function kernel. SVM hyperparameters ν and γ have been tuned to minimize generalization error, as discussed in [30], and respectively set to 0.5 and 0.1. Moreover, we remove from F_{SVM} all those features whose variance is equal to 0, because they do not add any information to the built model. The model creation phase [31] is performed on the collected historical data, which compose the *training set* used

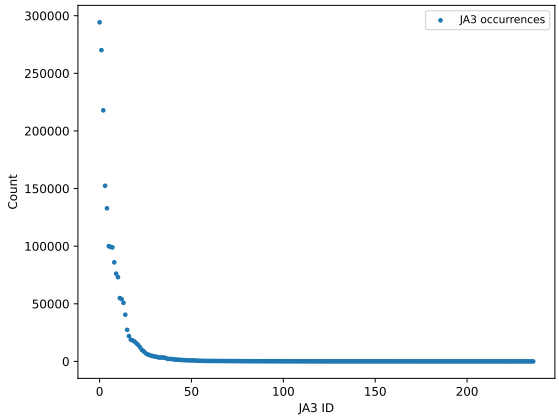


Fig. 3. Distribution of JA3 hash occurrences in the monotored network, observed during a period of 20 days. For privacy reasons, on the x axis, identifiers associated to unique JA3 hashes are reported instead of actual hashes.

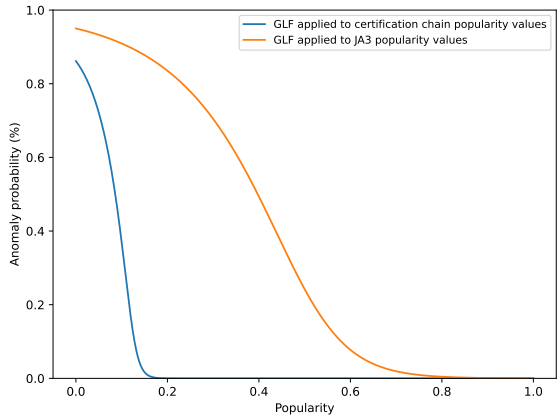


Fig. 4. Example of two generalized logistic functions that convert popularity values into anomaly probabilities. The two functions have been respectively obtained using the following parameters: $K = 0.95$, $S = 0$, $C = Q = 1.0$, $x_0 = 0.125$, $B = 95$, $\nu = 5$ and $K = 1.0$, $S = 0$, $C = Q = 1.0$, $x_0 = 0.5$, $B = 15$, $\nu = 2.5$.

to train the model. Models are periodically updated using 6 hours of secure traffic data. Outputs therefore represent secure connections which deviate – in one or more features – from the ones normally observed in the monitored network. Built machine learning models can be also helpful in detecting zero-day attacks, because malicious traffic samples are not required for the learning phase.

It is worth noting that connections identified as anomalous by the classifier are not necessarily malicious: for example, legit self-signed company certificates are correctly detected by the SVM module as anomalous, but they do not represent a cyber threat. To reduce false positives and be sure of analyzing only suspicious communications that, hence, are not widespread across the monitored network, the analytics filters out popular JA3 hashes. Since JA3 frequencies follow the power law distribution, as shown in Figure 3, we apply the Jenks' natural breaks optimization (described in Section III-C) to extract the least popular JA3s and take into account only rare JA3 hashes. By employing a variant of an R open-source algorithm for computing Jenks' natural breaks [32], the analytics first calculates which is the optimum number of breaks that allows to achieve the maximum goodness-of-variance-fit. Then, by running again the optimization method and providing the optimum number of breaks to divide input hashes in classes, the analytics selects all the JA3 hashes included in the last class (i.e., the least popular). At this point, all the hashes having a term frequency (see Equation 1) higher than the one of the least popular JA3 hashes are filtered out. After filtering, the remaining term frequencies are normalized by dividing them by the maximum observed frequency.

As shown in Figure 1, flows detected as suspicious by the SVM and whose JA3 hashes are not popular are fed to a second analysis module that computes an anomaly probability A_F. The module combines configurable weights and two different parametric generalized logistic functions, which both allow to tune A_F according to the needs of SOC analysts. In other words, a weight w_i is associated to each one of the boolean features F_b listed in Table II and an anomaly probability A_F is computed using a weighted arithmetic mean: $A_F = \sum_{i=1}^{|F_b|} w_i \cdot b'_i$, where b'_i represents the result of the conversion of a boolean value into either a 0 or 1, depending on its truth value (i.e., 0 if false, 1 otherwise). Each weight w_i is adjusted according to the nature of the traffic analyzed by aramis®[2]. As an example, a SOC analyst may set a higher weight to the feature representing whether the end-user certificate is self-signed: if the monitored network does not employ self-signed certificates, then the security analyst may assign a higher weight to this feature in order to spot possible connections relying on a self-signed certificate.

On the other hand, JA3 and server certificate chain popularities are converted to anomaly probabilities using two generalized logistic functions of the form

$$K - \frac{K - S}{(C + Qe^{-B(x-x_0)})^{1/\nu}}$$

This type of functions was initially proposed by Richards in 1959 to model plants growth rate [33], but it turned out to be able to generate more flexible S-shaped curves. In the above equation x are rank values, K is the upper asymptote, S is the lower asymptote value, C, Q, and x_0 are proper constant values, B is the growth rate, and ν affects the direction along which maximum growth occurs. Figure 4 shows an example of two logistic functions, deployed to production, in which the first one has been specifically designed to filter out known and/or highly visited external servers (i.e, whose certification chain popularity is greater or equal to 15%) by assigning them an anomaly probability equal to 0.

Anomaly probabilities resulting from the application of logistic functions are first averaged between them, and then with A_F to obtain the final anomaly probability A. Finally,

[2]Weights w_i reflect the importance given by SOC analysts to the corresponding features b'_i; for the experimental evaluation, assigned weight values are $w_0 = 0.2$, $w_1 = w_4 = w_5 = w_6 = 0.13$, $w_2 = w_3 = 0.005$, $w_7 = w_8 = w_9 = 0.09$.

TABLE III
MALWARE PACKET CAPTURES SUMMARY

Type	*Family*	*Threat actor*	*Domain*	*Year*	*Stage*	A_m
Banking trojan, info stealer	Emotet	Mummy Spider, Mealy Bug	womenempowermentpakistan.com, fynart.com, www.laminatedtube.com, ygpryd.com, thammynhp.com, shop.homenhealthy.com, rocketviral.com, www.campuscamarafp.com, snjwellers.com, pesquisacred.com, theaffiliateincome.com, stars-castle.ir, travianbot.net, lamajesteindustries.com, nanettecook.org	2020-2021	Dropper loading	0.48
Info stealer, banking malware	Ursnif	Golden Cabin	-	2021	Callback	0.63
Info stealer, banking malware	IcedID	Gold Cabin	marslayot.top, garrozalibbo.click	2021	Callback	0.70
Info stealer, banking malware	Bazarloader	UNC1878	-	2021	Callback	0.60
Banking trojan	Trickbot	Graceful Spider, UNC1878, Wizard Spider	api.ip.sb, barionexis.top, ident.me, api.ipify.org, liverpooldabestteamoftheworld.com	2021	Information gathering, dropper loading, and data exfiltration	0.62
Info stealer	Lokibot	Sweed, The Gorgon Group	makiyazhdoma.ru, itsssl.com, hiokurl .com, hyp.ae, pxlme.me bakercost.gq	2021	Dropper Loading	0.49
Info stealer	AgentTesla	Sweed	api.ip.sb, iplogger.[ru|org], ipinfo.io, 2no.co, connectini.net, [alb].xyzgame.cc, fb.xiaomishop.me, spark.lightburst.xyz, ezps.co.uk, shadow-vpn.net, iplis.ru, p6701.softemstore.xyz, 2no.co, www.profitabletrustednet9work.com, bucket.swiftlaunchx.com	2021	Information gathering and dropper loading	0.70
Info stealer	AZORult	The Gorgon Group	tradecontract.es, telete.in, iplogger.org, music-s.xyz	2021	Dropper loading	0.41
Remote Access Trojan	Jssloader	FIN7	injuryless.com	2021	Dropper loading	0.41
Remote Access Trojan	AsyncRAT	-	-	2021	Dropper loading	0.63
Dropper	Chopstick	APT28	cdnverify.net, mvband.net	2020	Dropper loading	0.81
Dropper	Gamaredon Downloader	Gamaredon	hastebin.com religonclothes.com	2020	Dropper loading and callback	0.50
Coin miner	Coinminer	-	iplogger.org	2020	Information gathering	0.51
Ransomware	GandCrab	-	www.billerimpex.com	2018	Callback	0.60

TABLE IV
TOOL PACKET CAPTURES SUMMARY

Tool	*Threat actor*	*License*	*Capabilities*	A_m
Empire	CopyKittens, FIN10, APT19, APT33, Turla, Wirte, Silence, Frankenstein, Wizard Spider, Muddy Water, APT41, Indrik Spider	Open-source	Remote administration and post-exploitation framework	0.69
Cobalt Strike	APT 29, APT32, APT41, Anunak, Cobalt, Codoso, CopyKittens, DarkHydrus, FIN6, Leviathan, Mustang Panda, Shell Crew, Stone Panda, UNC1878, UNC2452, Winnti Umbrella	Commercial	Penetration testing product that allows an attacker to deploy an agent on the victim machine	0.51
Meterpreter[3]	-	Open-source	Attack payload that provides an interactive shell from which an attacker can explore the target machine and execute code	0.60

A is further increased by using a configurable weight w_N ($w_N = 0.25$ in the experimental evaluation reported in Section V) if any of the features observed in the test set has not been seen before: as mentioned in Section IV-D, we filter out features in F_{SVM} whose variance is 0 and, for this reason, it may happen that some features observed in the test do not appear in the trained model. As an example, aramis® may not have seen during the SVM training time an expired certificate; thus, if the analytics observes a secure communication involving an expired certificate, then such anomaly is addressed by increasing the overall anomaly probability A by using w_N. If the anomaly probability A is above a configurable threshold A_{th}, then the security analysts are notified about suspicious secure connections established with external servers located outside the monitored network. A_{th} can be dynamically configured to provide in output only anomaly probabilities approximately greater than 16%, 30%, 44%, and 62%, according to SOC's analysis requirements.

V. EXPERIMENTAL EVALUATION

For the experimental evaluation of the proposed approach, we collected both a malicious and a benign dataset. The first one is constituted by packet captures generated from samples and tools, respectively, developed and used by worldwide leading threat actors [7], [34]–[36], which have been injected in aramis® in order to be processed and analyzed as ordinary traffic. Packet captures of samples and tools have been selected by our security analysts from three online malware analysis services, namely ANY.RUN[4], Hybrid Analysis[5] and

[3]Even though Meterpreter is not a tool used by threat actors, it allows to craft fake self-signed certificates

[4]ANY.RUN: https://app.any.run/

[5]Hybrid Analysis: https://www.hybrid-analysis.com/

TABLE V
LEGITIMATE NETWORK TRAFFIC DATASET SUMMARY

Statistics	*Count*	*One-hour mean*
Total number of SSL/TLS logs	2M	3,478
Total number of machines	578	58
Total number of unique server names	19,421	344
Total number of unique JA3	233	39
Total number of unique JA3s	487	80
Total number of unique cert. chain SHA1s	7,561	257

Malware Traffic Analysis[6]. Chosen samples contain, among legitimate network traffic, SSL/TLS malicious communications. Tables III and IV report a summary of the malicious assembled dataset: for malware, we indicate type and family while, for tools, we report their licenses and corresponding tool capabilities. In both cases, we detail threat actors according to the definitions given by MITRE[7] and Malpedia[8]. Analogously to FireEye [37], we listed also malware stages to better characterize and describe collected samples: dropper loading, callback and data exfiltration. In the majority of analyzed samples, SSL/TLS connections have been established for downloading droppers and contacting command-and-control centers. In addition to the stages described by FireEye, we introduce the information gathering stage, that represents a phase in which malware leverage external services to investigate and reveal external IPs of compromised machines. On the other hand, the benign communication dataset gathers legitimate SSL/TLS communications observed in a real corporate network during a period of 24 days. Table V summarizes general statistics about the network traffic taken into account. Both datasets have been injected in a controlled network environment provided with aramis®, the network security monitoring platform briefly discussed in Section I, to respectively measure the accuracy of the proposed approach and to evaluate the false positive rate on legitimate secure connections. Performed evaluations have been split because the malicious dataset is divided in samples, while the benign one in connections. Hence, depending on the input dataset, quantities of true and false positives and true and false negatives are, respectively, referred to either malware samples or benign connections. As an example, for the malicious dataset, true positives are defined as the number of correctly detected malware/tool samples, while false negatives as the number of malware/tool samples incorrectly labeled as benign. In this case, false positives and true negatives are equal to 0 because no benign sample is included in the malicious dataset. Analogous considerations apply to the benign dataset.

The accuracy measured on the malicious dataset, which includes 59 samples collected by our security analysts, is equal to 96.6%. On the other hand, the false positive rate calculated on the benign network traffic is approximately equal to 0.001%. Tables III and IV report in column A_m the mean of maximum anomaly probabilities computed when analyzing malicious samples. It is worth noting that, even if their values are not high, anomaly probabilities of legitimate communications are usually below these percentages, setting at an average of 40% with a standard deviation of 0.13, which results in $\approx 70\%$ of the false positives having an anomaly probability lower than malicious samples.

Regarding accuracy, among all samples, only two packet captures of IcedID family have not been successfully detected by our analytics. After an investigation, we found out that the JA3 generated by malware samples (i.e., a0e9f5d64349fb13191bc781f81f42e1) collides with several hashes produced by legitimate and common services, like Google APIs and Microsoft Office. This is a known downside of JA3 hashes [38] which may collide with hashes belonging to legitimate applications and prevents the detection of malicious samples. Nevertheless, we are still able to identify malicious encrypted communications performed by malware developed by some of the most famous threat actors. Concerning the false positive rate, it is important to note that the obtained value allows SOC analysts to proceed with deeper investigations avoiding unprocessable amounts of alert notifications.

VI. DISCUSSION

As mentioned in Section I, the proposed analytics is embedded in a network security monitoring platform able to support SOC analysts in investigating cyber threats, as presented further in this section. As an example, we analyze packet captures generated by a malware sample of the family Astaroth, a trojan and an information stealer widely known to attack companies in Europe and Latin America since late 2017 [39]. In July 2021, samples – like the one considered in this section – have been distributed in a Malspam campaign through mails containing an infected link. By accessing this link, the user downloads a compressed Powershell script that, if executed, performs its malicious activity in different infection stages: first, it creates a support file in the public "Videos" folder in which specifies a command-and-control center (C&C) domain; then, the script establishes an HTTP connection with this latter to download an XML file containing Javascript source code, which is later run to download 3 malicious DLLs and a compiler for guaranteeing persistance. Once the malicious DLLs are loaded, they extract sensitive information (e.g., mail, e-commerce and banking accounts) from the victim machine that, afterwards, are exfiltrated to either predetermined or algorithmically generated malicious domains.

aramis® has been able to successfully detect as malicious the downloads of the 3 DLLs and the compiler through an ACA that analyzes HTTP traffic. In parallel, DNS algorithmically generated requests have been detected by two other analytics, responsible for monitoring DNS traffic [40], [41]. Finally, the encrypted communications with a C&C have been correctly identified by the analytics presented in this paper. Interestingly, aramis® did not detect exfiltrations through HTTP with any ACA but, thanks to the platform monitoring capabilies, SOC analysts have been able to identify HTTP requests exfiltrating data, as reported in Figure 5. Our SOC analysts speculate that third level domains, depicted in

[6]Malware Traffic Analysis: https://www.malware-traffic-analysis.net/
[7]MITRE: https://attack.mitre.org/
[8]Malpedia: https://malpedia.caad.fkie.fraunhofer.de/

Fig. 5. HTTP POST requests performed by the Astaroth malware family sample to exfiltrate data from an infected machine.

the figure, are used by the C&C for categorizing exfiltrated information from infected machines.

VII. Conclusion

In this paper, we proposed a detection method for passively discovering anomalies in the encrypted traffic of a monitored network. Relying on a combination of machine learning and statistical methods, the proposed solution identifies anomalous and rare SSL/TLS connections that exchange certificates deviating from the ones usually used in the monitored network. The method has been evaluated both on a malicious and a benign dataset: results show high accuracy in detecting malicious samples developed by worldwide leading threat actors and a very low false positive rate on legitimate encrypted traffic.

References

[1] C. Skipper. (2020, 09) The relevance of network security in an encrypted world. [Online]. Available: https://blogs.vmware.com/networkvirtualization/2020/09/network-security-encrypted.html/

[2] A. P. Felt, R. Barnes, A. King, C. Palmer, C. Bentzel, and P. Tabriz, "Measuring https adoption on the web," in *Proceedings of the 26th USENIX Conference on Security Symposium*, 2017, p. 1323–1338.

[3] Decipher. (2019, 06) Encryption, privacy in the internet trends report. [Online]. Available: https://duo.com/decipher/encryption-privacy-in-the-internet-trends-report

[4] N. Shah. (2020, 08) Keeping up with the performance demands of encrypted web traffic. [Online]. Available: https://www.fortinet.com/blog/industry-trends/keeping-up-with-performance-demands-of-encrypted-web-traffic

[5] Google. (2021, 08) Google transparency report: Https encryption on the web. [Online]. Available: https://transparencyreport.google.com/https/overview?hl=en

[6] Cisco. (2019) Cisco encrypted traffic analytics. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/enterprise-networks/enterprise-network-security/nb-09-encrytd-traf-anlytcs-wp-cte-en.pdf

[7] ENISA. (2020) Enisa threat landscape - malware. [Online]. Available: https://www.enisa.europa.eu/publications/malware/at_download/fullReport

[8] M. Korolov, "Cyber security review," *Treasury & Risk*, 2012.

[9] R. W. Taylor, E. J. Fritsch, and J. Liederbach, *Digital crime and digital terrorism*. Prentice Hall Press, 2014.

[10] T. Yadav and R. A. Mallari, "Technical aspects of cyber kill chain," *arXiv preprint arXiv:1606.03184*, 2016.

[11] VMware. (2021) Vmware nsx network detection and response. [Online]. Available: https://www.vmware.com/products/nsx-network-detection-response.html

[12] F. Callegati, W. Cerroni, and M. Ramilli, "Man-in-the-middle attack to the https protocol," *IEEE Security & Privacy*, vol. 7, pp. 78–81, 2009.

[13] L. Bilge, D. Balzarotti, W. Robertson, E. Kirda, and C. Kruegel, "Disclosure: detecting botnet command and control servers through large-scale netflow analysis," in *Proceedings of the 28th Annual Computer Security Applications Conference*, 2012, pp. 129–138.

[14] B. Anderson and D. McGrew, "Machine learning for encrypted malware traffic classification: accounting for noisy labels and non-stationarity," in *Proceedings of the 23rd ACM SIGKDD International Conference on knowledge discovery and data mining*, 2017, pp. 1723–1732.

[15] ——, "Identifying encrypted malware traffic with contextual flow data," in *Proceedings of the 2016 ACM workshop on artificial intelligence and security*, 2016, pp. 35–46.

[16] C. Liu, L. He, G. Xiong, Z. Cao, and Z. Li, "Fs-net: A flow sequence network for encrypted traffic classification," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019.

[17] B. Anderson, S. Paul, and D. McGrew, "Deciphering malware's use of tls (without decryption)," *Journal of Computer Virology and Hacking Techniques*, vol. 14, no. 3, pp. 195–211, 2018.

[18] A. S. Shekhawat, F. Di Troia, and M. Stamp, "Feature analysis of encrypted malicious traffic," *Expert Systems with Applications*, vol. 125, pp. 130–141, 2019.

[19] H. Shi, H. Li, D. Zhang, C. Cheng, and X. Cao, "An efficient feature generation approach based on deep learning and feature selection techniques for traffic classification," *Computer Networks*, vol. 132, pp. 81–98, 2018.

[20] M. Shen, M. Wei, L. Zhu, and M. Wang, "Classification of encrypted traffic with second-order markov chains and application attribute bi-grams," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 8, pp. 1830–1843, 2017.

[21] C. Kaufman, R. Perlman, and M. Speciner, *Network Security: Private Communication in a Public World, Second Edition*, 2nd ed. USA: Prentice Hall Press, 2002.

[22] International Telecommunication Union. (2021) X.509 : Public-key and attribute certificate frameworks. [Online]. Available: https://www.itu.int/rec/T-REC-X.509-201910-I/en

[23] L. Swersky, H. O. Marques, J. Sander, R. J. Campello, and A. Zimek, "On the evaluation of outlier detection and one-class classification methods," in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2016, pp. 1–10.

[24] G. Jenks, *The Data Model Concept in Statistical Mapping*, 1967.

[25] B. Jiang, "Head/tail breaks: A new classification scheme for data with a heavy-tailed distribution," *The Professional Geographer*, 2012.

[26] Internet Assigned Numbers Authority. (2019, 02) Transport layer security (tls) extensions. [Online]. Available: https://www.iana.org/assignments/tls-extensiontype-values/tls-extensiontype-values.xhtml#tls-extensiontype-values-1

[27] Internet Engineering Task Force. (2018, 08) Elliptic curve cryptography (ecc) cipher suites for transport layer security (tls) versions 1.2 and earlier. [Online]. Available: https://tools.ietf.org/html/rfc8422#page-11

[28] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. USA: Cambridge University Press, 2008.

[29] R. Neuhaus. (2011) Gibberish-detector. [Online]. Available: https://github.com/rrenaud/Gibberish-Detector

[30] S. S. Keerthi and C.-J. Lin, "Asymptotic behaviors of support vector machines with gaussian kernel," *Neural computation*, 2003.

[31] L. Oneto, *Model selection and error estimation in a nutshell*. Springer, 2020.

[32] G. Alberti. (2017, 09) 'plotjenks': R function for plotting univariate classification using jenks' natural break method.

[33] F. J. Richards, "A Flexible Growth Function for Empirical Use," *Journal of Experimental Botany*, vol. 10, no. 2, pp. 290–301, 06 1959. [Online]. Available: https://doi.org/10.1093/jxb/10.2.290

[34] ENISA. (2020) Enisa threat landscape 2020 - botnet. [Online]. Available: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2020-botnet/at_download/fullReport

[35] CrowdStrike. (2021) Crowdstrike global threat report 2021. [Online]. Available: https://go.crowdstrike.com/rs/281-OBQ-266/images/Report2021GTR.pdf

[36] Sophos. (2020, 02) Nearly a quarter of malware now communicates using tls. [Online]. Available: https://news.sophos.com/en-us/2020/02/18/nearly-a-quarter-of-malware-now-communicates-using-tls/

[37] FireEye. (2019, 12) Stages of a malware infection. [Online]. Available: https://community.fireeye.com/s/article/000002205

[38] Abuse.ch. (2021) Ja3 fingerprints. [Online]. Available: https://sslbl.abuse.ch/ja3-fingerprints/

[39] MITRE. (2021) Astaroth. [Online]. Available: https://attack.mitre.org/software/S0373/

[40] S. Saeli, F. Bisio, P. Lombardo, and D. Massa. (2020) Dns covert channel detection via behavioral analysis: a machine learning approach.

[41] F. Bisio, S. Saeli, P. Lombardo, D. Bernardi, A. Perotti, and D. Massa, "Real-time behavioral dga detection through machine learning," in *International Carnahan Conference on Security Technology (ICCST)*. IEEE, 2017, pp. 1–6.

aramis: Aizoon Research for Advanced Malware Identification System

P. Lombardo, F. Bisio, D. Bernardi
aizoOn Technology Consulting

AUSTRALIA EUROPE USA
TECHNOLOGY CONSULTING

MOTIVATIONS

- Cybercrime is one of the most serious threats to the current society
- The knowledge and implementation of cyber security guidelines is crucial
- Malware and attacks rapidly evolve in time and are very heterogeneous (around 80% of malware found in breach investigations is specific to that organization)

COLLECTION OF NETWORK TRAFFIC

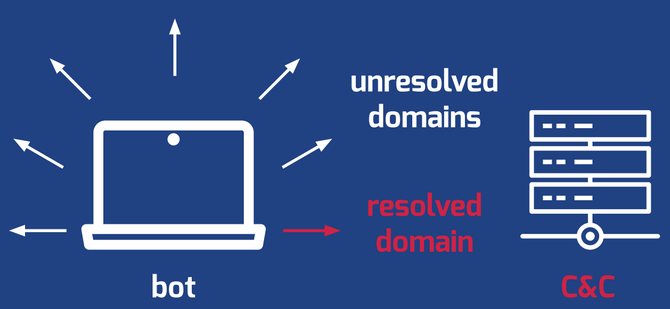
CYBER SECURITY KNOWLEDGE + **DATA MINING TECHNIQUES** → **ADVANCED CYBER ANALYTICS**

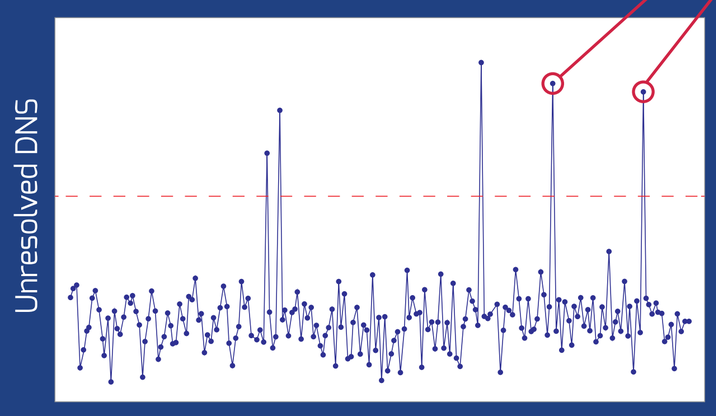
SELF-LEARNED KNOWLEDGE + **MACHINE LEARNING TECHNIQUES** → **MACHINE LEARNING ENGINE**

Each *Advanced Cybersec Analytics* recognizes a specific attack (e.g., **domain generation algorithm[1], drive by download[2], ransomware, IP-flux**) or analyzes a specific aspect of the network traffic (e.g., **network topology, IP geolocation, communication protocol, user agent, scheduled operations, constant data transmission**).

(1) DOMAIN GENERATION ALGORITHM

- Provides a communication channel ***bots*** ⟷ ***command and control (C&C)***
- Each bot generates many pseudo-random domains
- One of them is associated with the C&C and it is resolved

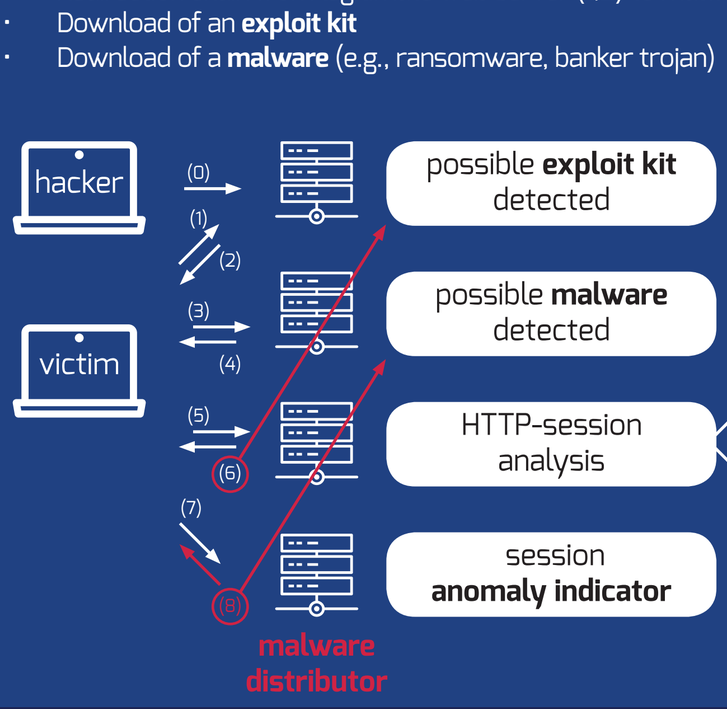




- collection of unresolved domains
- outliers detection
- extraction of resolved domains
- domain features extraction
- clustering (k-means)
- cluster **anomaly indicator**

(2) DRIVE BY DOWNLOAD

- Redirection chain from legitimate web server (1,2) to malware distributor (7,8)
- Download of an **exploit kit**
- Download of a **malware** (e.g., ransomware, banker trojan)

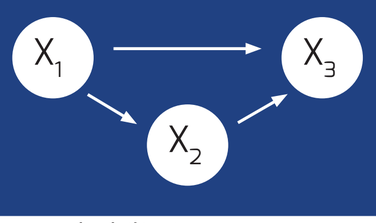


- possible **exploit kit** detected
- possible **malware** detected
- HTTP-session analysis
  - fraction of lecit traffic
  - entropy
- session **anomaly indicator**

The **Machine Learning Engine** analyzes network traffic with two different unsupervised classification algorithms.

BAYESIAN NETWORK

- direct acyclic graph

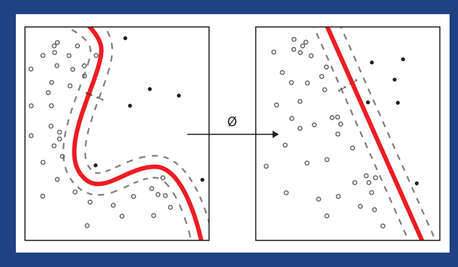


- probabilistic structure $P(X_1, X_2, X_3) = P(X_1)\, P(X_2 | X_1)\, P(X_3 | X_1, X_2)$
- the Bayesian network is built over η variables $X_1, .. X_\eta$

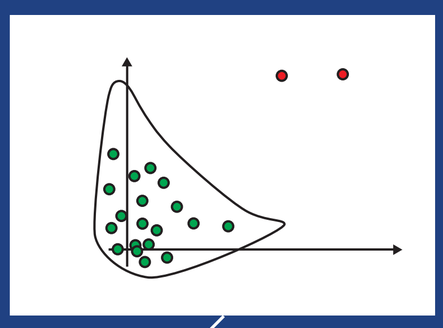
- for each event, for each X_i evaluated
- if $P(X_i | \{X_1, .. \hat{X}_i, .. X_\eta\}) < P_{th}$
- X_i anomalous
- BN **inconsistency indicator** ⟷ # anomalous variables

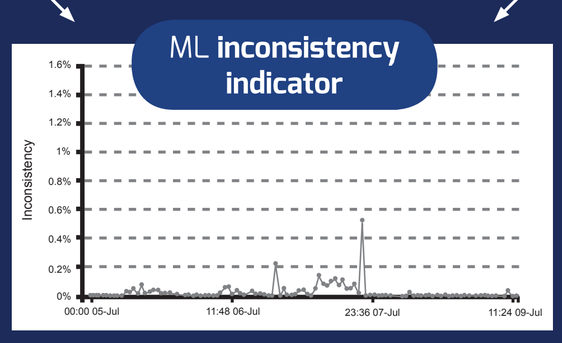
SUPPORT VECTOR MACHINE

- events are points in a multidimensional space
- maximize the *margin*, Gaussian *kernel*



- **one-class SVM**: the training distribution defines the class





Each event is evaluated over:
- a model representing the machine involved
- a model representing a homogeneous class of machines (e.g., clients, servers, etc.)

The variables used for both BN and SVM include communication protocol and service, destination port, duration and volume of the HTTP requests and ws, status code, user agent, etc.

THREAT DETECTION

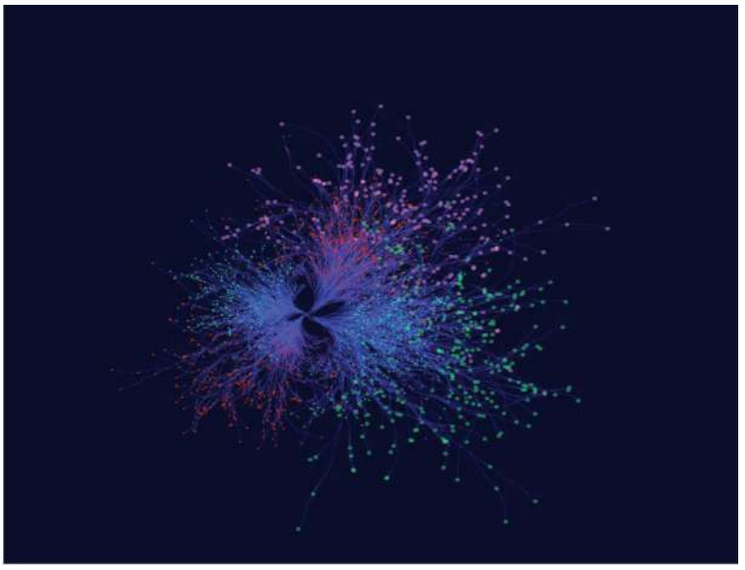
While the *advanced cybersec analytics* automate cybersec experts investigations as much as possible, the *machine learning engine* aims to spot any deviations from the usual behaviors in the network traffic. The combination of these two different approaches allowed for the following **detection results** (found after one month of aramis execution on the network of a medium-size company):

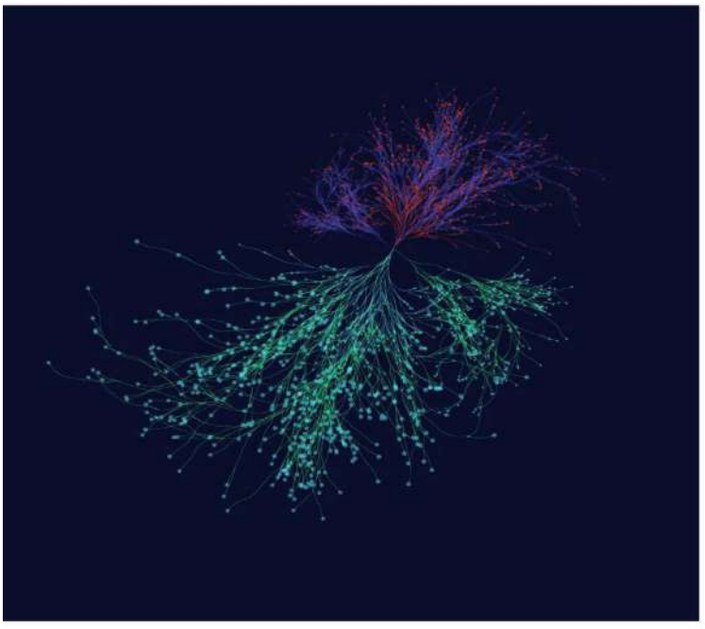
- 1 banking trojan (VawTrak)
- anomalous files exchange ($4 \cdot 10^4$ files/hour) from a client to advertising URLs
- 2 network and resource abuses
- 1 attempt of an Apache PHP remote exploit
- 106 unhautorized TOR connections

Machine learning

Incoherence Index: it represents the **deviation from the «normal» behaviour** of the network.
It is generated by the **Bayesian** algorithms designed by the Data Driven Innovation and Cyber Security teams.

Our Machine Learning engine is based on Bayesian Networks, Support Vector Machine and Bootstrap analysis.





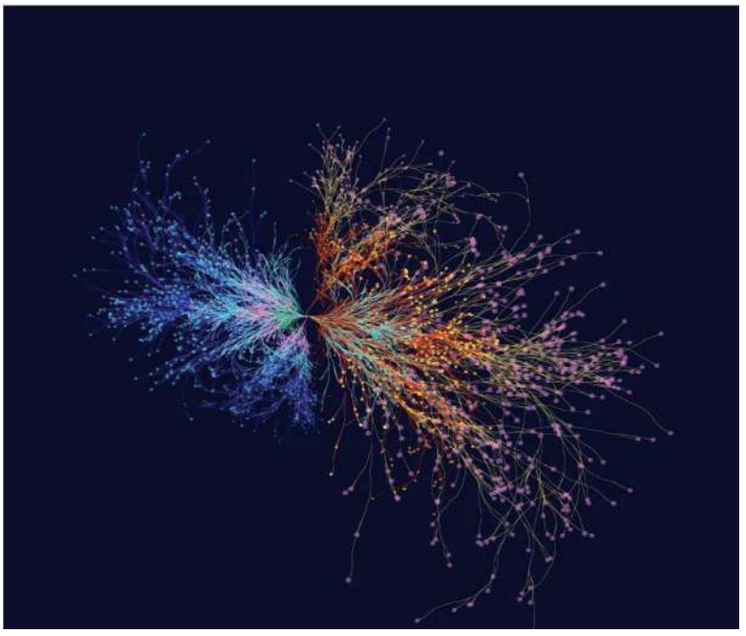
Data mining

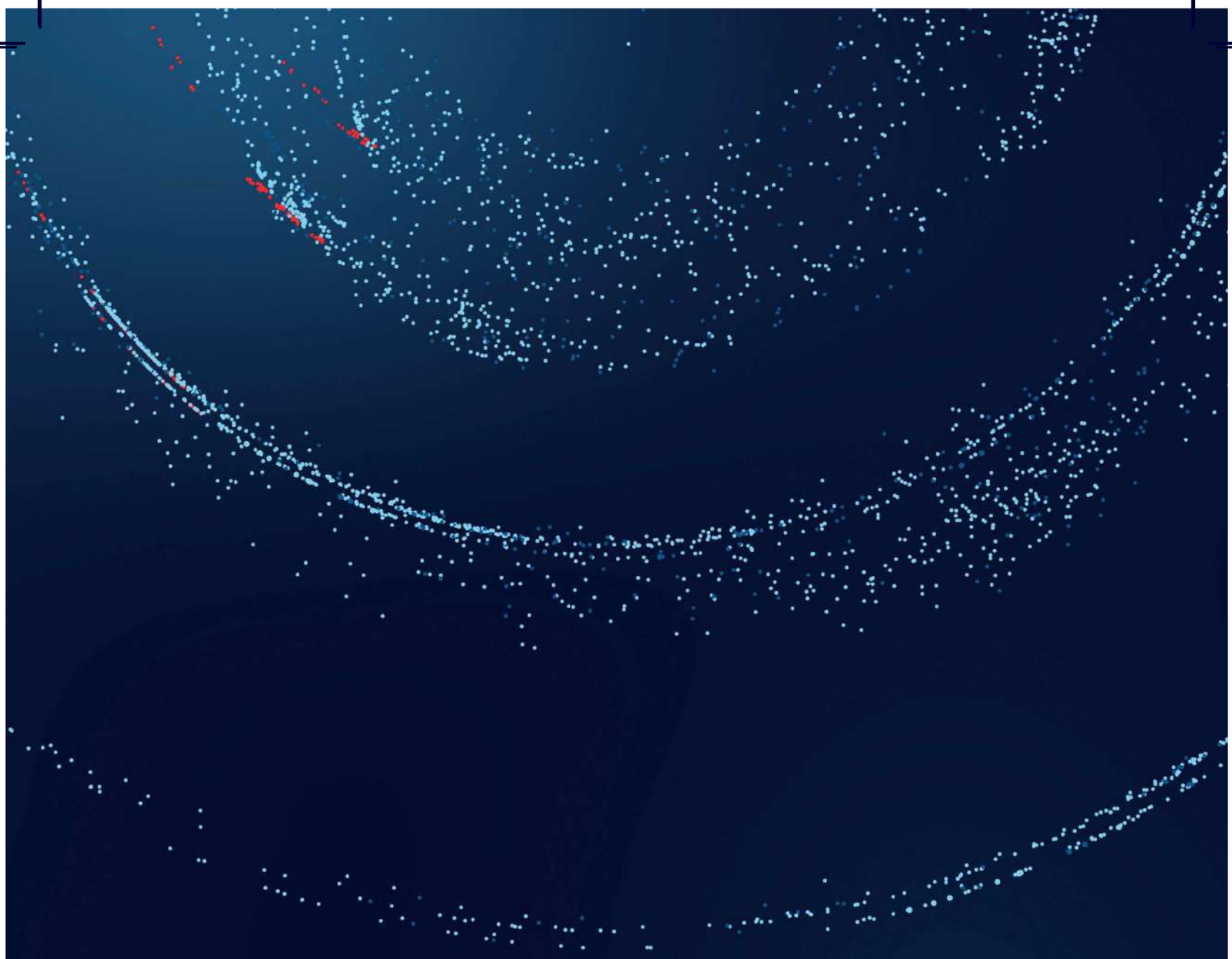
Alarms generated by the **advanced analytics** are developed by the Data Driven Innovation team working closely alongside our security experts.
They are able to do in near real-time more than the analysis that a human network forensic analyst would do. On top of the standard analysis, to check the network traffic connections such as geo-localisation, protocols, quantity of data and so on, our team has developed advanced analytics to spot threats such as Drive by Download, Ransomware, DGA, IP flux and more, with continual study and tuning of the analytics applied to ensure mutations of malicious tool and/or process are incorporated into the engine.

Threat intelligence

Combines the use of publicly available OSINT sources and threat intelligence provided by the **Malware Lab** to identify malicious IPs and DNS, TOR exit nodes and malware inside the traffic analysed by the sensors. We have numerous honeypots around the globe to spot the latest attacking techniques.

The threat intelligence engine gives capabilities such as automatic detection of malware and attacks, scans identification of schemas, correlations and attack patterns.





Get in touch:

APAC

+61 02 8299 7302
aramis@aizoon.com.au

EMEA

+39 06 97605931
aramis@aizoongroup.com

NORTH AMERICA

+1 866 398 6567
aramis@aizoon.us

aramisec.com

aizoon® AUSTRALIA EUROPE USA
TECHNOLOGY CONSULTING