2

Understanding Sources of Cybersecurity Data

Cyber threats often lead to loss of assets. This chapter discusses the multitude of datasets that can be harvested and used to track these losses and origins of the attack. This chapter is not about the data lost during cyberattacks but the data that organizations can scour from their networks to understand threats better so that they can potentially prevent or even predict future attacks.

2.1 End-to-End Opportunities for Data Collection

The information systems used to perform business functions have a well-defined process spanning over connected systems. In a typical client server scenario, as shown in Figure 2.1, a user connects to a system via an internet pipeline. The system has built-in application functionality important to run the business function. A return pipeline sends a response back to the user. The functionality of the system allows the delivery of the information commodity requested by the user.

As the example layout in Figure 2.1a shows, the logical view of the user requesting access to a business application can appear to be fairly straightforward. However, within this pipeline there could be several points through which the request and response pass, as shown in Figure 2.1b, leading to several opportunities in the end-to-end process for data collection to help understand when a cyber threat may occur in this process.

As we can see in Figure 2.1b, when the user requests a resource, it has to go through a complex networking pipeline. The user may have a firewall on their own system and the router through which they send out the request. This request can be filtered through the internet service provider, lookups can be performed in the domain name system (DNS) and the data can be routed through multiple paths of routers, which are linked through the routing table.

https://doi.org/10.1017/9781108231954.003 Published online by Cambridge University Press

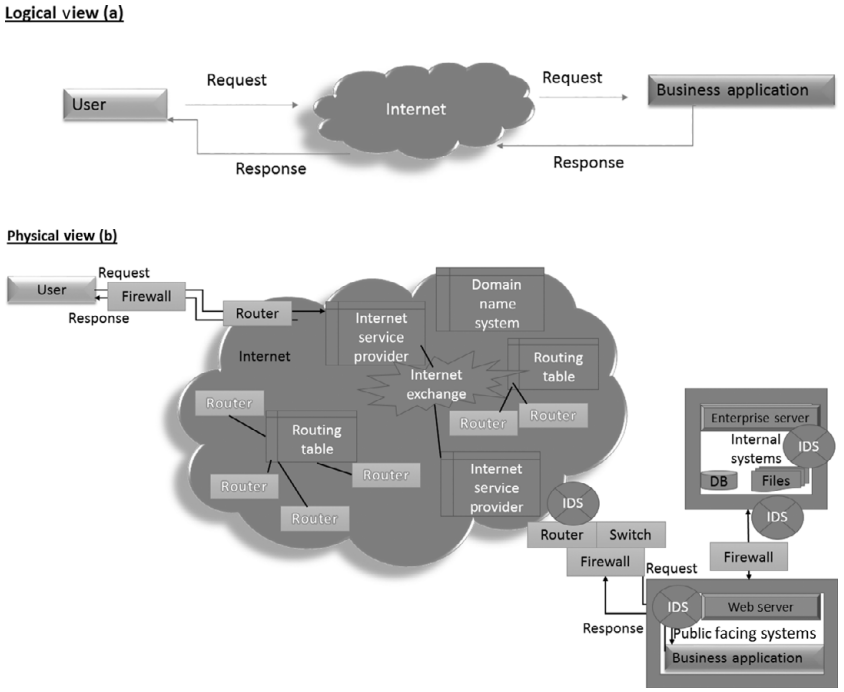

Figure 2.1 Logical and physical view of user request and response in a network-based environment.

The request on the other side may again have to pass through the routers and firewalls at multiple points in the system being accessed by the user. There may be multiple intrusion detection systems (IDS) posted throughout the systems to monitor the network flow for malicious activity. This is just one example scenario; different network layouts will result in different types of intermediate steps in this process of request and response, particularly based on the type of response, the type of network being used, the type of organization of business applications, the cloud infrastructure being used, to name a few factors. However, certain key components are always present that allow for multiple opportunities to glean and scour for data related to potential cyber threats.

There can be several opportunities to collect data to understand potential threats. Data collection can begin at a user access point, system functionality level, and commodity level (particularly if the data is being delivered). For example, at the user level, we can utilize data such as the following: (a) Who is the user? The psychology of the user, personality types, etc., can influence

https://doi.org/10.1017/9781108231954.003 Published online by Cambridge University Press

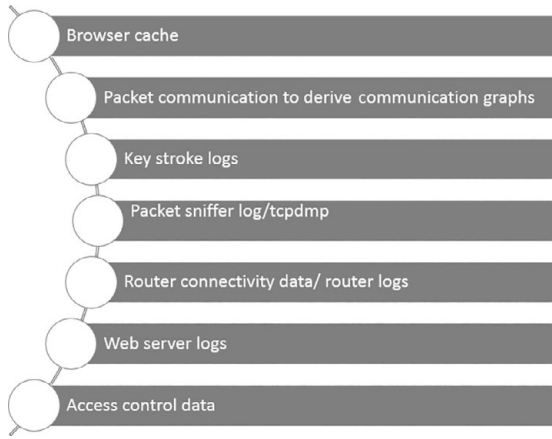

Figure 2.2 Common types of cybersecurity data.

whether a user will click on a link or give access to information to others. (b) What type of interface is being used by the user? Is there clear information about what is acceptable or not acceptable in the interface? (c) What type of access system is being used? Is there access control for users? (d) What data are available about the access pipeline, such as the type of network or cloud being used.

Several common types of datasets can be collected and evaluated, as shown in Figure 2.2, including various types of log data such as key stroke logs, web server logs, and intrusion detection logs, to name a few. We next discuss several types of such datasets.

2.2 Sources of Cybersecurity Data

Cybersecurity-related data collection will vary across the type of networks, including computer networks, sensor networks, or cyberphysical systems. The method and level of data collection will also vary based on the application domains for which the networks are being used and the important assets being protected. For example: (a) social media businesses, such Facebook, are primarily user data driven, where the revenue is based on providing access to user data and monitoring usage data; (b) e-commerce businesses, such as Amazon, are usage and product delivery based; (c) portals, such as Yahoo, are again user data driven but more heavily reliant on advertisements, which can target users based on what they see and use most often; (d) cyberphysical systems, such as systems for monitoring and managing power grids, are based on accurate functioning of physical systems and delivery of services to users over these physical infrastructural elements.

https://doi.org/10.1017/9781108231954.003 Published online by Cambridge University Press

In each of these types of systems, the underlying infrastructure has to be monitored to ensure accurate functioning and prevention, detection, and recovery from cyber threats. The level of monitoring and management of such data will vary with the level of prevention, detection, or recovery expected in the domain. Some domains have a high emphasis on prevention; others may have a high level of emphasis on detection or recovery. In all such cases, multiple types of datasets can be collected to provide intelligence on the cyber threats, and user behaviors can be evaluated to prevent future threats or even identify an insider propagating the threats.

In the following discussion for each dataset, we examine the following: (a) What is the data? (b) What is an example of its use in literature? (c) What type of detection can it be used for? In the chapters throughout this book, we will discuss how some of these datasets can be leveraged to discover anomalies identifying potential threats using data analytics methods.

2.2.1 Log Data

The nature of electronic communication and activities allows for several types of datasets to be logged. Some examples include the following: (1) intrusion detection system (IDS) logs including alarms raised by IDS; (2) key stroke logs; (3) router connectivity data/ router logs; (4) web server logs; and (5) firewall logs. This is not an exhaustive list but includes some of the major types of logs that can be collected.

2.2.1.1 Keystroke Logs

Keystroke logging or *key logging* is a mechanism to capture every key being pressed on a keyboard, but can also go beyond key presses to actions such as copying materials to the clipboard or other interactions with the user system. Key logging has been extensively studied for many applications, from writing to cognitive analysis to security threats. A survey on key logging (Heron 2007) outlines mechanisms, including hardware installation, kernel-level, system hook, and function-based methods, for key logging.

Key logging has also been studied for smart phones (Gupta et al. 2016, Cai and Chen 2011). A recent survey (Hussain et al. 2016) extensively outlines motion-based key logging and inference attacks that can result from smart phone key logging. This survey classifies key logging as in-band logging through the main channels of the keystrokes and out-of-band logging using side channels such as acoustics, power consumption, etc. Thus, key logging is not necessarily limited to keyboard-based data collection but can get quite sophisticated.

This type of data collection allows studying user behaviors but may also be used to maliciously detect user credentials, user preferences, or other

https://doi.org/10.1017/9781108231954.003 Published online by Cambridge University Press

sensitive information. Thus, it is also essential to understand the capabilities of key loggers to create any type of defense against threats utilizing key loggers.

2.2.1.2 Intrusion Detection System Logs

Intrusion detection system (IDS) log data (e.g., from Snort) provide data about alerts that are raised by matching any known signatures of malicious activities in the header and payload data. Generally, IDS will also provide an alert level of low, medium, or high. IDS logs analyze the packets based on malicious signatures and provide information on time stamp, service used, protocol, source, and destination. IDS can be placed at various points in a network, and multiple such datasets can be collected and correlated (Deokar and Hazarnis 2012). IDS logs are also commonly used for anomaly detection methods, which are utilized to detect threats beyond signature matching. Here anomalous packets indicate an unusual behavior with respect to the normal, where the normal can be discovered and predefined through various analytics methods.

IDS log data lends itself well to secondary analysis such as through data mining methods including association rule mining (such as Vaarandi and Podiņš 2010 and Quader et al. 2015), human behavior modeling (such as Quader and Janeja 2014 and Chen et al. 2014), and prediction of attacks, to name a few examples. Multiple IDS and other types of logs are also correlated to detect significant anomalies, which are not otherwise detectable (such as illustrated in Janeja et al. 2014 and Abad et al. 2003). Visualization of logs (such as in Koike and Ohno 2004) has been explored to facilitate the analysis of the logs by looking at the information selectively, slicing and dicing the data by certain features, such as by time or by event.

2.2.1.3 Router Connectivity and Log Data

The internet is a network of networks or subnetworks. The networks at each level are connected by routers. A router connects computer networks and forwarding data across computer networks. Each of these routers is connected for data transmission. This can range from a simple home router to corporate routers that connect to the internet backbone. A routing table stores information about the paths to take for forwarding and transmitting the data. The routing table stores the routes of all reachable destinations, including routers, from it. Various algorithms devise an efficient path through these connected routers (such as Sklower 1991 and Tsuchiya 1988).

A router provides not only route information but also all the raw IP addresses that pass through the router. These IP addresses can be mapped to

https://doi.org/10.1017/9781108231954.003 Published online by Cambridge University Press

identify possible malware activity when data are sent to suspicious geolocations in an unauthorized manner (Geocoding-Infosec 2013). However, care must be taken in using the IP addresses in isolation as they can be subject to IP spoofing, which hides the identity of the sender. Router data can also be utilized to study and possibly identify traffic hijacking (Kim Zetter Security 2013) and bogus routes by looking at historic route data stored in a knowledge base (Qiu et al. 2007).

2.2.1.4 Firewall Log Data

Firewalls act as a first line of defense that can stop certain types of traffic based on firewall security policies. In addition, these policies also have to be maintained to stay up to date with the changing landscape of the network usage. Essentially every access entry can be logged as it has to pass through the firewall. Some threats can be directly identified and blocked based on a clearly defined firewall policy or rule. For instance, if there is a clearly unauthorized access to an internal server, a well-configured firewall can block it to prevent access to the system. Major threat-related activities such as port scans, malware, and unauthorized access can easily be filtered through robust firewall rules. It essentially filters traffic based on the configuration of access to the systems protected by the firewall. Firewalls are typically designed to look at the header information in the data packets to match against prespecified rule sets. Firewalls can be host based or network based depending on whether they are deployed at an individual user's system or at a network interface.

Firewalls differ from IDS since they are generally limited to header information screening, whereas IDS can look at the payload data as well and block connections with malicious signatures. However, there has been a convergence in these functionalities in more recent times.

Firewall policy rules are one area where data mining may benefit by allowing the creation of a dynamic set of rules based on the traffic passing through the firewall. Analysis of policy rules and network traffic is used (Golnabi et al. 2006) to generate efficient rule sets based on the network traffic trends and potentially identify misconfigurations in the policy rules. This particular work uses association rule mining (ARM) and simple frequency counting of rules to generate firewall policy rules. In addition, it also identifies different types of policy anomalies, including blocking of legitimate traffic, allowing traffic to nonexisting services or redundant policy anomalies.

Similarly, Abedin et al. (2010) regenerates firewall policy rules and compares them with existing policies to discover anomalies.

https://doi.org/10.1017/9781108231954.003 Published online by Cambridge University Press

2.2.2 Raw Payload Data

Any data sent over the network are divided into multiple parts. Two key parts include (a) the header information, which stores data about source and destination among other things; and (b) the actual content being transmitted, referred to as the payload. There are several privacy concerns in accessing these payload data since these data are the actual content that is being sent, which may be under strict access control. Such payload data can be accessed only where legally allowed and users have provided permissions to access the data. Additionally, the data may be encrypted, so its usefulness as raw data to be mined is limited.

Payload data are accessible through packet sniffers such as Wireshark,[1] where the data dump of the traffic can be retrieved. Payload data can be massive even for a few minutes of data capture. Thus, it provides a strong motivation for using big data technologies to collect and mine such data where permissible. In addition, for web-based traffic the browser cache is another way to access the payload data from the client or end user's side.

Payload data have been shown (Wang and Stolfo 2004, Kim et al. 2014, Limmer and Dressler 2010) to be effective in identifying anomalous threats in network intrusion detection systems. For example, one recent study (Limmer and Dressler 2010) selectively analyzed parts of the payload, thus reducing the challenges in high-speed network intrusion detection systems. Parekh et al. (2006) utilize suspicious payload sharing in a privacy-preserving manner to identify threats across multiple sites.

Payload data can be used in multiple ways, such as to discover an individual user's behavior, the presence of malwares in the payloads, and other security threats that can be detected based on the actual content of the payload. One common use of payload data is to identify threats based on signatures of malware that may be present in the payloads. For example, if a virus is embedded in a packet and this virus has a known signature, then this can be captured by traditional intrusion detection system rules. One such open-source network intrusion detection system is Snort, which provides Snort rules (Snort 2020). Snort can also be used as a packet sniffer, like Wireshark, but can also be used as an IDS. Packets with malware embedded in them can be detected using multiple mechanisms such as simple keyword searches or complex regular expression matches and flagged. The traffic can be blocked or marked for further analysis, such as using Snort alarms or Wireshark coloring rules (Cheok 2014).

[1] www.wireshark.org.

https://doi.org/10.1017/9781108231954.003 Published online by Cambridge University Press

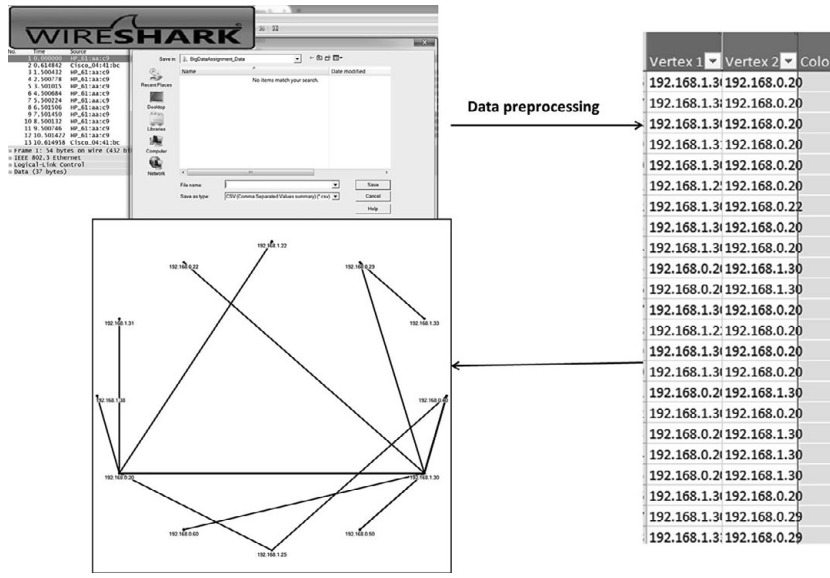

Figure 2.3 Example extraction of a communication graph from network traffic.

2.2.3 Network Topology Data

A computer network can be represented as a graph in terms of the structure of the network and in terms of the communication taking place over the network. Network traffic data dump can be used to generate the communication graph of all exchanges taking place over the network. As shown in Figure 2.3, for example, header data collected from a traffic dump file through Wireshark can be utilized to plot the communication between the source and destination IP addresses, which become the vertices, and the exchange between the two vertices forms the edge in the graph. In this example, NodeXL[2] is used to plot the graph data.

Once the communication data are in the graph form, graph metrics (for example, as discussed in Nicosia et al. 2013) can be computed, such as node-level metrics, including centrality, page rank, etc.; and network-level metrics, such as diameter, density, etc. In addition, based on the network properties, predictions can also be made about future network evolution. The example in Figure 2.4 illustrates one such task in a sample traffic data.

[2] www.smrfoundation.org/nodexl/.

https://doi.org/10.1017/9781108231954.003 Published online by Cambridge University Press

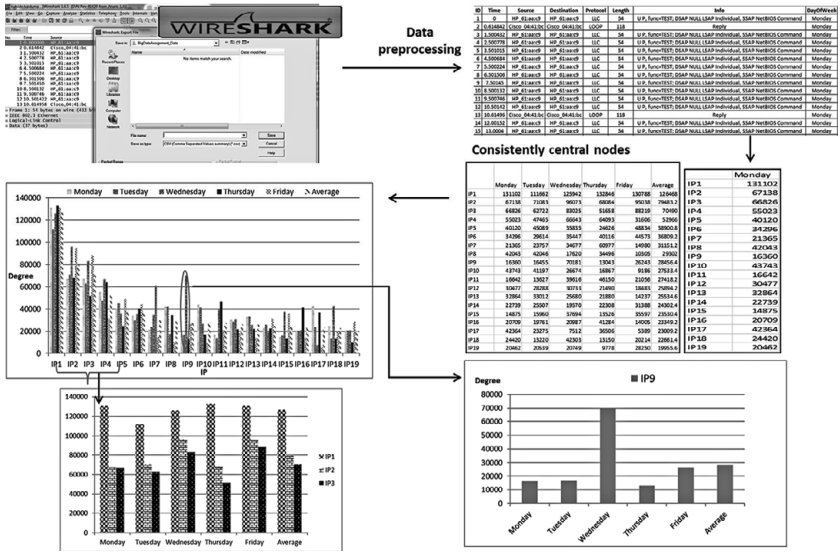

Figure 2.4 Exploratory analysis using Degree Centralities.

Data from network traffic is collected through packet sniffers such as Wireshark. To find communication behaviors of IP addresses along with anomalous fluctuations, exploratory analysis is performed on these data. The data from the network traffic need to be preprocessed, and this preprocessing will change with the task being performed. For instance, if in this example we wish to perform analysis by day of the week, the traffic data are sorted by the day of the week to get patterns by day, such as all Mondays or all Tuesdays. We can then compute the degree (i.e., number of edges incident on a vertex) of each node by day of the week. This can be performed for specific dates also; however, in this particular example we are interested to see behavior on certain days of the week by each of the IP addresses. We can sort the IP addresses by their degrees across days of the week, and the top ones appear to be consistently present in the traffic. Similarly, nodes with low degrees can also be identified. In such a scenario, it would be interesting to find a node, which is generally highly consistent as a high-degree node, to appear in the list of nodes with a lower degree, indicating a shift in the traffic pattern. Now let us consider the bar chart of the degrees for each IP address across each day of the week. We can observe that some IP addresses are consistently higher degree across all days of the week, which is further illustrated by the plot for IP1, IP2, and

https://doi.org/10.1017/9781108231954.003 Published online by Cambridge University Press

IP3 across all days of the week. We can also see that the degrees of IP9 and IP7 seem to be higher on some days but lower on other days. This is further clarified by the plot for IP9, which shows Wednesday as a day where IP9 has inconsistent behavior.

Thus, through such exploratory analysis it is not only possible to identify nodes that are inconsistent but also time points where the behavior is inconsistent. Alternatively, this method can also be used to identify highly connected nodes (such as nodes receiving higher than normal connections during a breach) or least connected nodes (perhaps nodes that are impacted by a breach and lose connectivity). This type of consistency and inconsistency can be identified at the node level and at the graph level as discussed in Namayanja and Janeja (2015 and 2017)

Another study (Massicotte et al. 2003) introduces a prototype network mapping framework that uses freely available network scanners (nmap, Xprobe) on built-in network protocols (ICMP, ARP, NetBIOS, DNS, SNMP, etc.) to create a real-time network topology mapping with the help of intelligence databases. It must be used in tandem with an intrusion detection system. Studies discussed earlier for graph metrics can be applied to such works as well after the topology is discovered.

2.2.4 User System Data

Figure 2.5 outlines several key features that can be extracted to monitor unusual activities at the individual system level. Example features include active process resident memory usage, which is available for all operating systems (OS) and allows for building a profile on the normal memory usage of a process over time. As an example, an abnormal spike in memory usage can be attributed to processing a large volume of data. This might be useful in detecting a potential insider threat, especially when integrated with other user behavioral data from sensors monitoring user stress levels or integrating with other log datasets. Similarly, CPU time utilization can be used for measuring system usage. Several OS-specific features, such as kernel modules and changes in registry values, are also identified in Figure 2.5. However, it is important to use multiple signatures over time from several of the features to eliminate the regular spikes of day-to-day operations. This is the key differentiator for a robust analysis where we do not simply rely on one or two features but multiple features and their stable signatures (as compared to historical data) to distinguish alerts. Tools such as OSQuery

https://doi.org/10.1017/9781108231954.003 Published online by Cambridge University Press

Feature name	OS specific
Active process name, active process filesystem path, active process ports and sockets, active process file access, active process resident memory usage, active process CPU time utilization, active process system calls, active process priority value, active process owner and group information, loaded peripherals drivers, key-store access patterns	All OS
Loaded kernel modules	Linux/Unix
Loaded kernel extensions	Mac OSX
Change in registry values	Windows
File system journaling (metadata) information	All major file systems (NTFS, ext4, HFS+)
Network routing tables	All major OS
Network firewall rules	All major firewall implementations
System-level sensors (current, voltage in different bus inside PC, CPU/GPU fan speed, etc.)	Almost all peripherals

Figure 2.5 OS-specific variables for CPU processing.

(OSQuery 2016) and Snare (SNARE 2016) can facilitate capture of these features.

Stephens and Maloof (2014) provide a very general framework for insider threat detection by gathering information from file read/write activities, printing, emailing, and search queries, then building a probabilistic Bayesian belief network from the sensor and context data, such as a user behavior profile from past actions. Van Meigham (2016) focuses on macOS malware detection using a kernel module to intercept system calls and generating a heat map analysis on the results.

2.2.5 Other Datasets

In addition to the datasets discussed, there are additional datasets that can be utilized to leverage knowledge about cyberattacks.

Access control data: These data can help better understand usage of the assets that need to be protected. Role mining (Vaidya et al. 2007, Mitra et al. 2016) from access control data can help shape and create better and more robust roles.

Eye tracker data: A user's behavior can be judged by the interactions of the user with the system being used. One such mode of input is the screen. Data collected from the user's eye gaze, captured through an eye tracker, can help analyze the user's level of engagement with a system and user preferences or positioning important items on the screen (such as those discussed in

https://doi.org/10.1017/9781108231954.003 Published online by Cambridge University Press

Darwish and Bataineh 2012) to evaluate browser security indicators. The data collected through the eye tracker can be mined for patterns such as associations between security cue locations on the screen and number of views or clicks. Clustering can be performed on eye gaze data to identify presence or absence of clusters around security cues. Associations can be analyzed between user's perception of security, backgrounds, and demographics to different zones of eye gaze foci in a stratified manner. If users perceive disclosing important information through emails as a low-risk activity, they are less likely to see the security cues. Similarly, if they see the security cues, their perceived risk of responding will be high. Studies have hypothesized that user education can change user's perception of security and help them to better see these security cues, increasing the likelihood of threat detection or identifying threats through visual cues such as in the case of phishing.

Vulnerability data: Software vulnerability is a defect in the system (such as a software bug) that allows an attacker to exploit the system and potentially pose a security threat. Vulnerabilities can be investigated, and trends can be discovered in various operating systems to determine levels of strength or defense against cyberattacks (Frei et al. 2006). Using the National Vulnerability Database from the National Institute of Standards and Technology (NIST) (NIST 2017), trends can be analyzed for several years and across major releases for operating systems to reinforce knowledge of choices for critical infrastructural or network projects.

NVD is built on the concept of Common Vulnerabilities and Exposures (CVE),[3] which is a dictionary of publicly known vulnerabilities and exposures. CVEs allow the standardization of vulnerabilities across products around the world. NVD scores every vulnerability using the Common Vulnerability Scoring System (CVSS).[4] CVSS is comprised of several submetrics, including (a) base, (b) temporal, and (c) environmental metrics. Each of these metrics quantifies some type of feature of a vulnerability. For example, base metrics capture characteristics of a vulnerability constant across time and user environments, such as complexity, privilege required, etc. The environmental metrics, on the other hand, are the modified base metrics reevaluated based on organization infrastructure. NVD allows searches based on subcomponents of these metrics and also based on the basic security policies of confidentiality, integrity, and availability. These searches can provide data for analysis to identify trends and behaviors of vulnerabilities across operating systems or other software for different types of industries.

[3] http://cve.mitre.org/. [4] https://nvd.nist.gov/cvss.cfm; www.first.org/cvss.

https://doi.org/10.1017/9781108231954.003 Published online by Cambridge University Press

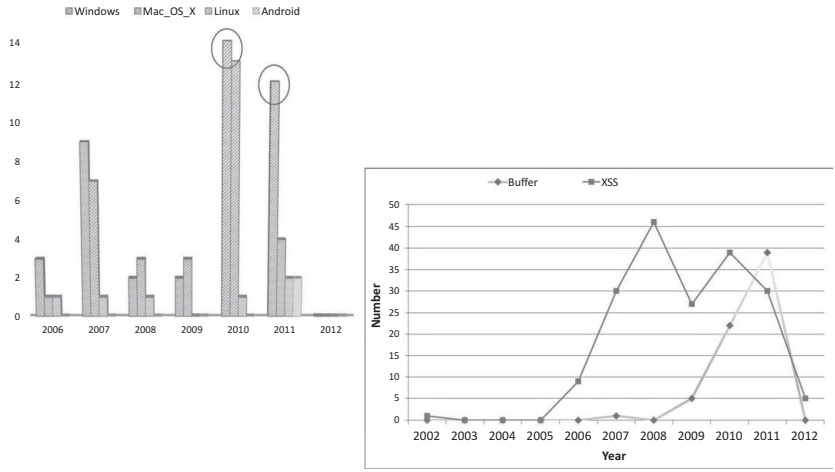

Figure 2.6 Comparison of vulnerabilities over operating systems.

Let us consider cross-site scripting vulnerability.[5] When data regarding the number of vulnerabilities are pulled from NVD across 2006 to 2012, we can see the trends of operating systems that are most impacted by this vulnerability, as shown in Figure 2.6. In addition, we can also compare the occurrences of different types of vulnerabilities such as cross-site scripting and buffer overflow. While this is a straightforward plotting of number of vulnerabilities across years, it provides insights into the robustness of operating systems for different types of vulnerabilities and across different CVSS metrics. Such analyses can be an important feed into decision making before choices for adopting software are made from a security point of view in organizational applications.

2.3 Integrated Use of Multiple Datasets

Let us consider a scenario where multiple datasets can be utilized to study potential cyberattacks. Cyberattacks are rare compared to the day-to-day traffic in a computer network; therefore, they appear in datasets as anomalies. Anomalies are essentially data points or patterns that are unusual with respect to the normal. It is clear that there needs to be a frame of reference that is

[5] https://tools.cisco.com/security/center/viewAlert.x?alertId=35601, Adobe Flash Player Cross-Site Scripting Vulnerability.

https://doi.org/10.1017/9781108231954.003 Published online by Cambridge University Press

"normal" compared to which something is deemed an "anomaly." A single dataset such as any of the ones discussed so far can be used for anomaly detection, but it is important to note that if multiple datasets result in similar types of anomalies, then the credibility of labeling an anomaly is higher.

One such integrated evaluation would be to discover anomalies in network traffic data with a temporal, spatial, and human behavioral perspective. Studying how network traffic changes over time, which locations are the sources, where is it headed, and how are people generating this traffic – all these aspects become very critical in distinguishing the normal from the abnormal in the domain of cybersecurity. This requires shifting gears to view cybersecurity as a holistic people problem rather than a hardened defense problem. By utilizing some of the datasets discussed in this chapter, we can answer the following important questions in studying these aspects:

Firstly, computer networks evolve over time, and communication patterns change over time. Can we identify these key changes that are deviant from the normal changes in a communication pattern and associate them with anomalies in the network traffic?

Secondly, as attacks may have a spatial pattern, sources and destinations in certain geolocations can be more important for monitoring and preventing an attack. Therefore, can key geolocations that are sources of attacks, or key geolocations that are destinations of attacks, be identified? Moreover, can IP spoofing be mitigated by looking at multiple data sources to supplement the knowledge of a geospatial traffic pattern?

Thirdly, any type of an attack has common underpinnings of how it is carried out; this has not changed from physical security breaches to computer security breaches. Can this knowledge be leveraged to identify behavioral models of anomalies where we can see patterns of misuse?

Recent work highlights some of these questions in discovering anomalies utilizing network data to study human behavioral models such as Chen et al.) 2014) and Quader and Janeja (2014). These will be discussed further in Chapter 10.

2.4 Summary of Sources of Cybersecurity Data

Through this chapter, multiple types of sources of cybersecurity data have been discussed. Table 2.1 summarizes these data under the following: (a) data source, (b) literature study examples, and (c) type of detection it can be used for.

https://doi.org/10.1017/9781108231954.003 Published online by Cambridge University Press

Table 2.1 *Summary of sources of cybersecurity data*

Source of cybersecurity data	Literature study examples	Type of detection it can be used for
Keystroke logging	Heron 2007, Cai and Hao 2011, Gupta et al. 2016, Hussain et al. 2016	User behavior, malicious use to detect user credentials
IDS log data	Abad et al. 2003, Koike and Ohno 2004, Vaarandi and Podiņš 2010, Deokar and Hazarnis 2012, Chen et al. 2014, Janeja et al. 2014, Quader and Janeja 2014, 2015	Association rule mining, human behavior modeling, log visualization, temporal analysis, anomaly detection
Router connectivity and log data	Tsuchiya 1988, Sklower 1991, Qiu 2007, Geocoding Infosec 2013, Kim Zetter Security 2013	Suspicious rerouting, traffic hijacking, bogus routes
Firewall log data	Golnabi et al. 2006, Abedin et al. 2010	Generate efficient rule sets, anomaly detection in policy rules
Raw payload data	Wang and Stolfo 2004, Parekh et al. 2006, Limmer and Dressler 2010, Kim et al. 2014, Roy 2014	Malware detection, embedded malware, user behavior
Network topology	Massicotte et al. 2003, Nicosia 2013, Namayanja and Janeja 2015, 2017	Consistent and inconsistent nodes, time points corresponding to anomalous activity
User system data	Stephens and Maloof 2014, Van Meigham 2016	User profiles, user behavior data, insider threats
Access control data	Vaidya et al. 2007, Mitra et al. 2016	Generate efficient access control roles
Eye tracker data	Darwish and Bataineh 2012	Browser security indicators, security cues, user behavior
Vulnerability data	Frei et al. 2006	Vulnerability trend discovery

https://doi.org/10.1017/9781108231954.003 Published online by Cambridge University Press