Using CyberScore for Network Traffic Monitoring

Luca Deri
ntop
Pisa, Italy
deri@ntop.org

Alfredo Cardigliano
ntop
Pisa, Italy
cardigliano@ntop.org

***Abstract*—The growing number of cybersecurity incidents and the always increasing complexity of cybersecurity attacks is forcing the industry and the research community to develop robust and effective methods to detect and respond to network attacks. Many tools are either built upon a large number of rules and signatures which only large third-party vendors can afford to create and maintain, or are based on complex artificial intelligence engines which, in most cases, still require personalization and fine-tuning using costly service contracts offered by the vendors.**

This paper introduces an open-source network traffic monitoring system based on the concept of cyberscore, a numerical value that represents how a network activity is considered relevant for spotting cybersecurity-related events. We describe how this technique has been applied in real-life networks and present the result of this evaluation.

***Index Terms*—Security Score, Deep Packet Inspection, Network Intrusion Detection, Traffic Measurement, Open-Source.**

I. Introduction and Motivation

Cybersecurity is a hot topic as the number of incidents and attacks are constantly increasing [1], and companies need to deploy and maintain security systems able to protect relevant assets in compliance with law regulations [2]. Even on private networks protected by firewalls or other protection systems such as host-based EDRs (Endpoint Detection and Response), security incidents can still happen and thus it is compulsory to operate cybersecurity-aware monitoring systems able to detect suspicious activities and thus block attacks. As many cybersecurity tools are designed just for security, they often offer poor network visibility as their only focus is to detect threats, with constant network administrator's supervision. This is because many of them are signature-based, and thus can only detect those activities for which there is a configured rule, and have no knowledge of available network services as they are basically network sensors that have no global network knowledge. AI (Artificial Intelligence) based systems [3] are theoretically more sophisticated than rule-based systems as they can detect host behavioural changes, but in addition to being usually quite expensive, they are often offered as a service as these systems and not open. Furthermore, tuning learning parameters is often an activity that only the vendor, and not end-users, can do, as these systems often require supervised maintenance to limit the occurrence of false positives.

The limitations of signature-based systems, and the complexity and cost of AI-based tools, have been the motivation for developing a novel method able to combine behavioural traffic analysis and encrypted network traffic analysis via DPI (Deep Packet Inspection), which is based on statistical methods and thus light and simple to operate in comparison to AI-based systems [6]. ntopng [4] is an open-source network traffic monitoring system developed by the authors, that leveraging on nDPI [5], an open-source DPI toolkit, can inspect and analyze network traffic, detect security issues and track host activities. In ntopng we have implemented the concept of *cyberscore* that is a numerical relevance indicator assigned to every observed network activity: the higher the value is, the higher the severity of this activity. Every network flow is inspected using nDPI, which can detect and report a set of flow risks whenever an unexpected issue, such as an expired TLS certificate, a DGA/Punycode/IDN domain name or credentials transfer in clear text, is detected. To date, nDPI support 45 different flow risks classification for both clear-text and encrypted traffic. In addition, ntopng performs network traffic analysis through checks that are executed on various entities including (but not limited to) flows, hosts and networks to implement the high-level logic. For instance, network scan detection and unusual host behaviour cannot be implemented at the flow level, but at the host level through checks that take into account flows generated by hosts, and use some logic to decide whether the observed traffic is suspicious.

A cyberscore is thus assigned to flows, based on the detected risks and traffic checks in ntopng. A cyberscore is also computed for hosts, which is, at any given time, the sum of all active flows cyberscore and the cyberscore of all alerts triggered by host checks that are still active. The cyberscore value can be used to further trigger alerts when it crosses a threshold or when its value changes compared to past values, which usually indicates a change of behaviour.

The goal of this paper is to show how cyberscore has been effectively used to monitor real networks, report the findings, evaluate the results, and position this technique against the results obtained on the same network with other AI-based tools. We highlight that the use of cyberscore is pretty lightweight in terms of resources, produces good results and it does not require annotated data or remote expert assistance, typical of most commercial AI-based tools.

II. Using Cyberscore to Monitor Network Traffic

As stated before, the cyberscore is a numerical value assigned to every monitored entity such as flows, hosts,

978-1-6654-9952-1/22/$31.00 ©2022 IEEE

Autonomous Systems, VLANs etc. that is used to define how the behaviour of such entity is malicious in term of cybersecurity. The monitoring system deployed close to the network gateway in order to be able to observe all traffic entering/leaving the network. As DPI works best with network packets, it is desirable to feed the system through a span port as this enabled nDPI to analyze packet payload and detect issues such as expired or invalid TLS certificates. When this is not possible, the system can also analyze network flows (i.e. NetFlow/IPFIX or sFlow) with the limitation that packet inspection will not be possible, and thus application protocol needs to be guessed (e.g. using IP address, protocol and ports) as well checks on the packet payload are not possible and thus the system is less effective.

Whenever nDPI detects am unexpected condition, it labels the flow with a bitmap, where each bit is an indication of a specific issue named flow risk. To date nDPI support over 45 flow risks that belong to the following families:

- Suspicious Data Transfer (e.g. binary application transfer).
- Data Exfiltration (e.g. over ICMP and DNS).
- Unexpected Traffic (e.g. DNS packets larger than 512 bytes, TLS traffic with no SNI).
- Alerts based on communication with a remote host present on a third-party blacklist (e.g. Cisco Thalos or Abuse.ch).
- Suspicious Traffic (e.g. suspicious HTTP user-agent or unidirectional unicast UDP traffic).
- Elephant (i.e. large uploads/downloads) or Long-Lived Flows.
- Insecure or Obsolete Protocol versions (e.g. TLSv1 or obsolete SSH client version).

Each nDPI risk has a cyberscore value in the range 10 to 250 and it identifies the severity of the identified risk. Lowest risk values are assigned to minor issues such HTTP requests sent to a numerical (instead of symbolical as it should be) server or DNS request for an IDN name. Middle risk values, in the range from 50 to 100, are assigned to not severe issues such as communications with packets containing clear-text credentials, whereas high risks have the highest cyberscore value of 250 such as in case of binary application transfer. Due to space constrains, the table below reports a subset of the risks supported by the system.

TABLE I
SOME NDPI FLOW RISKS

Flow Risk	Cli Score	Srv Score
XSS Attack	225	25
SQL Injection	225	25
Binary Application Transfer	125	125
Known Protocol on Non Standard Port	25	25
TLS Expired Certificate	50	50
TLS Certificate Mismatch	50	50
SMB Insecure Version	90	10
TLS Suspicious ESNI Usage	25	25
Clear-Text Credentials	90	10

The flow cyberscore is the sum of the client and server cyberscore that is bound to the flow peers. This is to split the detected risk according to the identified issue. For most risks, the value is equally split, but for others such as XSS attack the server has a low value compared to the client, that instead is the attack originator. Using this algorithm, the flow cyberscore is distributed to hosts, and recursively to VLANs, Autonomous Systems, and Networks to which such hosts belong. The host cyberscore is a dynamic value that is computed as the sum of the active flows score. As flows expire when idle or terminated, the host cyberscore decreases as flows are purged from memory. Note that a flow can have multiple flow risks associated and thus the total flow risk can exceed the value of 250.

In addition to the flow risk score, for each monitored entity we have defined the concept of check that sit on top of the flow checks, designed to verify the behaviour of the monitored entity. Host checks families include:

- Threshold-based Alerts (e.g. a non-NTP server host that contacts over 4 different NTP servers, or a host that contacts too many hosts/ASNs/countries in the last minute).
- Network and Port Scan Detection.
- Behavioural Alerts (e.g. a host has an unusual number of client/server flows with respect to its recent past).

Whenever a flow or host violates a check, an alert containing a non-zero cyberscore value is triggered, whose value depends on the severity of the alert and host role. For instance, an alert for a local (i.e. a host belonging to the service provider) host that contacts a blacklisted host has a higher cyberscore than an alert for a local host being contacted by a remote blacklisted host: in the first case, the contact was voluntary and thus might be due to a malicious software running on the alerted host, contrary to the second case that was not predictable.

The cyberscore principle is effective if the flow risk and checks are not increased when not necessary. To avoid false positives or adding noise to the calculation, it is important to do an initial system exception setup, to silence checks for communications that are unpleasant but considered acceptable such as a server that uses a self-signed TLS certificate. With cyberscore there is no need to perform a training/annotation as with AI-based systems, but it is necessary to silence specific checks to avoid unexpected noise that will influence the host cyberscore value. For behavioural alerts, it is also not necessary to train the system as ntopng metrics use exponential smoothing [8] implemented by nDPI which is used to detect when a metric deviates from its expected value. This has the advantage to avoid using static metric thresholds and generating false-positive alerts as individual hosts can have a very different behaviour that is enforced individually. However, in some checks (e.g. scan detection), we combine smoothing with upper/lower thresholds (set to extreme values that should never be crossed) as we want to avoid our system failing to trigger alerts for a misbehaving host that persist to misbehave (e.g. a scanner host which is continuously scanning other hosts). This is because with smoothing it is possible to

detect changes in behaviour but it does not check the absolute metric value that needs to be checked with other means (e.g. by using thresholds or comparing it against the same metric of other hosts to spot outliers).

This work has been implemented as open source and it has been validated extensively by the authors using traffic traces containing attacks and coming from popular sites (e.g. https://www.malware-traffic-analysis.net). However, we have realised validation limited to short (in time) traces containing malicious traffic would not be enough to guarantee that cyberscore works in reality. The following section covers our experiments and findings in these two different scenarios and reports lessons learnt. The ntopng tool implementing cyberscore has been deployed for over one year in the two networks used for experiments. Being our tools open-source, thousand of users contributed to its development and provided implementation feedback on various other networks to which we have no access and thus that are not reported in this paper. The limitation of this approach is that due to privacy concerns it is not possible to provide traffic traces to reproduce the results described below, but as the source code is available researchers could run similar experiments on their test networks.

III. Using Cyberscore In Practice

Two different networks have been selected for validating this work and evaluating the use of cyberscore: a leading service provider and a large country-wide private corporate network. The reason behind this choice lies in the fact that they have very different monitoring requirements:

- A service provider has limited control over the hosted services that are operated by its customers. In this case, the monitoring goal is not to tackle all the security issues, but to only identify and isolate attackers or infected systems hosted on the provider network, this to protect the network assets by remote attackers that gained control of some hosted systems. In this network, besides protection for DDoS attacks, there are almost no security policies in place as customers should freely use the Internet.
- In a corporate network, traffic is continuously enforced by security devices such as firewalls and IDS/IPS (Intrusion Detection/Prevention System), and Internet traffic is limited to selected computers to the allowed destinations. Theoretically, this should be a secure network, but the use of personal devices and external consultants that need to connect to the corporate network is a potential source of trouble. Furthermore, in companies with remote sites, it is important to accurately monitor traffic leaving/entering the core network as these sites are often administered by people with heterogeneous security skills that can propagate local weaknesses to the core network.

A. Internet Traffic Monitoring

Due to a large amount of traffic of the service provider under analysis, the data sample which has been used for the evaluation has been limited to 30 days (from mid-February to mid-March 2022) worth over 4 billion flow records stored by ntopng on a single-node ClickHouse database. We have mirrored Internet traffic at the two main Internet gateways connected with 10 Gbit upstream links and analyzed on a Ubuntu Linux system equipped with an Intel Xeon Silver 4116 CPU and 64 GB of RAM that also runs the database. Traffic is heterogeneous as some customers have servers they manage and thus with full Internet access. Others, use hosting services with limited services used, typically web access and SSH (Linux) or RDP (Windows) administration. The network has over 2'500 active hosts, some of which serve several domain names web pages for a total of over 100'000 hosted domains. Contrary to residential networks where most traffic is originated by clients, in this network most hosts are servers, meaning that they originate limited client traffic mostly due to updates and remote administration, and for this reason, egress is about 10 times more than ingress traffic. As this work is based on cyberscore, we only focus on flows for which a check has detected an issue which is about 25 % of the total. The table below shows the top 5 flow alerts.

TABLE II
Top Flow Alerts (Service Provider)

Alert Type	Flows (%)
Flow with errors (e.g. DNS error)	35.9%%
TLS Flows Not Carrying HTTPS	24.3%
Connections Over Insecure Protocol	9.5%
Service on a Non-Standard Port	8.6%
Suspicious DGA Domain	3.3%

1) Blacklists: In our experiments, hosts listed in blacklists are responsible for about 74 % of alerts. As they usually scan a network for victims, the system would be able to catch them with the scan check, avoiding blacklists at all. However as these blacklisted hosts are often probing hosts using techniques that exploit known or zero-day vulnerabilities, we believe blacklists are still useful to prevent blacklisted hosts to talk with other hosts as they might successfully exploit some systems before ntopng detects them. In our experiments, we have not reported false positives when using high-quality blacklists such as those from Emerging Threats and AlienVault, whereas blacklists that are not constantly updated, or a patchwork of various less reliable lists, should be avoided as sometimes they lead to false positives.

2) Attackers Detection: As stated before, cyberscore does not require prior training, but it works better when the monitoring system has some network knowledge. For this reason, ntopng introduces the concept of host pool, which is the ability to group homogeneous devices, by specifying the IP or MAC address. Contrary to IP subnets, host pool members do not need to have contiguous IP addresses which would be a major limitation for an Internet company with customers that are constantly added/removed or change their business nature over time. A service provider can have a pool for web hosting, VPSs (Virtual Private Servers), as well as a pool for network equipment (e.g. routers, storage systems, and VPN concentrators). As each pool has a different behaviour,

it is possible to define an expected traffic pattern for each pool. For instance, a remote host that wants to download or send emails from a host belonging to network equipment is probably a scanner, and this information can be used to feed the host cyberscore. This is because attackers usually do not have prior network knowledge and thus during their attack activities cannot differentiate their attack vectors based on the target host pool. In other words, this can be used as a simple honeypot to improve the host cyberscore.

3) Misbehaviour Detection: Host checks allow alerts to be triggered when suspicious host behaviour is detected. Remote scanners are detected by keeping a simple counter for <Protocol, Source IP, Destination IP, Destination Port> which is reset every minute. In normal traffic conditions, a remote host contacts a few hosts and ports of the service provider, thus a threshold set to a reasonably small value (e.g. 25) allows to reliably find attackers. Brute force attacks can be detected by keeping a counter of flows that a remote host has initiated towards the same destination host and port.

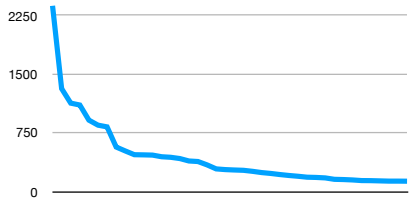

Fig. 1. Top One Minute Triplet Count Distribution

Figure 1 shows the top number of triplet <IP source, IP destination, destination port> flow count in one minute: on the ordinate, it is represented the number of flows with the same triplet, on the abscissa the top unique triplets. Besides a few limited exceptions such as DNS, SMTP, and HTTP/HTTPS that can generate a high number of flows, on all other ports/services we expect the triplet count to be low in value. This means that setting a relatively high threshold (e.g. 500) is enough to reliably spot misbehaving attacker hosts. This analysis enables scanners to core network services to be detected, as well as to quickly detect monitoring hosts (e.g. hosts performing nightly vulnerability scans) that instead should be listed in flow check exceptions as these traffic patterns are part of their daily duties.

4) Slow Scan Detection: Scanning is usually a quick activity that attackers want to carry on as fast as possible. However, there is a low percentage of scanners (<5%) that instead perform slow scans hoping not to be detected. In order to find them, ntopng takes into account the connection frequency, and the flow size that in normal traffic has a high bytes variance [9] compared to attack traffic. In our setup, we have verified that flows sent from an attacker towards various destination hosts on the same destination port (network scan) have a variance close to zero as they are basically probing attempts that can either fail (i.e. no response or reset in case of TCP connections) or have the same content meaning that the requests are made by a bot. The analysis of the flow content has been performed using data binning techniques implemented in nDPI, as explained in detail in [5].

5) DNS Traffic Analysis: DNS traffic is an important source of information for detecting possible misconfigurations or attacks, as well for understanding what security countermeasures have been used in the network. Indeed, DNS is no longer just used to resolve addresses but it is a key component of cybersecurity as many tools use it for querying reputation databases (e.g. for spam detection in mail servers). With this service provider, about 52 % of the DNS queries were issued for non-name resolution related tasks. These queries are useful to map services to hosts and to indirectly inspect Internet traffic that will likely be transferred in encrypted channels (e.g. email). The rest of the DNS queries are analyzed in terms of query type/error and queried domain name. DNS error responses can sometimes happen, but if the error ratio request/response is too high (15 % is already a high threshold) either there is a service misconfiguration or the host is probably performing some sort of scanning. Through a trigram-based analyser [10] implemented in nDPI, the DNS content check can detect queries that are likely to be DGA (Domain Generation Algorithms) [10]. All these checks increment the host cyberscore and can be used to passively create a DNS cache for helping the DPI engine when unable to classify traffic.

6) Service Behaviour Analysis: As previously discussed, individual flow analysis is complemented with host behavioural analysis. The idea is to create a model for every host of the service provider and trigger an alert whenever the system reports an unexpected behaviour. Earlier in this paper, we have discussed how exponential smoothing techniques have been used to trigger alerts whenever a relevant host metric (e.g DNS response Error/OK ratio) changes its behaviour and thus increase the cyberscore. In addition to this, we have implemented a service mapping facility based on nDPI that allows used/provided services to be mapped and that we have named *service map*. Practically, ntopng creates a graph of local (i.e. those that belong to the service provider) host services using passive leaning of the monitored traffic. The learning period, typically set to 24 hours, is used by the system to record interactions between hosts, and assigning them a default ‘pass‘ verdict. During or after learning, network administrators can modify the verdict for individual interactions, to mark some of them as ‘block‘ meaning that an alert will be generated. As soon as the learning period is over, communications not falling into the set of the allowed ones, trigger alerts that contribute to the cyberscore. In essence, the system records interactions for each observed local IP in a hash table for each triplet <Source IP, Destination IP, Application Protocol> a ‘pass‘ or ‘block‘ verdict; for non-local hosts, a special ‘any IP‘ value is used to avoid creating a huge hash for every remote-to-local or local-to-remote interaction. Using this technique we have an additional mechanism for detecting scans and, even more important, lateral movements [15], which is a technique used by attackers or compromised hosts to move into the

network for finding new victims. The main risk we have when monitoring an open network (i.e. mostly not firewall-protected, as in the case of customer hosts of the service provider) is that the high number of scans and attacks can hide signals coming from local compromised hosts. Using the service map we have been able to spot hosts that change their typical behaviour, as well as malicious applications running on hosts. For a service provider hosting thousands of domains, a typical threat is due to insecure websites that are not continuously maintained, and thus patched, as new vulnerabilities are discovered. In particular, many websites run outdated and insecure versions of CMSs (Content Management System) that through public CVEs (Common Vulnerabilities and Exposures) enable attackers to gain access to the remote system and run scripts for launching attacks or mining cryptocurrencies. The use of cyberscore has been successfully used to detect such systems as when these events happen:

- The service map reports new communication flows, sometimes using application protocols not used before (e.g. when a miner is installed in the remote host).
- Certain protocols such as DNS spikes in traffic compared to the recent past, and often there are cyberscore increases due to queries for suspicious DGAs or high DNS error response rate for non-existing domains (NXDOMAIN).
- Hosts have a typical role, either client or server. A web server that runs malicious code used to attack other hosts, has an increase in client activities both in terms of active flows and cyberscore. These changes in behaviour are detected both in terms of absolute values (i.e. a high cyberscore value) and also using alerts reported by checks computing exponential smoothing on the cyberscore value.

To efficiently implement many checks described above, we have used a few probabilistic algorithms we implemented in nDPI in order to implement an efficient system both in terms of performance and resource usage. In particular, we have used:

- HyperLogLog for cardinality estimation (e.g. for counting the number of different domain names contacted by a host).
- Data binning techniques for detecting similar timeseries (e.g. the host cyberscore timeseries) and clustering them [12]. This is a useful feature in case of misbehaving hosts, as it allows to easily detect similar hosts that might have been affected by the same issue.

B. Corporate Traffic Monitoring

As already discussed, corporate networks are usually protected by firewalls and expose none or just a few Internet services (e.g. corporate website). In most cases, contrary to service providers, firewalls are configured with a default deny policy, which means blocking all traffic that has not been expressly permitted. This applies especially to inbound traffic, considering that usually there is no service exposed, with the exception of VPNs and other services that provide connectivity to remote sites, home workers, and consultants. This is the case of the corporate network we have evaluated: it consists of about 2'000 active hosts, including the core network and remote sites connected through VPNs. Outbound and intra-LAN traffic is also limited to core services and enforced using:

- DNS servers, all devices inside the network are supposed to use the DNS server enforced by the network that is also used as honeypot and a first level of defence as it filters requests for suspicious and malware domains listed on blacklists.
- Web proxy server, capable of filtering HTTP(S) requests and dropping unwanted traffic.
- Mail servers, with anti-spam and anti-malware protection.

In this scenario, our monitoring goal is to identify anomalous network activities carried on by internal devices, rather than protect the network from remote attackers. This includes:

- Spotting new devices: discover when a new device is connected to the network and check if that device is allowed to do so.
- Detecting abnormal traffic: hosts generating high traffic volumes, compared to baselines or configured traffic thresholds.
- Identifying compromised machines: detect access to blacklisted hosts, unexpected activities such as scans, and HTTP requests that exchange binary applications or sensitive data (e.g. passwords).
- New services detection: hosts providing or using services that were not previously active (i.e. unexpected FTP traffic on a database server)
- Suspicious DNS traffic patterns: requests to DNS servers other than those enforced by the network.

In such a large corporate with many devices connected, it is important to monitor all devices joining the network, by means of network discovery and inventory tools. The above checks not only detect when a new device or service configuration (e.g. a new DNS server is started to be used) appears on the network, but are also used to detect hosts that silently scan the network. Most of the alerts detected on this network are about application on non-standard port and TLS issues. The first family of alerts highlights the presence of some protocol used on a non-standard port (e.g. HTTP traffic detected on TCP/22 instead of the default TCP/80). The second family includes TLS sessions with missing SNI (Service Name Indication) or obsolete TLS version that need to be investigated. The list of detected alerts also included the exchange of binary executables, something unexpected and very dangerous, is due to sub-optimal custom software developed to automate company tasks.

1) Encrypted Traffic Analysis: In corporate networks, it is not uncommon to use self-signed certificates to expose services. While this check should be disabled for known legit hosts, it should not be disabled for all hosts as it is important to track communications that can hide sensitive data behind encrypted channels. In particular for malware detection, it is also possible to enable checks that match the TLS JA3 fingerprint [13] against a database of known mali-

cious JA3s (e.g. https://ja3er.com) for increasing the detection of malware-based communications. We have also enabled a check for triggering alerts due to missing SNI or lack of ALPN (Application-Layer Protocol Negotiation) negotiation thus hiding the nature of the communication that is often non-HTTP related but used in VPNs or other encrypted communications, that can be used to exfiltrate data. As future work, for TLS/QUIC we plan to store in the service map the certificate fingerprint (when present) in order to detect unexpected changes in the server configuration.

2) Beaconing Detection: As shown earlier in this section, flow checks are good for individual flow analysis. Nevertheless they should be complemented with additional behavioural analysis in order to provide insights on network activities that are cross-flows such as beaconing detection. Beaconing [14] is usually used by malicious applications to connect with peers and are represented by periodic, low-volume communications that can be easily hidden in the overall traffic. Detecting such activities is crucial to identifying compromised hosts. These activities can be easily detected by keeping track of the <Source IP, Destination IP, Destination port, Layer 4 Protocol > quadruplets and monitoring their periodicity. This information, combined with the application protocol information provided by nDPI, allows us to identify malicious beaconing with a high level of confidence. For example, periodic communications using an unknown protocol or a potentially malicious protocol (e.g. IRC) is definitely suspicious and this information can be used to trigger an alert.

IV. Cyberscore Evaluation

Our experiments confirmed that cyberscore has enabled us to implement a numerical value that simplifies the detection of cyberthreats as it is simple to compare contrary to the list of individual signals used in checks. Its main limitation is that it should be based on reliable checks that have very few false positives (e.g. when analyzing DGA domains in DNS queries), as well as silence checks for selected communications (e.g. a monitoring host performing periodical active scans that can be detected as malicious activities) that would pollute the cyberscore value. Hence even if cyberscore does not require a training phase as with AI-based tools, it is necessary to spend some time at the first run to silence checks for selected hosts and communications whose traffic is acceptable according to the specified security policy.

The main advantage of cyberscore compared to AI-based tools, is that it can be enriched by easily developing new checks that are general in behaviour and that should not be confused with IDS rules that instead are very event-specific. Additionally, cyberscore does not require traffic annotation and it is able to create a lightweight traffic model for each host based on the service map rather than a single fat model as most AI tools do.

Conclusions

This paper described the concept of cyberscore, a numerical value that is used to identify relevant cybersecurity events including attacks, scans, and other security threats. The proposed method combines theoretical and practical approaches by providing a numerical indicator of whether the monitored traffic is likely to be malicious. It has been implemented in two open source tools, that are actively developed by a large community, and that are easily extensible by means of a mechanism called check. This paper reports experiments performed on two different network types that represent a significant sample of the different types of networks. The result of these experiments confirmed the feasibility and effectiveness of the cyberscore, and this work demonstrates that by combining efficient algorithms with statistical analysis it is possible to develop tools that are simple in design, resource-savvy and produce the same results as costly AI-based tools that are more complex and often require skilled support from the manufacturer for tuning and deployment.

Source Code

The source code of the tools described in this paper is available under the GNU GPL license at https://github.com/ntop/.

References

[1] Crowdstrike Inc., "2022 Global Threat Report" https://www.crowdstrike.com/global-threat-report/, 2022.

[2] European Commission, "The EU Cybersecurity Act," https://digital-strategy.ec.europa.eu/en/policies/cybersecurity-act, Regulation (EU) 2019/881, 2019.

[3] Blessing, G., et al. "The Emerging Threat of Ai-driven Cyber Attacks: A Review." Applied Artificial Intelligence (2022): 1-34.

[4] Deri, L., Martinelli, M., Cardigliano, A. (2014). "Realtime High-Speed Network Traffic Monitoring Using ntopng". In 28th large installation system administration conference (LISA14) (pp. 78-88).

[5] Deri, L., Fusco, F. (2021, July). "Using Deep Packet Inspection in CyberTraffic Analysis". In 2021 IEEE International Conference on Cyber Security and Resilience (CSR) (pp. 89-94). IEEE.

[6] Vähäkainu, P., Lehto, M. (2019, February). "Artificial intelligence in the cyber security environment". In ICCWS 2019 14th International Conference on Cyber Warfare and Security: ICCWS 2019 (p. 431). Academic Conferences and publishing limited.

[7] Imasheva, B., Azamat, N., Sidelkovskiy, A., Sidelkovskaya, A. (2019, July). The practice of moving to big data on the case of the nosql database, clickhouse. In World Congress on Global Optimization (pp. 820-828). Springer, Cham.

[8] Hyndman, R., Koehler, A. B., Ord, J. K., Snyder, R. D. (2008). Forecasting with exponential smoothing: the state space approach. Springer Science Business Media.

[9] Ahmed, M., Mahmood, A. N. (2015). Novel approach for network traffic pattern analysis using clustering-based collective anomaly detection. Annals of Data Science, 2(1), 111-130.

[10] Wang, W., Zang, T., Lan, Y. (2018). The Rapid Extraction of Suspicious Traffic from Passive DNS. In ICISSP (pp. 190-198).

[11] Sivaguru, R., Choudhary, C., Yu, B., Tymchenko, V., Nascimento, A., De Cock, M. (2018, December). An evaluation of DGA classifiers. In 2018 IEEE International Conference on Big Data (Big Data) (pp. 5058-5067). IEEE.

[12] Deri, L. (2021, December). A Gentle Introduction To Timeseries Similarity in nDPI (and ntopng), https://www.ntop.org/ndpi/a-gentle-introduction-to-timeseries-similarity-in-ndpi-and-ntopng/.

[13] Oh, C., Ha, J., Roh, H. (2021). A Survey on TLS-Encrypted Malware Network Traffic Analysis Applicable to Security Operations Centers. Applied Sciences, 12(1), 155.

[14] B. AsSadhan, J. M. F. Moura and D. Lapsley (2009). Periodic Behavior in Botnet Command and Control Channels Traffic. GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference.

[15] Bohara, A., Noureddine, M. A., Fawaz, A., Sanders, W. H. (2017, September). An unsupervised multi-detector approach for identifying malicious lateral movement. In 2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS) (pp. 224-233). IEEE.