Computer Networks 182 (2020) 107474

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Multi-observable reputation scoring system for flagging suspicious user sessions

Wassila Lalouani, Mohamed Younis *

Department of Computer Science and Elect. Eng., University of Maryland Baltimore County, MD, United States

ARTICLE INFO

Keywords:
Firewalls
Networks security
Intrusion detection
Dynamic multi-observable analysis
Reputation scoring

ABSTRACT

Conventionally, network and cloud infrastructure security is handled by firewalls which monitor traffic and block malicious access by matching certain observables, e.g., IP, and DNS, to blacklisted entries in intelligence databases. Therefore, such an approach fails to deal with emerging threats that utilize unclassified observables, and to report suspicious activities of individual users. In this paper we propose MuSeR, a novel approach to assign reputation scores for observables, even when no prior information is available, and flag suspicious sessions by conducting inter-observable analysis of user requests. In essence, MuSeR opts to assist network and cloud administrators mitigate attacks while avoiding unwarranted blocking of benign access. MuSeR achieves such an objective by associating session reputation scores based on the trustworthiness of the user navigation pattern, and conducting dynamic analysis of individual observables involved within requests. Specifically, MuSeR employs a new machine learning model for classifying observables using features specifically chosen to factor in evidence provided by blacklists, and access patterns of known attacks. To determine a request score, MuSeR maps the classifier probabilities to adaptive subjective logic and then uses multinomial fusion to leverage evidence from the different observables. Given the request scores, MuSeR further promotes a novel session reputation scoring model that uses three-valued subjective logic to handle trust propagation and aggregation over user requests. The effectiveness ofMuSeR is validated using a large dataset obtained from popular databases such as WHOIS, CYMUS, and passive DNS databases.

1. Introduction

Modern societies have become increasingly reliant on information technology, where networked computers constitute a core infrastructure for enterprise operation, trading, banking, retail, etc. Moreover, cloud service providers have become prominent by offering inexpensive and reliable computation and storage resources on-demand. Unfortunately, the criticality of these computation, communication, and storage platforms has also attracted cyber-criminals [1]. Examples of prominent attacks include the malicious Zeus command and control (C&C) server hosted on Amazon EC2 [2], the SpyEye banking Trojan found to be using Amazon S3 storage [3], and the Android malware that exploited the Google Cloud Message service [1] [4]. Thus, guarding network and cloud resources are of utmost importance. The main objectives are to detect malicious user sessions and prevent attackers from accessing the system by blocking connection requests. Decisions for blocking access are usually made based on pre-knowledge of bad IP, DNS and URL that can be observed at the firewall [5]. This is often referred to as static observable analysis and falls short in dealing with unknown observables.

This paper strives to effectively tackle network protection when the categorization of observables as bad or good is not available due to lack of reports, and to detect suspicious activities within a user session. We promote a novel reputation scoring framework for alerting the administrator about suspected observables that are unknown before and providing evidence to assist in taking the right action. Basically the administrator is often challenged with how to appropriately deal with new observables that have some attributes matching bad observables, yet no report is available to support a certain classification. Our approach assesses the user session as well in order to better qualify the evidence about observables so that the administrator gains increased fidelity about malicious access/activities if the session score is low, and does not unnecessarily block certain observables as a precautionary measure if the session is deemed unsuspicious.

Particularly we associate session scores based on the trustworthiness of the user navigation pattern and the analysis of individual observables involved within requests in the session. Again, observables include

* Corresponding author.
E-mail addresses: lwassil1@cs.umbc.edu (W. Lalouani), younis@cs.umbc.edu (M. Younis).

https://doi.org/10.1016/j.comnet.2020.107474
Received 10 December 2019; Received in revised form 3 August 2020; Accepted 4 August 2020
Available online 8 August 2020

parameters tracked by a firewall, e.g., IP, DNS, and URL. The contemporary approach is to actively probe public IP addresses or DNS independent of the context of their usage [6]. Unlike such an approach, we factor in the correlation between observables within the same requests. Indeed, those observables are tightly related to each other, each IP belongs to a domain and each URL involves a domain and a set of IPs. Moreover, we do not just provide nominal categorization of observables as bad or good; instead we track the degree of suspension, in terms of confidence, for each user request according to attribute similarity to known bad/good observables. We further consider correlation among requests and aggregate the trustworthiness of requests within a session. Thus, the focus of our solution is to track and probe individual observables and determine whether to trust a user session, so that a network/cloud administrator becomes able to not only counter suspicious access attempts but also identify user sessions involving malicious activities. Scoring user sessions is also quite useful for cloud systems to point out suspected virtual machines. To our knowledge, such session assessment methodology has not been pursued in the literature.

Motivation: The conventional approach applied by firewalls is to detect malicious individual observables, namely, IP, URLs and domains involved within the session, by matching to blacklists provided by popular intelligence databases [7-9]. Contemporary techniques such as rating-based reputation [10], could be then applied to provide an aggregate measure for the trustworthiness of the session. Unfortunately, the effectiveness of such static blacklisting is limited because there is a large number of new attacks appearing every day and attackers frequently switch their IP, domain and URL to evade detection, thus making it difficult to keep blacklists up-to-date [11]. In other words, static blacklisting does not quickly adapt to new attacks.

Dynamic analysis of observables overcomes the limitations of static blacklists and enables the detection of new malware related to domains or IPs [11]. Most existing dynamic analysis based solutions perform classification based on the known attributes of an observable in order to discriminate between malicious and benign ones. However, these solutions consider observables individually without factoring in the correlation among them. Domain analysis has demonstrated that the relationship between observables provides evidence that boost the effectiveness of malware detection [12]. However, the scope of such domain analysis involves the entire network traffic; thus it is computationally intensive and lacks responsiveness. To overcome the aforementioned shortcomings, we promote MuSeR, a novel multi-observable session reputation scoring mechanism that exploits evidence and suspicious patterns of some observables to score the user requests. To the best of our knowledge, such dynamic scoring has not been pursued before.

Contribution: In MuSeR, a session is viewed as a set of requests, each involving multiple observables. MuSeR, which means “ponder” in old French, explores the interrelationship between the attributes of the involved observables in order to capture similarity with known malicious patterns. The results of such analysis will be captured as a reputation score for the request. A session reputation is then determined by aggregating the scores of the user requests. In other words, the solution consists of three main phases: inter-observables analysis, user request scoring, and session scoring.

During the inter-observables analysisphase, MuSeR employs a new supervised machine learning model using features specifically chosen to factor in the relationship between observables. To illustrate, from the URL, we extract lexical and statistical features of the host, domain, subdomain, path, and any IP contained within it. For a domain, we exploit evidence provided by blacklists and informational databases for its records and the corresponding IPs. Finally, the IP statistical features are based on external reports, historical registry logs along with ASN (Autonomous System Number) information. This mutual correlation between the distinct observables is illustrated in Fig. 1. The selected features of entries in blacklists, suspicious patterns of known malware, botnet, etc. are used as training data. The machine learning classifier

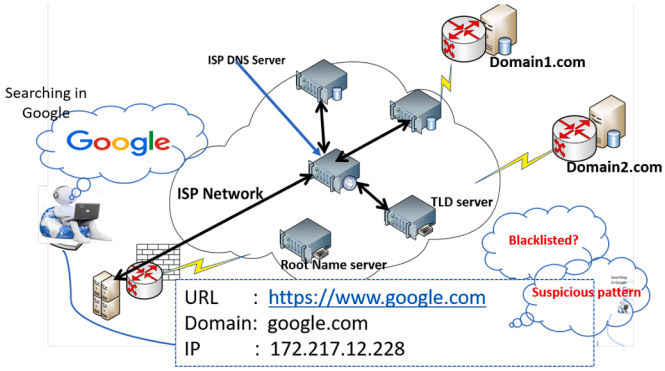


Fig. 1. The multi-observables relationship will check not only the reputation of individual observables, e.g., IP address, but also the correlation between the observables in the request, meaning the URL, domain name, and IP address, within a user request.

provides a probabilistic assessment (score) for whether an observable is malicious or benign. Since the precision of such a score depends on the classifier and training data, during the request analysis the observable scores are modeled as subjective binomial logic [13] in order to assert the degree of trust, distrust and uncertainties provided by each observable. To derive a request score, we use cumulative source fusion [14] to aggregate the scores of the individual observables within the request.

Given the request scores, we define a session score as the probability that no malicious activities are involved. Generally, a suspected session could include a mix of benign and malicious activities. For scoring a session, we consider the dependency between requests to infer the degree of trust for the specific user activities. We formalize the problem as a graph (trust network [15]) where vertices correspond to requests and an edge between two requests implies dependency. The session score is then calculated by applying distortion and aggregation using three-valued subjective logic (3VSL) operators [16] [17]. The effectiveness of MuSeRis validated through extensive experiments using dataset from major intelligence databases. We also analyze the effectiveness of the session scores in comparison with an alternative scheme in the literature. The results have demonstrated that MuSeR achieves high probabilities of detection with low false positive rates. In summary, the paper makes the following contributions:

- Anovel approach for detecting malicious user sessions. Our approach leverages evidence obtained from existing blacklists and features extracted from known attack patterns.
- A rigorous evaluation ofthe attributes of observables in order to identify features that enable detecting malicious observables even when no IP or URL reputation information is available.
- A novel request scoring mechanism that factors in both evidence and uncertainties.
- A novel formulation for the session scoring problem based on the request pattern.

The rest of the paper is organized as follows. The next section sets MuSeR apart from published schemes. Section III highlights the attack model and provides an overview of our system. MuSeR is described in detail in Sections IV and V. Section VI reports the validation results. Finally the paper is concluded in Section VII.

2. Related work

We categorize prior work based on the type of analysis into static and dynamic; and on the scope into single and multi-observables. Overall very little attention has been given to multi-observable analysis and most published techniques consider only a single observable, as

discussed below.

Proactive blacklisting: This category of approaches is based on referencing existing databases of observables that have been associated with malicious behavior, e.g., a host that spreads a malware. As pointed out earlier and noted by prior studies like [18] [19], such analysis is deemed static as it does not adapt well to new attacks and its hit rate decreases significantly over a period of time. Some work strives to mitigate this shortcoming. For example, Sato et al. [20] have proposed a way to extend current blacklists by observing the co-occurrence of domain information. If a domain name *DN* frequently co-occurs with a known blacklisted name, *DN* will be suspected. Meanwhile, Felegyhazi et al. [21] use a blacklist as a seed source to extract all name-servers that have resolved a blacklisted domain within a certain period, and track domains that have switched to the same name-server at the same time. However, the performance of such an approach depends on the availability of historical and registration information. Our work takes advantage of the evidence provided by these intelligence databases and pursues dynamic observables analysis.

Dynamic Observable Analysis: Analyzing DNS data has been a popular means for dynamic detection of attacks against hosts, networks, or the global Domain Name System itself. Some approaches identify infected network nodes by monitoring the DNS traffic and/or the behavior of groups of machines. For example, in [22-25] anomaly-based botnet detection mechanisms are proposed by monitoring group activities in DNS traffic of a specific network. Some work focuses on a specific type of attack. For example, Garera et al. [26] rely on URL properties to detect "phishing" activities while Holz et al. [27] use statistical features to detect fast flux networks. Meanwhile, Anderson et al. [28], Hao et al. [29] and Qian et al. [30] identify and characterize spam tactics to increase the accuracy of spam-oriented blacklisting. The approaches of [6] [31] [32] rely on analyzing the suspicious pattern of blacklisted domains, and domains that are extracted from spam mails in order to predict future attacks. MuSeR is a general framework and can handle various kinds of malicious observables, such as phishing sites, spamming domains, drop zones, and botnet command and control servers, etc., by factoring in their suspicious pattern over a set of features meticulously chosen for such purpose.

Dynamic observable analysisapproaches can also be categorized based on how DNS data is collected into: active DNS probing, passive DNS, and WHOIS requests. The data collected affects the feature extraction and thus the results of the analysis. The methodology pursued by [33-35] is torepeatedly issue queries with the objective to detect the abnormal patterns. Although such methodology performs well in detecting new botnets, it imposes excessive overhead that degrades the network performance. Furthermore, it does not preserve the privacy of the IP addresses of the clients that issued the DNS queries. MuSeR reduces such overload by exploiting aggregated domains over the passive DNS with the focus on the activities of clients. On the other hand, Zdrnja et al. [36] have studied passive DNS, i.e., how the domain data can be aggregated and used for detecting spams. Perdisci et al. [37] perform passive DNS analysis on recursive DNS traffic collected to detect malicious Fast-Flux services. Passive DNS monitoring has also been pursued for the identification of malicious domains in [11] [12] [27] [37-40]. The main drawback of passive DNS analysis is ignoring the relationship between observables, which increases vulnerability once the features extracted are known by attackers. We argue that a combination of WHOIS information and passive DNS provides useful insight about the observables overloading the network with excessive traffic.

Multi-Observable Analysis: A category of existing work factors in the relationship between the IP and domain. For example, Khalil et al. [41] argue that with single observable analysis, attackers may know what features are being employed in detecting malicious domains and use such knowledge to evade detection. To address this issue, they have developed graphs reflecting the global associations among domains and IPs, and proposed a path-based mechanism to derive a malicious score for each domain based on their topological connection to known malicious domains. However, the complexity of such a domain-domain similarity graph and path-based interference scoring is high and does not suit dynamic analysis. The same research group [42] has adopted belief propagation on the graph to identify domains controlled by the same entity. To detect malicious domain/IP, Najafi et al. [43] have pursued belief propagation on a graph using the domain to IP resolution, domain to domain referral and sub-domain relationship.

Graph-based classification has also been used to infer Homophily relationships between devices and their installed apps to detect unknown compromised [44]. Generally, most graph inference based approaches use belief propagation to assess maliciousness based on association. On the other hand, MalRank [45] uses a knowledge graph to model the association among entries in DNS logs. The authors argue that belief propagation enables labeling a node based on prior knowledge of its neighbors in the graph; yet a balance among labels is assumed, which would bias toward the assessment towards benign classifications. The result of MalRank outperforms existing belief propagation approaches in detecting malicious behavior. MuSeR further considers the uncertainty on observable scores and captures the inter-play between observables within a request. In Section VI, the performance of MuSeR will be compared to MalRank.

Peng et al. [46] also detect malicious domains by focusing on the domains that are not resolved to IP addresses directly, and only appear in DNS CNAME records. Their rationale is that domains connected by CNAME resource records share intrinsic relations and are likely to be similar to one another. Watkins et al. [47]have proposed a semi-supervised machine learning approach to filter out non-malicious domains using passive DNS data and the association between domain names and IP addresses, i.e., based on their interaction. Meanwhile, in [48] the dependency between domain and IPs is also factored in in order to characterize time-series patterns of DNS queries and extract temporal behaviors. Ma et al. [49] also consider IP dependency for large domain names and present a graph-based method to mitigate advanced persistent threats. MuSeR stands out in terms of factoring in the uncertainty within the observables relationship and at the same time focusing on the reputation of the user session.

3. Attack model and solution strategy

This section highlights the specific challenges and states the design objective of MuSeR. In addition, we present the architecture of MuSeR and provide an overview of its capabilities and operation. We further discuss how MuSeR can be integrated with existing network security infrastructure.

Adversary Model: This work considers contemporary threats to networked computer infrastructure where an attacker tries to intrude into the system by gaining remote access. If successful a broad range of malicious activities could be launched such as implanting malware, spreading a worm, deleting files, generating spam emails, launching denial of service attacks, etc. To guard the system, a firewall is often employed to monitor packet traffic. The firewall is provided with a set of rules, defined by the system administrator, to determine how to filter packets. For example, the rule could be to discard packets originated from IP addresses, URL, and DNS that belong to a provided list of known malicious entries. Such a list is often compiled from intelligence databases that are made available and maintained by multiple public and private sources. The databases are updated based on reported attacks after being subjected to an extensive analysis. Although firewalls could also apply dynamic analysis of observables, as pointed out earlier, such analysis is computationally heavy and often cannot be conducted on the firewall itself. Thus, an adversary could exploit unknown observables to launch attacks relying on the tardiness in updating the firewall rules and the dissemination of the new intelligence. Moreover the accuracy of dynamic analysis is not high with high false-positives and false-negatives.

MuSeR opts to tackle the aforementioned vulnerability by

proactively classifying new observables by matching the attributes to known bad/good ones and providing a measure for the level of suspicion as a guide to the administrator. Nonetheless, such classification does not constitute evidence to block an IP, DNS or URL. This is especially true when the level of suspicion is not very high. Therefore, additional analysis would be required to avoid penalizing benign observables. MuSeR fills such an important technical gap by analyzing user requests and aggregating the assessment across sessions. Such session-based analysis will not only provide more evidence to the administrator but also predict malicious users before an attack takes place. The latter is particularly invaluable for cloud systems where virtual machines can be exploited; this capability is not possible through contemporary cloud security techniques.

Approach Overview:MuSeR provides the following three key capabilities that: (i) employing a multi-observable analysis to enable detecting malicious observables even when no IP or URL reputation information, (ii) quantifying the level of suspicion in a user request while factoring in both evidence and uncertainties, and (iii) aggregating the request scores across sessions to alert administrators misbehaving users. These capabilities are invaluable for increasing resilience to cyber-attacks, and are not supported by prior work, to the best of our knowledge. MuSeR operates in a training mode and a regular mode. For the training mode, we collect a knowledge basethat includes whitelists and blacklists of individual observables, namely, IP, DNS, and URL. For each observable, we extract a set of features and assign a label that reflects whether the observable is legitimate or malicious. In particular, we consider three sets: blacklist based features, suspicious pattern features and inter-attributes based features.

MuSeR then leverages machine learning techniques to analyze the feature vectors to define rules that can effectively distinguish between malicious and legitimate observables. In our experiments, as explained in Section VI, we have compared eight classifiers in order to select the most appropriate for each observable. Such supervised learning enables MuSeR to determine the statistical characteristics of malicious observables and access patterns in order to monitor new requests. Given an observable, the statistical classifier S assigns a label and a reputation score, which expresses whether the query/use patterns of such an observable resembles either known legitimate or malware behavior, and with what probability. In order to account for possible classification errors, MuSeR considers the precision of the classifier as well as the vector of probabilities to assert evidence (trust and distrust) and uncertainties. Thus, the evidence indicates whether the observable could be trusted and associates a confidence level (belief and disbelief). Finally, MuSeR aggregates the reputation scores of the different observables within a request using cumulative multi-source fusion [14]. Given the aggregated scores of requests, MuSeR models the requests as a trust network in order to infer the session reputation score. Fig. 2 summarizes the main steps of our solution.

System Deployment: From a system architecture point of view, MuSeR could be deployed at the firewall itself relying on agents like Snort [50] to collect the data. Fig. 3 shows an articulation for such a scenario where the system administrator would load the intelligence databases to the firewall and define the action rules. MuSeR is to report the session scores to the administrator who in turn determines what action to be made, or even craft rules for the actions to be taken by the firewall autonomously. Alternatively, MuSeR could be assigned to a designated server or on the cloud; in such a case the administrator will enable the dissemination of the data collected by the firewall to where MuSeR is to execute. Such a deployment option could be necessary if the firewall is constrained in terms of computational resources.

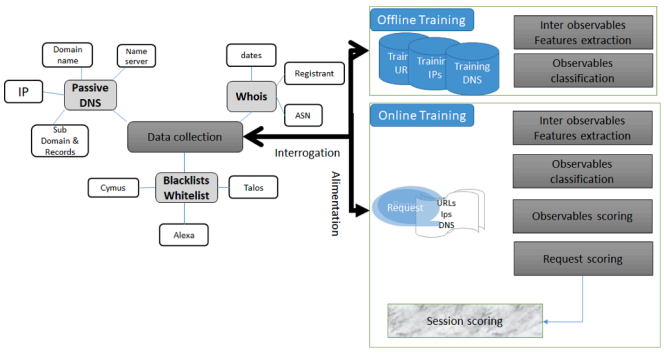


Fig. 2. An overview of MuSeR's operation; the left side highlights the relevant data that MuSeR considers to perform the steps on the right side.

4. Multi-observable analysis

As pointed out in the previous section, MuSeR provides three novel capabilities. This section focuses on the first, namely, the multi-observable analysis that, unlike prior work, conducts inter-observable correlation by considering the attributes of the observables within a user request. The scoring of requests and sessions are covered in the next section. The multi-observable analysis involves three steps, specifically, data collection, features selection and observable score.

4.1. Data collection and features selection

As explained earlier, MuSeR employs a supervised learning model to distinguish between malicious and benign observables through the classification of their respective features. In order to determine relevant features, MuSeR exploits the relationship among observables within a request. Basically, the URL includes three parts: the path, host and domain. The domain includes a set of records [51], most of which have association with distinct IPs. An IP, in its turn, involves an ASN number and BGP ranking. In the balance of this section, we describe the selected features for IP, URL and DNS and explain why we believe that they may be indicative of malicious behavior. We distinguish three types of statistical features:

1) *Blacklist-based*: Ma et al. [33]have shown that the capabilities of a classification system can be increased by combininginformation-frommultiple intelligence databases. Therefore, to define blacklist-based features we factor in the evidence of malicious behavior based on which an observable is blacklisted.
2) *Suspicious pattern*: Previous work on data analysis provides valuable insight about patterns of known malicious behavior like spam, phishing, fast flux, etc. MuSeR exploitsattack databases to define suspicious pattern based features.

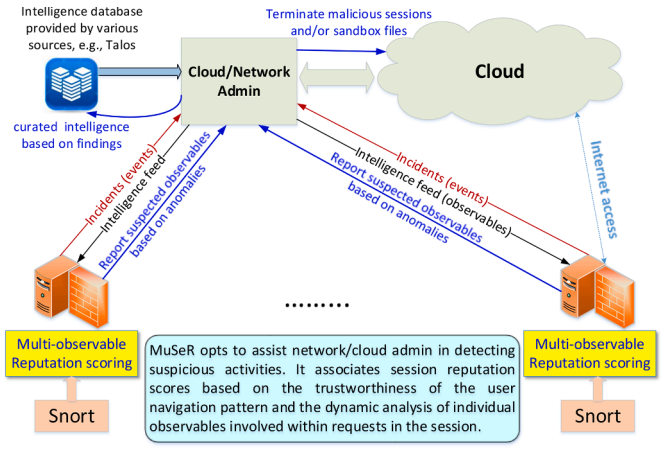


Fig. 3. An articulation of the system architecture showing where MuSeR could be integrated. We note that MuSeR could also be executed on the cloud based on the data provided from the firewall to cope with resource limitations.

3) *Inter-attributes based*: The identity of the malicious domain and IP, the registration update log along with the correlation with particular registrar contact information for an observablecould provide valuable insight about unknown observables. Thus, MuSeR defines inter-attributes correlation features.

URL Features: Considering the different parts of a URL, we exploit a combination of lexical features such as: the URL length, the number of dots in the URL and a binary bag-of-words representation of the hostname and the path [52]. As most benign URLs consist of meaningful words, we count the number of meaningful words within each URL part. Similar to [52], we check whether the URL contains some unsafe keywords and whether the IP address is associated with the hostname. Using the domain name included in the URL, we consider also its ranking that is provided by Alexa [53]as well as safe browsing lookup tool [54]. To train the MuSeR's supervised learning model, we use the whitelisted popular Global Sites from Alexa since well-known URLs are very unlikely to be malicious. We also use malicious URL extracted from the following intelligence sources [55], and [56].

DNS Data and Features: To identify relevant DNS features for distinguishing between benign and malicious domains, we first have recorded the passive DNS information from [57], the registration data from WHOIS [58] and the domain records of DIG [59]. The passive DNS data includes the name of the queried domain, the first and last time queries were issued along with the type of their respective domain record. We have also exploited data provided by the WHOIS database, particularly, the registrar information, dates for recent updates, and ASN information. Using WHOIS, we can determine when a domain was first registered, when it was inserted in and removed from the zone file or transferred between registrars. In order to factor in the DNS records, we collect DIG data [59], namely, A, AAA, SOA, TXT and NS. Then, we extract the list of IP addresses that are associated with each queried domain. It is important to mention that those records complement each other and provide useful insight about the domain. To illustrate, the A, AAA and NS records are used for conversion of domain names to/from corresponding IP addresses. Thus, they are indicative of the frequent usage of the domain. In addition, CNAME records are used for creating aliases of domain names and can hold arbitrary non-formatted text strings. Those records may hint malicious domain if their usage is more frequent then the domain conversion records. Moreover, the SOA record specifies core information about a DNS zone like the primary name server, the email of the domain administrator and more importantly the domain serial number, and refresher timers that may be used to infer the request patterns and their irregularities.

Based on the collected data, we classify the domain, by inspecting the diversity of IP addresses associated with the DNS records. Basically, we determine the number of IP addresses, analyze the location of each IP address, to which ASN it belongs, whether it has been blacklisted, and factor in whether the IP addresses of the associated A, MX and NS records are located within the same AS. Although the approach of [60]also checks the diversity of the IPs, the analysis is limited to domains that are included in spam emails. By considering suspicious patterns, it has been shown by Zhou et al. [61]that domain generation algorithms (DGAs) are often used for a short period of time (active time) and have similar life and query style. We argue that WHOIS databases can also provide valuable information about domain lifetime duration, the difference between creation and expiration date, and thus possible presence of DGA-created domains.

It has been shown in [62] that DNS query response patterns of known malicious domains are irregular and observed that many freshly registered domains are used for malicious activities. Since we are unable to track the DNS activities due to the overhead of communication as explained in the previous section, we thus factor in the freshness of the DNS records and the length of the period of activities. Freshness is expressed as the duration once the last record has been seen, while the activity duration reflects the time between last seen and first seen. The authors of [62]also demonstrate that zone based features carry discriminative power and show empirically that a legitimate domain will not change its hosting name-server very often, while malicious domains tend to do so. The most notable features here are the maximum number of days for a name-server to host a domain (TTL) and ratio of active versus non-active name-servers for a domain. Not only do we use the TTL value of the start of authority (SOA) records but we extend their list of features to include not only the frequent change in the SOA but also A, AAA and NS records.

CNAME records have been used in the literature to identify inactive domains in the absence of the other records. We compare the relative number of CNAME and TXT records to the overall records in order to detect fictitious domains. Liu et al. [63] have shown a relationship between blacklisted spam and the management activity on these domains as recorded in WHOIS databases and DNS zone files. Therefore, we track the registration information like contact names and date of registration in order to factor in the domain with the same contact information of malicious domain. A list of the considered features can be found in Appendix A.

IP Data and Features: To identify the appropriate IP features, we check IP statistics that include the total number of packets blocked from an IP and the number of unique destination IP addresses for these packets [64]. In addition to the aforementioned IP features we note the period until the last reported attack, the last date that the IP has been seen, the size of the AS, the number of reported time, and the number of observed attacks. Furthermore, MuSeR uses the BGP ranking of the ASN, obtained using [65], to determine the rank of the host as well as the country for the IP. Similar to domains, MuSeR uses IPWHOIS to find out the owner type of an IP address, e.g., enterprise; MuSeR also factors in the IP address lifetime using IPWHOIS registration and expiration dates. Moreover, MuSeR promotes inter-attribute correlation features in order to check for suspicious registrar and the consistency of the registration. The whitelisted IPs in MuSeR training set are the IP corresponding to Alexa popular domain [53]. The considered blacklists for malicious IPs are obtained from Virus Total [9], Talos [66]and DNSBL [67]. In the next subsection, we will highlight how MuSeR exploits the aforementioned features to detect suspicious patterns in requests and sessions.

4.2. Observable scoring

Our objective is to score each observable given the probabilities vector $P(P_q^t, P_q^{\bar{t}})$ provided by the supervised learning classifier, where $P_q^t, P_q^{\bar{t}}$ indicate, respectively, the probability that an observable q is benign or malicious. It is important to note that in some cases, the classifier may indicate that a certain observable takes one of several possible states, but it is not clear which one in particular. On the other hand, these probabilities depend on the classifier precision and training data. Moreover, the precision of a classifier depends on the type of observables (IP, URL and domain) that should be later aggregated within the same request score. Given the aforementioned three notes, it is often practical to consider not only the possible binomial value for an observable (benign and malicious) but also their composites values, which constitute the uncertainty of the classification along with the degree of belief for each possible value. By doing so, we can lower the reputation score as we have more evidence of "bad associations" with malicious observables. Therefore, MuSeR factors in the precision of the classifier in order to assert the degree of trust, distrust and uncertainties for each observable. Then, MuSeR employs an adaptive subjective logic approach to infer the observable score.

A subjective opinion over a variable q is represented in subjective logic by a quadruple of real numbers $\omega_q = (b_q, d_q, u_q, a_q)$ where b_q, d_q, u_q, a_q are the belief, disbelief, uncertainty and relative atomicity of q, respectively. Meanwhile, a_q is the base rate probability distribution expressing prior knowledge about the specific class of random variables, so that in case of significant uncertainty about a specific variable, the

base rate indicates the default likelihood. In the case of unknown observable, a_q would equal 0.5 assuming equal probability for an observable being good and bad. The sum of the belief masses is less than or equal to 1, and is complemented with an uncertainty mass which reflects the opinion's confidence level. Therefore, $b_q + d_q + u_q = 1$. In particular, when $u_q = 0$, i.e., we have a dogmatic opinion because there no uncertainty while $b_q = 1$ is known as an absolute opinion. In contrast, an opinion with complete uncertainty, is called a vacuous opinion.

The following equation indicates how MuSeR converts the classifier probabilities to subjective opinion:

$$\left.\begin{aligned} b_q &= Pr_C \times P_q^t \\ d_q &= Pr_C \times P_q^{\bar{t}} \\ u_q &= 1 - b_q - d_q \\ E_q &= b_q + a_q * u_q \end{aligned}\right\} \quad (1)$$

where: P_q^t, and $P_q^{\bar{t}}$ are the probabilities assessed by the classifier that an observable q is malicious and benign, respectively (i.e., $P_q^t + P_q^{\bar{t}} = 1$); Pr_C represents the precision of classifier C; b_q, d_q, and u_q are the belief, disbelief, uncertainty about the trustworthiness of q (i.e., q is benign); E_q is the expected probability of q being benign (while factoring the uncertainty in the classifier assessment).

It is also important to note that the subjective logic operator overcomes the binomial classification of observables. In fact, simple visualizations for binomial and trinomial opinions are based on a barycentric coordinate system. As illustrated in Fig. 4, the Barycentric Coordinates are simply an equilateral triangle with vertices belief, disbelief and uncertainty [13]. The opinion is represented as a center of gravity (barycenter or geometric centroid) of locating three masses M_A, M_B, and M_C at the triangle vertices. These masses are represented by b_q, d_q, and u_q, respectively, and located over three axes perpendicular over the opposite triangle side of each vertex. The base rate a_x is represented by a point on the side of belief and disbelief. The line connecting the uncertainty vertex to the point represented by a_q is called the director. The projected probability E_q of an opinion ω_q (reputation score of observable q) can be determined by drawing a line from the opinion point ω_q to the base and parallel to the director line [13]. In the next section, we will aggregate the IP, URL and DNS scores to provide the score for the request.

5. Score aggregation mechanisms

The multi-observable analysis yields reputation scores for the unknown observables. MuSeR analyzes the individual scores within a user request and aggregates them across the various requests within a session. The resultant session score reflects the evidence and uncertainty that malicious activities are taking place within the session. To the best of our knowledge MuSeR is the first to provide such quantitative assessment.

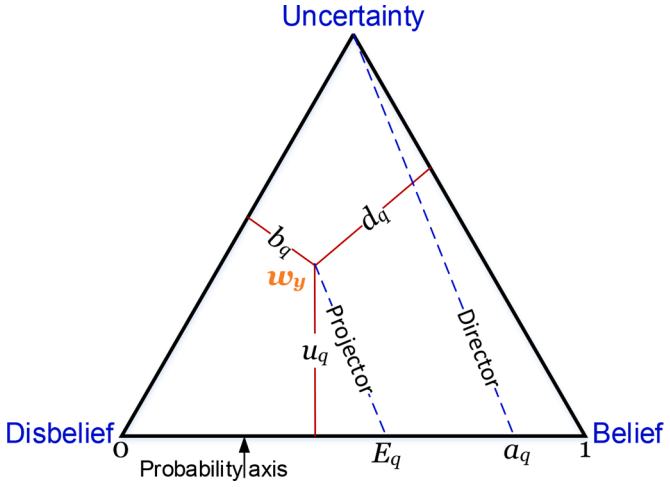


Fig. 4. Illustrating the application of subjective logic using Barycentric Coordinates.

5.1. Request scoring

The objective of the request scoring is to aggregate the knowledge about the distinct observables in the same request. Some observables within a request may have malicious characteristics, while the other do not. Such conflicting classification of observables may yield wrong assessment if the degree of certainty is not considered. Therefore, MuSeR factors in the classification fidelity in order provide fine-tuned tracking of any malicious behavior of the user. Basically, the observables represented by their subjective logic vector constitute multiple separate sources that can produce different and possibly conflicting opinions about the degree of trust for a request x. Therefore, MuSeR mathematically fuses these multi-source assessments; we denote such fusion by $\Omega(\omega_{ip}, \omega_{url}, \omega_{dns})$.

However, it is challenging to identify the correct fusion operator for a specific situation as previous descriptions of subjective opinion fusion have been expressed in terms of just two sources [13]. Furthermore, the different belief fusion operators proposed in the literature vary significantly depending on the purpose and nature of the fusion process [13]. MuSeR opts to accurately score the request by factoring in the degree of evidence and confidence provided by the individual IP, URL and DNS classifiers. In [14], subjective opinion logic has been generalized to analyze belief fusion situations involving an arbitrary number of sources and present cumulative and averaging multi-source belief fusion in the formalism of subjective logic. The Cumulative Belief Fusion (CBF) is appropriate for cases when the amount of independent evidence increases by including additional sources; thus CBF perfectly fits the objective of MuSeR. The basic idea of belief fusion is illustrated in Fig. 5, where the cumulative fused opinion $\Omega(b_x^{\blacklozenge}, d_x^{\blacklozenge}, u_x^{\blacklozenge})$ is expressed as follows:

$$\left\{\begin{aligned} b_x^{\blacklozenge} &= \frac{\sum_{C\in\mathfrak{C}} b_x^C \prod_{C_j\neq C} u_x^{C_j}}{\sum_{C\in\mathfrak{C}}\left(\prod_{C_j\neq C} u_x^{C_j}\right) - (N-1)\prod_{C\in\mathfrak{C}} u_x^C} \\ u_x^{\blacklozenge} &= \frac{\prod_{C\in\mathfrak{C}} u_x^C}{\sum_{C\in\mathfrak{C}}\left(\prod_{C_j\neq C} u_x^{C_j}\right) - (N-1)\prod_{C\in\mathfrak{C}} u_x^C} \\ d_x^{\blacklozenge} &= 1 - \left(b_x^{\blacklozenge} + u_x^{\blacklozenge}\right) \end{aligned}\right. \quad (2)$$

where: $\mathfrak{C} \in \{URL, ip, dns\}$, while $b_x^{\blacklozenge}$, $d_x^{\blacklozenge}$, and $u_x^{\blacklozenge}$ are the belief, disbelief,

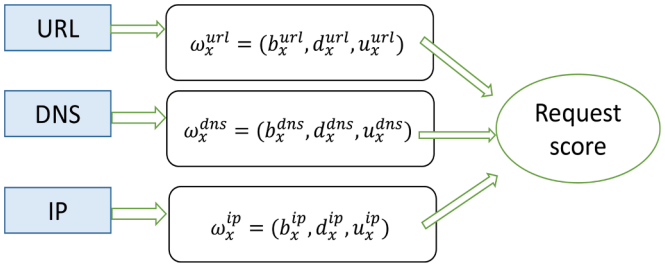


Fig. 5. MuSeR applies cumulative belief to fuse the scores of the individual observables (subjective opinion) within a user request.

and uncertainty for request x, respectively. The intuition behind Eq. (2) is as follows. The belief of $b_x^{\blacklozenge}$ for a given request x reflects the degree of trust that there is no malicious activities within x. The belief for x is expressed as the summation of belief for each observable within x subject to the uncertainty about the other observables. This is further normalized over each combination of possible (N-1) observable uncertainties, where N is the number of observables in x. Meanwhile, the uncertainty of the request score, $u_x^{\blacklozenge}$, is the aggregate of the individual observable uncertainties. Using Eq. (2), the overall trust for request x is based on evidence related to the observables in x. As we have more evidence, the uncertainty for x decreases and consequently, we can have higher certainty (belief + disbelief) as the three components of the subjective logic sum to 1. Overall, the request score is expressive and reflects possible threat caused by the appearance of as little as a single malicious observable as we show in the following Lemma.

Lemma 1. If a score below 0.5 reflects bad reputation and is deemed alarming, the request score will raise alarm in the presence of a single suspicious observable that has the least uncertainty.

Proof: Fundamentally the Lemma asserts that if two observables are good and one is bad with less uncertainty than the two good ones, the request score will be less than 0.5. Let us assume that the first observable is suspicious with a score of less than 0.5 while the other two observables are not, i.e., having a score that exceeds 0.5. That is:

$b_x^1\langle 0.5,\ b_x^2\rangle 0.5,$ and $b_x^3 > 0.5.$

Since $u_x + b_x < 1$, both u_x^2 and u_x^3 are less than 0.5.

Based on the Lemma statement: $u_x^1 < u_x^2 < u_x^3$. Thus, $u_x^1 < 0.5$.

To prove the Lemma, we need to show that $b_x^{\blacklozenge} < 0.5$. Assume that $b_x^{\blacklozenge} = \Delta$, i.e.,

$$b_x^{\blacklozenge} = \frac{b_x^1\left[u_x^2u_x^3\right] + b_x^2\left[u_x^1u_x^3\right] + b_x^3\left[u_x^1u_x^2\right]}{\left[u_x^2u_x^3\right] + \left[u_x^1u_x^3\right] + \left[u_x^1u_x^2\right] - 2u_x^1u_x^2u_x^3} = \Delta$$

Thus,

$$b_x^1\left[u_x^2u_x^3\right] + b_x^2\left[u_x^1u_x^3\right] + b_x^3\left[u_x^1u_x^2\right] = \Delta * \left[\left[u_x^2u_x^3\right] + \left[u_x^1u_x^3\right] + \left[u_x^1u_x^2\right] - 2u_x^1u_x^2u_x^3\right]$$
$$\left[b_x^1 - \Delta\right]\left[u_x^2u_x^3\right] + \left[b_x^2 - \Delta\right]\left[u_x^1u_x^3\right] + \left[b_x^3 - \Delta\right]\left[u_x^1u_x^2\right] = -2\Delta\, u_x^1u_x^2u_x^3$$

Under the assumptions in the Lemma, it suffices to show that the above equation holds for $\Delta < 0.5$.

Let $\Delta = 0.5$, then

$$\left[b_x^1 - 0.5\right]\left[u_x^2u_x^3\right] + \left[b_x^2 - 0.5\right]\left[u_x^1u_x^3\right] + \left[b_x^3 - 0.5\right]\left[u_x^1u_x^2\right] = -u_x^1u_x^2u_x^3$$

Since $b_x^2 > 0.5$, $[b_x^2 - 0.5][u_x^1u_x^3]\rangle 0$; similarly $b_x^3 > 0.5$ implies that $[b_x^3 - 0.5][u_x^1u_x^2]\rangle 0$.

Thus, $[b_x^1 - 0.5][u_x^2u_x^3]\langle -2u_x^1u_x^2u_x^3$, which implies that: $[b_x^1 - 0.5] < -u_x^1$, or $b_x^1 < 0.5 - u_x^1$

Since $u_1 < 0.5$ and $b_x^1 < 0.5$, the above inequality holds and hence $b_x^{\blacklozenge} < \Delta$, for $\Delta = 0.5$

5.2. Session analysis

Given a set of request scores, $\Gamma = \{\Omega_1, \ldots, \Omega_n\}$, we define a session score as the probability that no malicious activities are involved, or equivalently the level of trust that the system has in the user activities as portrayed by the session. Generally, a suspected session could include a mix of benign and malicious activities. The straightforward approach for aggregating the request scores could be based on applying either a cumulative or averaging fusion operator to all Ωi's, in a similar manner to the request scoring. However, such an approach does not handle trust propagation over a dependent set of requests, i.e., inter-request relationships. To illustrate, we can refer to an example where a user accesses "www.google.com" to make a search and ends up visiting some blacklisted or malicious websites that appeared in the search results. Such a pattern differs from visiting the blacklisted website by typing the URL. A subjective logic fusion operator in the form of [*belief* (trust), disbelief (*distrust*), *uncertainty*] does not consider trust transitivity between the original request "www.google.com" and the site whose link appeared in the search. In other words, a subjective logic fusion operator does not consider trust discounting over a dependent set of requests.

To address the aforementioned issue, we formulate the session scoring problem as a trust network [15]. A trust network is modeled as a directed graph (V, E) where a vertex $x \in V$ represents a request, and an edge (x, y) $\in E$ denotes the succession of requests $x$and y. In a trust network, two edges are in series if they are incident to a vertex of degree 2 and are parallel if they join the same pair of distinct vertices. In other words, we build a browsing activity tree, where a vertex in the tree is the request and there is an edge between two vertices if the request for the URL of the child vertex is triggered from the URL of the parent vertex. Note that the dependency between requests is known in the referrer field in the HTTP header. Each sub-path connects a set of requests expressed as a set of subjective opinions. MuSeR measures the degree of trustworthiness of each sub-path and then aggregates their scores to determine the session reputation. The objective is to discount the degree of trust over a particular sub-path and fuse the trustworthiness evidence over a parallel path. To do so, MuSeR defines trust propagation using Three-Valued Subjective Logic (3VSL) [16], where trust is defined as the probability that a trustee (user) will behave as expected by the trustor (system).

Theoretically, the capability of the 3VSL operator is based on the Dirichlet distribution [68] and is shown to be effective for capturing trust propagation in social networks [17]. While trust in social networks is usually reflected as reputation based on ratings, and recommendation preferences, MuSeR focuses on the definition of normal/abnormal behavior in the realm of network security to quantitatively capture user misbehavior based on collected evidence.Unlike subjective logic, the 3VSL operator defines trust as a trinary event (trust, distrust, neutral) instead of a binary event (belief, disbelief). The neutral state expresses the posteriori trustworthiness uncertainty caused by trust propagation, which is ignored in subjective logic. Fundamentally the neutral state keeps the evidence distorted from certain spaces when trust propagates from one entity to another. Thus, the 3VSL distinguishes the posteriori and priori uncertainties existing in trust. Let $\Omega_A = (b_A, d_A,\ n_A, e_A)$ be the trustworthiness vector of a request A, where b_A, d_A, n_A, e_A are, respectively, the belief, disbelief, the posteriori and prioriuncertainty. Using the trustworthiness vector of requests, MuSeR assesses the session score using 3VSL aggregation and the distortion operators. Although the 3VSLoperator captures the trust change over time, it is based on the assumption of known trustworthiness criteria and does not consider the malicious manipulation of trust. To overcome such a shortcoming, we associate the trust/distrust according to the data-driven evidence collected through the previously presented multi-observable analysis. In the following, we highlight how to use the aggregation and distortion operators to monitor dependent and independent user's requests as well as mitigating possible manipulation of trust.

Trust Propagation over Dependent Requests: A trust discounting operator captures the effect of request dependency. Let A and B be two requests and there is a path connecting A and B within the session's trust network model. Then, the trust discounting operation $\Delta(\Omega_A, \Omega_B)$ is carried out as follows [17]:

$$\Delta(\Omega_A, \Omega_B) = \begin{cases} b_{AB} = b_A \times b_B \\ d_{AB} = b_A \times d_B \\ e_{AB} = e_B \\ n_{AB} = 1 - b_{AB} - d_{AB} - e_B \end{cases} \quad (3)$$

Eq. (3) computes the session belief b_{AB}, disbelief d_{AB}, posteriori uncertainty n_{AB}, and priori uncertainty e_{AB}. Clearly the belief of a session is dependent on the belief of requests, i.e., conditional probability of being benign; the session belief will be the product of the belief of the requests,

i.e., equals ($b_A \times b_B$). The disbelief is conditional to the disbelief in the new request (d_B) given the belief in the previous request (A) or session score, is thus equal to ($b_A \times d_B$). The session uncertainty, e_{AB}, is due to the uncertainty on request B. However, the posteriori uncertainty, n_{AB}, that it is initially zero will increase over time to capture the cumulative uncertainties over the previous requests. This is captured by subtracting the degree of belief, disbelief and uncertainty in the current session score. This way the trust will decrease over time because the posteriori uncertainty increases each time. Thus, a user session is trusted by default; however such trust is diminished with bad dependent requests, because we are accumulating evidence and the distrust for each request is added to the posteriori probability.

Similarly, it is important to mention that the posteriori uncertainty exists because of evidence distortions, which is initially equal to zero, for each path of dependent requests and is later updated according to (3). According to the $\Delta(\Omega_A, \Omega_B)$ operator, we have to evaluate the trustworthiness of the sequence of requests ($A \rightarrow B$) in terms of belief, disbelief, priori and posteriori uncertainties. The belief constitutes conditional probability that depends on certainty of both requests. However, the priori uncertainty depends on the lack of evidence, thus it depends on the uncertainty of B since the uncertainty of the request A has been reinforced by the degree of distrust on B. Thus, $\Delta(\Omega_A, \Omega_B)$ discounts B's opinion to obtain the trustworthiness of the sub-path (A-B), some certain evidence from A will be distorted and will be reflected into the posteriori uncertainty of the resulting opinion. However, the priori evidence will keep reflecting the uncertainty in B.

Fig. 6 illustrates the effect of $\Delta(\Omega_A, \Omega_B)$ using the scores of the sequence of dependent requests in Table 1. Basically, the distorted evidence is saved into the posteriori uncertainty space and increases as the disbelief in subsequent requests grows. The discounting operation is analogous to electromagnetic wave propagation where the original signal is distorted into a weak one at the receiving side. Since the uncertainty determined by $\Delta(\Omega_A, \Omega_B)$ is distorted and captured in the posteriori uncertainty space of Ω_B, the same joint evidence (belief and disbelief) among all requests in the sub-paths will be preserved. Therefore, for multi-request distortion path, the resulting opinion of a discounting operation shares exactly the same evidence space with the original opinion. It is also important to note that the trust discounting operation is associative but not commutative. This implies that the order of execution of the distortion operation should reflect exactly the order of issued requests.

Trust Aggregation of Independent Requests: Given the scores for the dependent set of requests (path), MuSeR needs to calculate the expected score for the session while also considering independent requests. The introduction of neutral state makes the operations in 3VSL different from subjective logic, which makes the cumulative multi-source fusion operator impractical [17]. Eq. (4) shows the aggregation operator. The scores for the session are fused in real time as the scores for the independent set of requests are accumulated progressively given that the operator is associative and commutative.

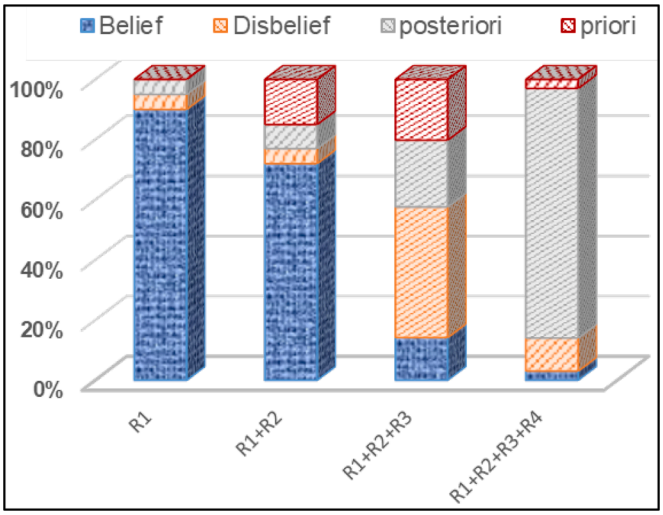


Fig. 6. Illustrating the effect of applying a trust discount operator.

Table 1
Example of fusing request scores within a session.

Score	Request	Trust	Distrust	Uncertainty
0.95	google.com	0.932	0.024	0.05
0.89	yahoo.com	0.866	0.09	0.044
0.37	subuys.com	0.354	0.608	0.039
0.36	subuys.com/6v5r7thh	0.345	0.616	0.039

$$(\Omega_B, \Omega_A) = \begin{cases} b = \dfrac{e_B * b_A + e_A * b_B}{e_A + e_B - e_A * e_B} \\ d = \dfrac{e_B * d_A + e_A * d_B}{e_A + e_B - e_A * e_B} \\ n = \dfrac{e_B * n_A + e_A * n_B}{e_A + e_B - e_A * e_B} \\ e = \dfrac{e_A * e_B}{e_A + e_B - e_A * e_B} \end{cases} \quad (4)$$

According to Eq. (1), the expected score for the session that consists of two independent requests A and B, is: $E_s = b_{\theta(\omega_A, \omega_B)} + a * n_{\theta(\omega_A, \omega_B)}$. To find the belief, b, of the session score, we accumulate the disjoint belief space of requests A and B. This is achieved by considering the conditional probability of the uncertainty of one request over the belief on the other, as shown in Eq. (4). The disbelief, d, is similarly computed. The priori and posteriori uncertainties reflect the conditional probability of the uncertainty for both requests.

Countering Score Manipulation: Therefore, we conclude that to assess session trust, the original request scores are combined by merging the evidence they provide into the final opinion. Although the aforementioned aggregation operator considers the evidence of the independent requests scores, it does not prevent malicious users from manipulating the scoring by alternating patterns of good and bad requests to neutralize the session scores. To illustrate, Fig. 7 reflects the result of the session score for the set of requests indicated in Table 1, assuming that they are independent. It is important to note that for the requests indicated in Table 1, the session is classified as bad using the distortion operator of Eq. (3) with a score of 0.1, while the aggregation operator of Eq. (4) generates a score of 0.54, which implies a neutral session.

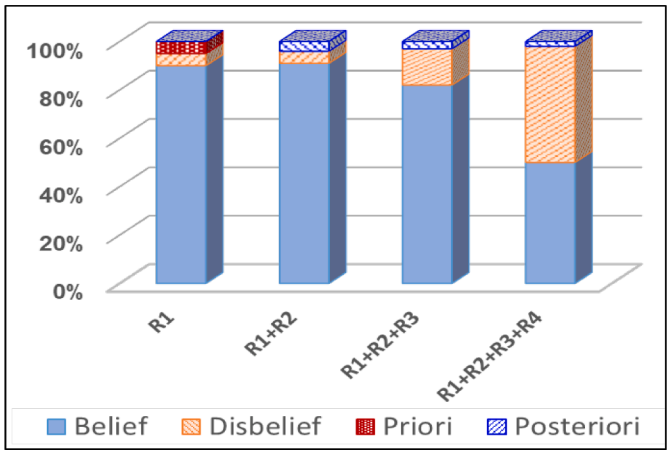


Fig. 7. Illustrating session trust evolution using trust aggregation of our requests, namely, R1, R2, R3, and R4.

To address the aforementioned shortcomings of the aggregation operator, we will introduce a distortion factor μ following each occurrence of a bad request. Such a factor will grow in significance as a function of the number of bad requests. By introducing this factor, the objective is to impose excessive penalties for users that frequently alternate good and bad requests in order to hide the malicious characteristic of the session. In fact, when we consider the set of requests as dependent, we can discount the trust progressively as we have more evidence of distrust. However, when we have a set of independent requests, their evidence is considered equitably when the requests do not have any new bad pattern. Therefore, by applying the distortion factor μ each time a request is classified as suspicious, the aggregator operator will be inflicting a penalty that diminishes the trust. Example of μ settings could be:

$$\mu = \alpha, \text{ or } \mu = e^{-\alpha} \; where \; \alpha < 1$$

After the first application, α could be decreased by a ratio β (i.e., $\alpha = \alpha\,\beta$), in order to grow the significance of the distrust penalty. The penalty could be incorporated in the aggregation operator in (4) as:

$$\begin{aligned} b^* &= b_{AB} * \mu, \\ d^* &= d_{AB} + b_{AB} * (1-\mu), \\ e^* &= e_{AB}, \\ u^* &= u_{AB}, \end{aligned}$$

where b^*, d^*, e^*, u^* are the session belief, disbelief priori and posteriori uncertainties, respectively. It is important to note that the distortion factor varies according to parameters α and β, which are determined by the network/cloud administrator.

6. Pefromance validation

The effectiveness of MuSeR is validated through extensive experimentation. This section discusses the setup, performance metrics and results.

6.1. Simulation environment and performance metrics

We gauge the effectiveness of MuSeR inter-observables analysis using the following contemporary metrics: *precision, recall, F1-score*, and *Receiver Operating Characteristic* (ROC). Since initially we have binary classification, we track the samples that get correctly predicted, denoted by true positive (*TP*) and true negative (*TN*) for the benign or malicious categories, respectively. Analogously, fault positive (*FP*) and fault negative (*FN*) reflect wrongly classified samples. The precision is defined *TP/(TP+FP)* and gauges the classifier's ability to avoid designating a malicious observable as benign. Thus, the smaller the number of *FP* is, the higher the precision becomes. The accuracy is the percentage of *TP* and *TN*. On the other hand, the *Recall* (*Sensitivity*) metric measures the proportion of benign observables that are correctly identified, i.e., *TP/(TP+FN)*. Hence, the smaller the number of *FN* is, the higher the *recall* gets. The F1-score is the harmonic mean of precision and sensitivity, and is calculated as $2TP/(2TP+FN+FP)$. Meanwhile, the *ROC* curve is constructed using the values of true positive rate (*Recall*) and false positive rate. In a perfect scenario, the true positive rate is close to one (no false negatives) and the false positive rate is nearly a zero (no false positives). In the experiments, we compare the performance of the following classifiers: "Linear SVM", "RBF SVM", "Decision Tree", "Neural Net", "AdaBoost", and "Naive Bayes". We compare the MuSeR's multi-observable analysis to contemporary single-observable features used in the literature [52] [61-64]. Using the observable scores, we also have conducted validation of the request and session scores. For that, we use MalRank [45] as a baseline for comparison. MalRank employs a knowledge graph to model the associations among observables and then applies a graph-based inference algorithm to assess a node maliciousness score based on its associations to other nodes (observables) in the graph.

6.2. Experiment setup and results

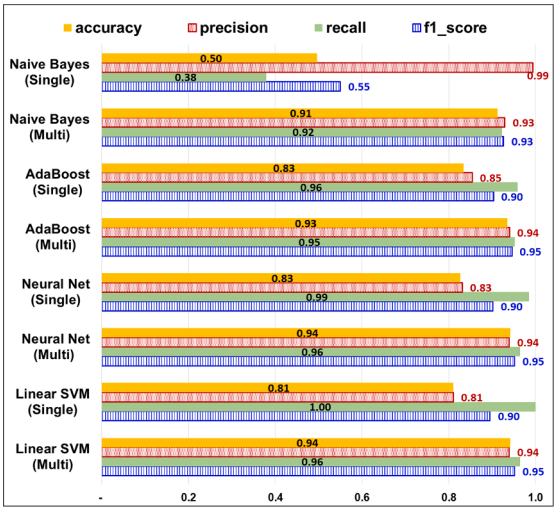
We divide the analysis into two categories, namely, classifier and the score effectiveness.

Classifier performance: The objective of these experiments is to validate our proposed multi-observable analysis for unknown IP, URL and Domains. We assess the effectiveness of the classification compared to a single-observable analysis; basically, we exclude any feature related to other observables. Appendix A enumerates all considered features, and distinguishes those used in the single-observable analysis. We note that all listed features in the appendix are factored in MuSeR's multi-observable analysis, and not just those marked with "Multi". We assess the classification performance using the following three datasets, namely, A, B and C, for the three set of observables domain, IP and URL, respectively. The entries for the domain and the IP have been randomly picked from Alexa and Virus Total [8], Talos [60] and DNSBL [61]. The entries in the URL set are randomly picked from Alexa and from [49], and [50]. The features for each of those entries (observables) are extracted from diverse data sources like WHOIS, CYMUS, and passive DNS databases as explained in detail in data collection (Section IV.A). We have selected the aforementioned databases due to their reliability and popularity so that the scores produced by MuSeR can be validated. The overall dataset contains 7367 domains (Good= 5984, Bad=1383), 94447 IPs (50479 Good and 43968 Bad), 7496 URLs (Good= 2620, Bad= 4876). We have divided randomly our dataset into 80% used as training data and 20% that serves as test data.

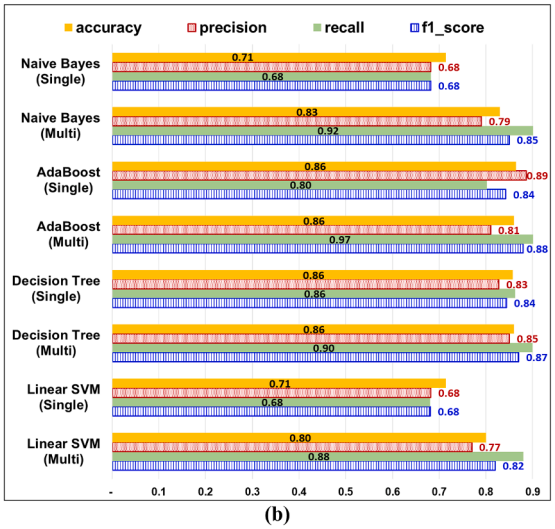
Fig. 8 shows the classifier performance results, where we compare single- and multi-observable analysis. While we have experimented with six classifiers, we are showing the best performing four in each plot for clarity of the presentation. When using the set A, we see over Fig. 8(a) that the best classifier for domains yields accuracy of 94% under MuSeR, which surpasses the best results for a single-attribute analysis, where "AdaBoost" provides an accuracy of 0.83. "Decision Tree" has provided 71% accuracy, which is the worst for domains classification, and consequently "Decision Tree" is not shown in Fig. 8(a). For MuSeR, the "Neural Net" and "Linear SVM" classifiers achieve the highest precision and recall. To factor in the relation between these two metrics, we illustrate the ROC curve in Fig. 9(a). Our evaluation shows that the TP rate jumps very quickly to 94%, which is close to the ideal scenario. The single-attribute analysis has much inferior ROC results.

As indicated by the results for IP classification in Fig. 8(b), the "AdaBoost" classifier leads the way with an accuracy score of 86% and precision of 81%. The "Decision Tree" classifier comes next in this case. Although the MuSeR's multi-observable approach slightly boosts accuracy for these two classifiers compared to the single-observable analysis, MuSeR significantly improves their precision; on the other hand, both the accuracy and precision for the "Naïve Bayes" and "Linear SVM" classifiers, have experienced major improvements through MuSeR, as shown in Fig. 8(b). Unlike DNS, IP classification does not involve many multi-observable features as evident from Appendix A. When comparing the ROC results in Fig. 9(b), we also observe higher precision and recall values for "AdaBoost" and "Decision Tree", where the curve reaches close to 1 at low FP rate, as indicated in Fig. 9(b). Note also that the multi-observable analysis consistently improves the classification performance. Even though the precision or recall measures are high for some of single-observable classifiers, the corresponding F1-score reflects imbalance between the precision and recall values.

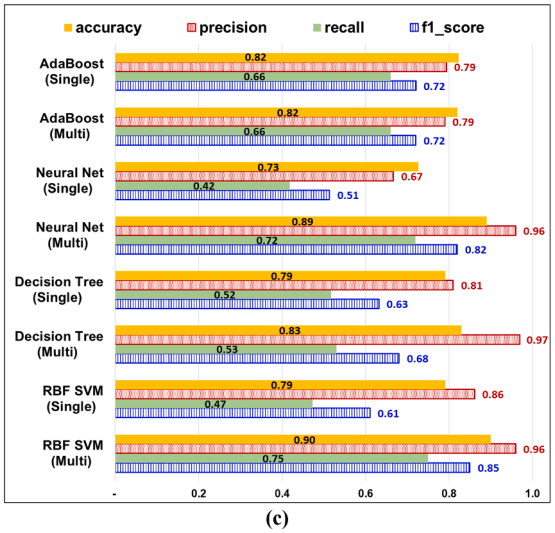
Meanwhile Figs. 8(c) and 9(c) report the evaluation results for the URL classification. For MuSeR, the highest accuracy has been accomplished by "RBF SVM", while "Neural Net" follows very closely. The best ROC value performance is achieved by "RBF SVM". We can clearly see the impact on using MuSeR as URL features are mainly syntactical. Basically, the multi-observable analysis involves more context for the names of domain and subdomains within the URL. We note that MuSeR considers the results of the various classifiers as evidence with some degree of uncertainty and applies subjective logic to fuse these results to



(a)



(b)

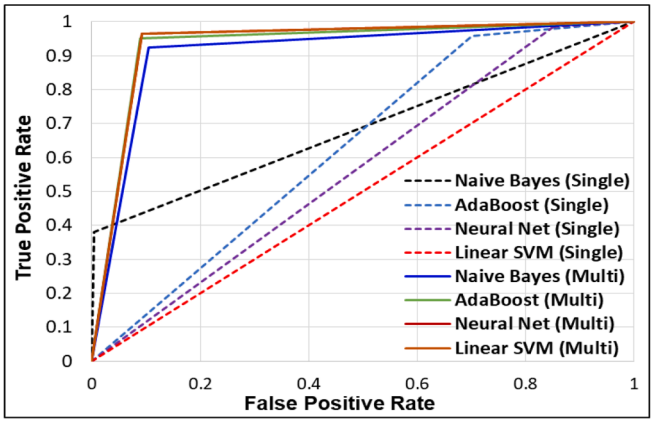


(c)

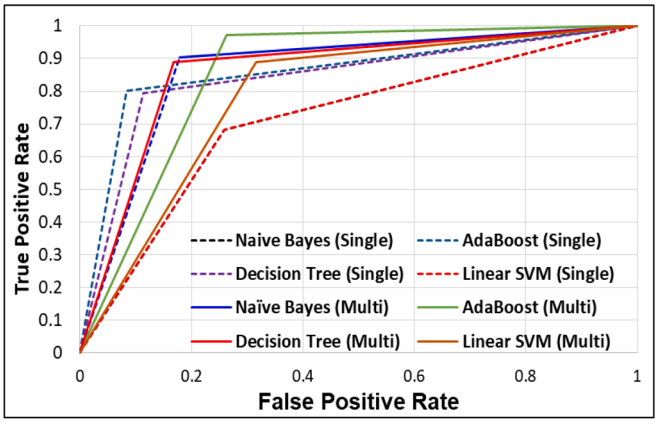
Fig. 8. Classifier effectiveness for (a) domains, (b) IP, and (c) URL.

assign a request score, as validated next.

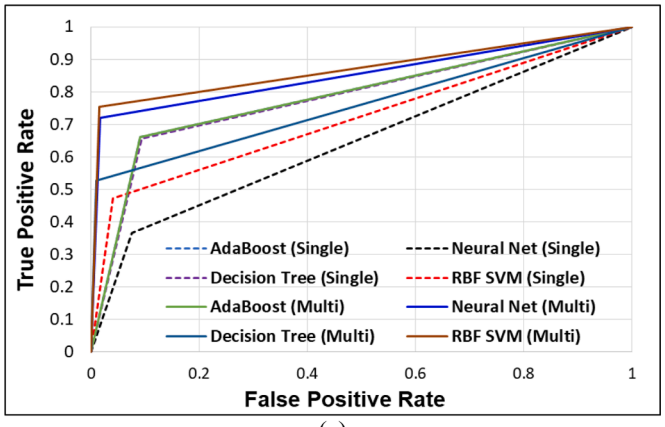
Request score expressiveness: The objective of this experiment is to validate the effectiveness of our request scoring. The observables within the same request may have conflicting scores according to the evidence and the similarities of their features to known labeled observables



(a)



(b)



(c)

Fig. 9. Receiver Operating Characteristics (ROC)for (a) domains, (b) IP, and (c) URL.

within their respective category. MuSeR factors in the degree of evidence to come up with a consistent score for the entire user request. To assess the expressiveness of request scoring, we have performed experimentation over sample requests for which we have manipulated the malicious characteristics of some observables. For a given request within the sample, we have varied the malicious behavior by switching the degree of trust and distrust for one of the involved observables. In order to capture the effect of the degree of uncertainty, we have manipulated the uncertainty of the malicious observable so that it has the least, middle and most uncertainty in the request. The results are reported in Fig. 10, where three levels of uncertainty are shown and compared to the "original" non-manipulated score. Based on the results, we can note that when the malicious observable has the highest degree of evidence (least

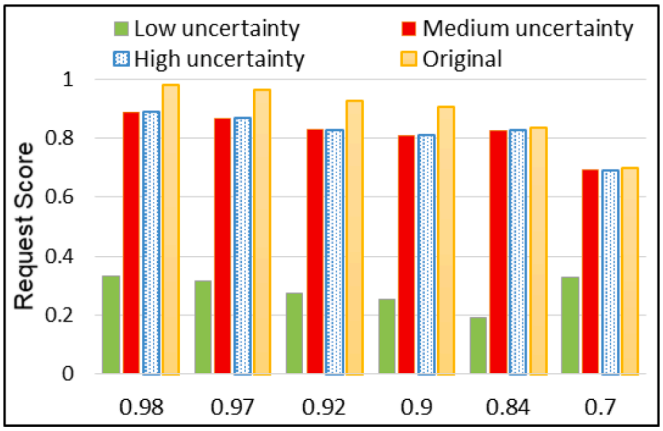


Fig. 10. The request score diminishes the most when associated with low uncertainty. The baseline is shown as "Original", and reflects the case where the malicious characteristics of observables are not manipulated.

uncertainty) the request score drops significantly. This is expected and is consistent with Lemma 1.

Session scoring analysis:To illustrate the efficiency of the session scores, we have used a sample of good and bad requests. We have studied the effect of alternating good and bad requests on the session score. Basically, we vary the number of occurrences of good requests following each bad request, which not only captures the effect of frequency of malicious requests but also how forgiving the trust assessment is to suspicious activities. The objective is to show how expressive the scores of MuSeR are and the effect of trust distortion. We have experimented using two trust distortion functions, namely multiplicative and exponential, as indicated in Section V-B. Fig. 11(a) and (b) show the results for multiplicative and exponential distortion factors, respectively. The value α and β are set to 0.8, and 0.7, respectively. The x-axis reflects the number of good requests made after each bad one. The figure shows how the belief increasingly gets distorted based on the multiplicity of the bad requests in the session. Fig. 11(a) implies that the multiplicative function is more tolerant to the first bad request and becomes more penalizing with repeated occurrence. On the other hand, as shown in Fig. 11(b) the exponential function inflicts high penalty on the first bad request in the session and slowly grows the trust distortion level as more bad requests are made.

On the other hand, the baseline approach, MalRank, also seems to decrease the scores significantly with high frequency of bad requests. This is evident from Fig. 11(c). However, when the frequency of bad requests diminishes, the performance of MalRank drops significantly, meaning that it provides scores that do not much reflect suspicion in the session.Thus, a misbehaving user can manipulate the session by ensuring that the number of good requests consistently exceeds the bad ones. In addition, MalRank does not reflect the severity of the situation when the number of occurrences varies; for example the MalRank score for a combination of 2 good and 1 bad requests is similar to having 4 good followed by 2 bad ones. In summary, good requests in MalRank always neutralize the bad ones in the same way, which introduces vulnerability. Consequently, MalRank fails to make any alarming observation and can be fooled. On the other hand, given the evidence based trust assessment, MuSer session scores stay indicative of the malicious activities even if the attacker increases the number of good requests to evade suspicion.

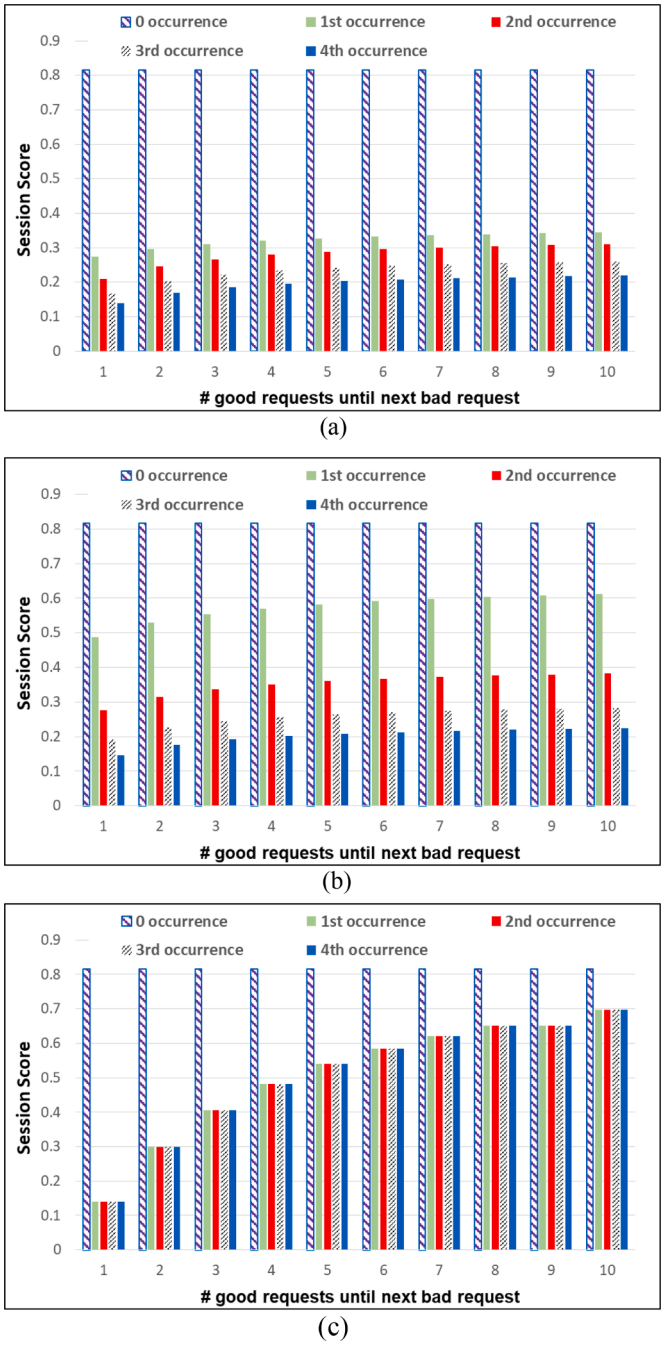


Fig. 11. The effect of the distortion function on session scores as a function of the frequency of bad requests, i.e., how many good requests follow a bad request.

7. Conclusion

In this paper, we have presented MuSeR, a novel approach for user session reputation scoring. MuSeR factors in evidence from known blacklists and suspicious use patterns of observables to classify the user requests. In essence, MuSeR assists a network/cloud administrator in determining the trustworthiness of a user session based on the navigation pattern and the dynamic analysis of individual observables involved within requests in that session. Specifically, MuSeR employs a machine learning model using features that are carefully chosen to factor in evidence provided by blacklists, and access patterns of known attacks. To determine reputation scores for observables, MuSeR maps the classifier probabilities into subjective logic and then uses cumulative fusion to calculate user request scores. Given the request scores, MuSeR applies an adaptive version of three-valued subjective logic to handle trust propagation and aggregation over user requests.MuSeR has been subject to extensive evaluation using data from existing databases. The evaluation results have demonstrated that MuSeR provides high accuracy, detects unknown malicious observables, and outperforms competing approaches in the literature. In addition, the utility of request

and session scores is analyzed. For the future work, we would like to extend MuSeR to alert the presence of collusive malicious activities across multiple user sessions. The idea is to subject sessions with low scores to further analysis that correlates the involved observables and checks whether there is coordination about the malicious attempts of the involved users.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work is supported by Cisco under contract # 12430. The authors like to thank Dr. Yatish Joshi, Mr. Pramod Chandrashekar of Cisco, as well as Dr. Karuna Joshi, and Dr. Vandana Janeja of UMBC for their feedback and fruitful discussion.

Appendix A

DNS Features

- Single: Lifetime of the domain
- Single: Freshness of the domain requests
- Single: Idle time (Time from the last request)
- Single: Number of countries to which the ASN belongs
- Single: The TTL of the name server
- Single: The ratio of active name servers
- Single: The TTL of A and AAA records
- Single: The min, max and avg. TTL, serial, refresh, retry, and "expire" of the individual SOA records
- Single: The number of CNAME records
- Single: The number of TXT records
- Single: NS check if valid name server
- Single: The number of NS records
- Single: The popularity rank of the domain
- Single: Country rank
- Multi:Number of IPs associated with the domain
- Multi:Number of blacklisted IPs within the domain
- Multi:Number of Popular IPs within the domain
- Multi:Max popularity/BGP ranking of IPs
- Multi:Min popularity ranking/BGP of IPs BGP
- Multi:AVG popularity ranking/BGP of IPs
- Multi:Number of ASNs

IP Features

- Single: Number of packet blocked for an IP
- Single: Number of unique destinations from an IP
- Single: Number of observed attacks using an IP
- Single: IP freshness (time since an IP was last seen)
- Single: Number of target addresses hit from an IP
- Single: Time since the last reported attack
- Single: Number of packets originating from an IP.
- Single: IP lifetime
- Single: Host ranking (Alexa popularity)
- Single: Rank of the country
- Single: City associated with geolocation of IP address
- Single: IP type (Organization/individual)
- Single: IP lifetime
- Multi:Size and BGP ranking of the associated ASN
- Multi:Registrar qualification (Suspicious or trusted)

URL Features

- Single: URL length
- Single: Number of tokens in the URL
- Single: The occurrence of any IP
- Single: AVG token length of host, domain and path
- Single: MAX tokenlength of host, domain and path
- Single: Presence of security sensitive words
- Single: Number of meaningful words
- Single: Safe-browsing lookup
- Multi:Domain popularity ranking
- Multi:Rank of the country
- Multi:ASN number rank
- Multi:IP address is associated with the hostname

References

[1] X.Han, N.Kheir, and D.Balzarotti. "The Role of Cloud Services in Malicious Software: Trends and Insights," in the Proceedings of the 12th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment - Volume 9148 (DIMVA 2015), Magnus Almgren, Vincenzo Gulisano, and Federico Maggi (Eds.), Vol. 9148. Springer-Verlag, Berlin, Heidelberg, pp. 187-204, 2015.
[2] P.Schoo, V.Fusenig, V.Souza, M.Melo, P.Murray, H.Debar, H.Medhioub, and D. Zeghlache, "Challenges for cloud networking security," HP Laboratories Technical Report, No. 137, pp. 1-17, 2010.
[3] D.Bestuzhev, "Financial data stealing malware now on amazon web services cloud," https://securelist.com/financial-data-stealing-malware-now-on-amazon-web-services-cloud/30647/, 2011. [accessed Oct-10-2018].
[4] R.Unuchek, "Gcm in malicious attachments," http://www.securelist.com/en/blog/8113/GCM_in_malicious_attachment, 2013. [Oct-10-2019].
[5] J. Vacca, Managing Information Security, 2nd edn., Elsevier, USA, 2014.
[6] J. Nazario, T. Holz, As the net churns: Fast-flux botnet observations, in: the Proceedings of the 3rd IEEE International Conference on Malicious and Unwanted Software (MALWARE 2008), Alexandria, Virginia, October 2008.
[7] WatchGuard - Reputation Authority -http://www.reputationauthority.org.
[8] McAfee (Threat Intelligence -https://www.mcafee.com•IP-ADDRESS.com -http://ip-address.com.
[9] Virus Total -https://virustotal.com.
[10] R.Ismail Jøsang, C. Boyd, A survey of trust and reputation systems for online service provision, Deci. Supp. Syst. 43 (2) (2007) 618–644.
[11] M. Antonakakis, et al., From Throw-Away Traffic to Bots: Detecting the Rise of DGA-Based Malware, in: the Proceedings of the 21st Usenix Security Symposium, Bellevue, WA, August 2012.
[12] L. Bilge, E. Kirda, C. kruegel, M. Balduzzi, Exposure: Finding malicious domains using passive dns analysis, in: the Proceedings of the 18th Network and Distributed System Security Symposium (NDSS 2011), San Diego, CA, February 2011.
[13] Jøsang, Subjective Logic: A formalism for reasoning under uncertainty, Springer Verlag, Switzerland, 2016.
[14] G. Liu, Q. Yang, H. Wang, X. Lin, M. Wittie, Assessment of multi-hop interpersonal trust in social networks by three-valued subjective logic, in: Proceedings of the 33rd IEEE International Conference on Computer Communications (INFOCOM), 2014, pp. 1698–1706.
[15] J.-H. Cho, K. Chan, S. Adali, A Survey on Trust Modeling, ACM Computing Surveys (CSUR) 48 (2) (November 2015) 1–40.
[16] D. Ciucci, D. Dubois, Three-Valued Logics, Uncertainty Management and Rough Sets, Transactions on Rough Sets XVII, Lecture Notes in Computer Science 8375 (2014) 1–32.
[17] G.Liu, Q.Yang, H.Wang, X.Lin and M. P.Wittie, "Assessment of multi-hop interpersonal trust in social networks by Three-Valued Subjective Logic," IEEE INFOCOM 2014 - IEEE Conference on Computer Communications, Toronto, ON, 2014, pp. 1698-1706, 10.1109/INFOCOM.2014.6848107.
[18] S. Sinha, M. Bailey, F. Jahanian, Shades of grey: On the effectiveness of reputation-based blacklists, in: the Proceedings of the 3rd IEEE International Conference on Malicious and Unwanted Software (MALWARE 2008), Alexandria, Virginia, October 2008.
[19] J. Zhang, P. Porra, J. Ullrich, Highly predictive blacklisting, in: in the Proceedings of the 17th Usenix Security Symposium, San Jose, CA, July 2008.
[20] K. Sato, K. Ishibashi, T. Toyono, N. Miyake, Extending black domain name list by using co-occurrence relation between dns queries, in: the Proceedings of the 3rd USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET'10), San Jose, CA, April 2010.
[21] M. Felegyhazi, C. Keibich, V. Paxson, On the potential of proactive domain blacklisting, in: the Proceedings of the 3rd USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET'10), San Jose, CA, April 2010.
[22] H. Choi, H. Lee, H. Kim, Botnet detection by monitoring group activities in DNS Traffic, in: the Proceedings of the 7th IEEE International Conference on Computer and Information Technologies, Aizu-Wakamatsu, Fukushima, Japan, October 2007.
[23] R. Villamarn-Salomn, J.C. Brustoloni, Bayesian bot detection based on DNS traffic similarity, in: the Proceedings of the ACM symposium on Applied Computing (SAC'09), Honolulu, HI, March 2009.
[24] W. Li, J. Jin, J. Lee, Analysis of Botnet Domain Names for IoT Cybersecurity, IEEE Access 7 (2019) 94658–94665, https://doi.org/10.1109/ACCESS.2019.292735.

[25] W.X. Zang, J. Gong, S. Mo, A. Jakalan, D. Ding, Identifying Fast-Flux Botnet With AGD Names at the Upper DNS Hierarchy, IEEE Access 6 (2018) 69713–69727, https://doi.org/10.1109/ACCESS.2018.2880884.
[26] S. Garera, N. Provos, M. Chew, A. Rubin, A framework for detection and measurement of phishing attacks, in: the Proceedings of the ACM workshop on Recurring Malcode (WORM'07), Alexandria, VA, November 2007.
[27] T. Holz, C. Gorecki, K. Rieck, F.C. Freiling, Measuring and detecting fast-flux service networks, in: the Proceedings of the 15th Network and Distributed System Security Symposium (NDSS 2008), San Diego, CA, February 2008.
[28] D. Anderson, C. Fleizach, S. Savage, G. Voelker, Spamscatter: Characterizing internet scam hosting infrastructure, in: the Proceedings of the 15th Usenix Security Symposium, Santa Clara, CA, June 2007.
[29] S. Hao, N. Syed, N. Feamster, A. Gray, S. Krasser, Detecting spammers with SNARE: Spatiotemporal network-level automatic reputation engine, in: the Proceedings of the 17th Usenix Security Symposium, San Diego, CA, June 2009.
[30] Z. Qian, Z. Mao, Y. Xie, F. Yu, On network level clusters for spam detection, in: the Proceedings of the 17th Network and Distributed System Security Symposium (NDSS 2010), San Diego, CA, February 2008, 2010.
[31] E. Passerini, R. Paleari, L. Martignoni, D. Bruschi, Fluxor: Detecting and monitoring fast-flux service networks, in: the Proceedings of The International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA 2008), D. Zamboni (eds), Lecture Notes in Computer Science, Vol 5137. Springer, Berlin, Heidelberg, 2008.
[32] N.Feamster, M.Konte, and J.Jung. "Fast Flux Service Networks: Dynamics and Roles in Hosting Online Scams," Technical Reports GT-CS-08-07, School of Computer Science, Georgia Tech University, 2008.
[33] J. Ma, L.K. Saul, S. SAVAGE, G.M. Voelker, Beyond blacklists: Learning to detect malicious web sites from suspicious urls, in: the Proceedings of the SIGKDD Conference, Paris, France, June 2009.
[34] F. Weimer, Passive DNS Replication, in: the Proceedings of 17th Annual FIRST Conference, Singapore, 2005.
[35] K. Dan, N. Kitagawa, S. Sakuraba, N. Yamai, Spam Domain Detection Method Using Active DNS Data and E-Mail Reception Log, in: the Proceedings of the IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), Milwaukee, WI, 2019, pp. 896–899, https://doi.org/10.1109/COMPSAC.2019.00133.
[36] B. Zdrnja, N. Brownlee, D. Wessels, Passive Monitoring of DNS Anomalies, in: the Proceedings of the 4th international conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA '07), Bernhard Hämmerli and Robin Sommer (Eds.). Springer-Verlag, Berlin, Heidelberg, 2007, pp. 129–139.
[37] R. Perdisci, I. Corona, D. Dagon, W. Lee, Detecting Malicious Flux Service Networks through Passive Analysis of Recursive DNS Traces, in: the Proceedings of the 25th Annual Computer Security Applications Conference (ACSAC'09), Honolulu, Hawaii, December 2009.
[38] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, N. Feamster, Building a Dynamic Reputation System for DNS, in: the Proceedings of the 19th Usenix Security Symposium, Washington, DC, August 2010.
[39] M. Antonakakis, R. Perdisci, W. Lee, N. Vasiloglou, D. Dagon, Detecting Malware Domains at the Upper DNS Hierarchy, in: the Proceedings of the 20th Usenix Security Symposium, San Francisco, CA, August 2011.
[40] S. Torabi, A. Boukhtouta, C. Assi, M. Debbabi, Detecting Internet Abuse by Analyzing Passive DNS Traffic: A Survey of Implemented Systems, IEEE Commun. Surv. Tutor. 20 (4) (2018) 3389–3415, https://doi.org/10.1109/COMST.2018.2849614, 4th Quarter.
[41] T.Yu Khalil, B. Guan, Discovering malicious domains through passive DNS data graph analysis, in: the Proceedings the 11th ACM on Asia Conference on Computer and Communications Security (ASIA CCS '16), May 2016, pp. 663–674. Xi'an, China.
[42] M Khalil, B. Guan, M. Nabeel, T. Yu, A Domain is only as Good as its Buddies: Detecting Stealthy Malicious Domains via Graph Inference, in: the Proceedings of the 8th ACM Conference on Data and Application Security and Privacy (CODASPY '18), March 2018, pp. 330–341. Tempe, AZ.
[43] P. Najafi, A. Sapegin, F. Cheng, C. Meinel, Guiltby-Association: Detecting Malicious Entities via Graph Mining, in: the Proceedings of the International Conference on Security and Privacy in Communication Systems (SecureComm 2017), 2017, pp. 88–107. Niagara Falls, Canada.
[44] E.Choo, and M.Nabeel, M.Alsabah, I.Khalil, T.Yu, and W.Wang, "DeviceWatch: Identifying Compromised Mobile Devices through Network Traffic Analysis and Graph Inference," arXiv:1911.12080v1, 2019.
[45] P. Najafi, A. Mühle, W. Pünter, F. Cheng, C. Meinel, MalRank: a measure of maliciousness in SIEM-based knowledge graphs, in: the Proceedings of the 35th Annual ACM Computer Security Applications Conference (ACSAC'19), December 2019, pp. 417–429. San Juan, PR.
[46] C. Peng, X. Yun, Y. Zhang, S. Li, J. Xiao, Discovering malicious domains through alias-canonical graph, in: the Proceedings of the IEEE Trustcom-BigDataSE-ICESS Conference, September 2017, pp. 225–232. Sydney, Australia.
[47] L. Watkins, S. Beck, J. Zook, A. Buczak, J. Chavis, W.H. Robinson, J.A Morales, S. Mishra, Using semi supervised machine learning to address the Big Data problem in DNS networks, in: the Proceedings of the 7th Annual IEEE Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, March 2017.
[48] Q.Fu Lei, J. Ni, F. Wang, M. Yang, K. Xu, Detecting Malicious Domains with Behavioral Modeling and Graph Embedding, in: the Proceedings of the IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, 2019, pp. 601–611, https://doi.org/10.1109/ICDCS.2019.00066.
[49] Z. Ma, Q. Li, X. Meng, Discovering Suspicious APT Families Through a Large-Scale Domain Graph in Information-Centric IoT, IEEE Access 7 (2019) 13917–13926, https://doi.org/10.1109/ACCESS.2019.2894509.
[50] Jeffrey Carr (2007-06-05). "Snort: Open Source Network Intrusion Prevention". Retrieved 2010-06-23.
[51] "RFC 8162 - Using Secure DNS to Associate Certificates with Domain Names for S/MIME," Internet Engineering Task Force. May 2017. Retrieved 17 October 2018.
[52] H. Liu, X. Pan, Z. Qu, Learning based Malicious Web Sites Detection using Suspicious URLs, in: the Proceedings of the 34th International Conference on Software Engineering, Zurich Switzerland, June 2012.
[53] Alexa Web Information Company. http://www.alexa.com/topsites/.
[54] Google Safe Browsing, http://www.google.com/tools/firefox/safebrowsing/.
[55] https://github.com/stamparm/maltrail.
[56] https://github.com/cbuijs/ut1.
[57] RUS-CERT. DNS replication. http://cert.uni-stuttgart.de/dienste/dns-replication.html, 2011.https://www.farsightsecurity.com/solutions/dnsdb/.
[58] RFC1834, "Whois and Network Information Lookup Service, Whois++," 1995, http://www.faqs.org/rfcs/rfc1834.html.
[59] P. Albitz, C. Liu, DNS and BIND, 5th Edition, Nutshell Series, O'Reilly and Associates, Inc., 2006.
[60] S. Hao, M. Thomas, V. Paxson, N. Feamster, C. Kreibich, C. Grier, S. Hollenbeck, Understanding the domain registration behavior of spammers', in: the Proceedings of the 2013 conference on Internet measurement conference (IMC '13), 2013, pp. 63–76, https://doi.org/10.1145/2504730.2504753.
[61] C. Zhou, K. Chen, X. Gong, P. Chen, H. Ma, Detection of fast-flux domains based on passive DNS analysis, Acta Scientiarum Naturalium Universitatis Pekinensis 52 (3) (2016) 396–402.
[62] Bilge, Exposure: a passive dns analysis service to detect and report malicious domains, ACM Trans. Inform. Syst. Secur. (TISSEC) 16 (4) (2014). #14.
[63] H. Liu, et al., On the effects of registrar level intervention, in: the Proceedings of the 4th USENIX Workshop on Large-Scale Exploits and Emergent Threats, Boston, MA, March 2011.
[64] http://isc.sans.edu/api.
[65] Team Cymru. IP to ASN mapping. http://www.team-cymru.org/Services/ip-to-asn.html, 2011.
[66] https://www.talosintelligence.com/.
[67] DNSBL - Spam Database Lookup. http://www.dnsbl.info/. 2010.
[68] S. Kotz, N. Balakrishnan, N.L. Johnson, Continuous Multivariate Distributions, Volume 1: Models and Applications, Wiley, New York, 2019. ISBN 0-471-18387-3Chapter 49: Dirichlet and Inverted Dirichlet Distributions.

Wassila Lalouani is currently associated to the Embedded Systems and Networks lab at the University of Maryland Baltimore County. Prior to that, she was a lecturer at the High National School of Computing Science, Algiers, Algeria. She received her BS, MS and Ph.D degrees from the Department of Computer Science of the University of science and technologies Houari Boumediene (USTHB), Algiers, Algeria. Her research interest include security, and network architectures and protocols.

Mohamed F. Younis is currently a professor in the department of computer science and electrical engineering at the university of Maryland Baltimore County (UMBC). Before joining UMBC, he was with the Advanced Systems Technology Group, an Aerospace Electronic Systems R&D organization of Honeywell International Inc. While at Honeywell he led multiple projects for building integrated fault tolerant avionics and dependable computing infrastructure. He also participated in the development of the Redundancy Management System, which is a key component of the Vehicle and Mission Computer for NASA's X-33 space launch vehicle. Dr. Younis' technical interest includes network architectures and protocols, wireless sensor networks, fault tolerant computing, secure communication and cyber-physical systems. He has published over 280 technical papers in refereed conferences and journals. Dr. Younis has seven granted and three pending patents. In addition, he serves/served on the editorial board of multiple journals and the organizing and technical program committees of numerous conferences. Dr. Younis is a senior member of the IEEE and the IEEE communications society.