Reducing the Attack Surface for Private Data

George O. M. Yee
Computer Research Lab, Aptusinnova Inc., Ottawa, Canada
Dept. of Systems and Computer Engineering, Carleton University, Ottawa, Canada
e-mail: george@aptusinnova.com, gmyee@sce.carleton.ca

***Abstract*—Breaches of private data have been occurring at an alarming rate, to the embarrassment and expense of companies that hold the data. It would appear that in each breach, the attack surface for the data has been sufficiently large to attract attackers. Reducing this attack surface is a way to lessen the likelihood of breaches. This paper presents methods for reducing the attack surface of private data held in the online computer systems of organizations. The methods are applied to a software system's architecture early in the design process, as an approach for designing-in security. This work defines the attack surface for the data, and then uses this definition to obtain a formula for calculating the attack surface. The definition further leads to identifying methods that can be used to reduce the attack surface. Reducing the attack surface may not prevent breaches, but it will make them less likely to occur.**

Keywords-privacy; private data; breaches; attack surface identification; attack surface reduction.

I. INTRODUCTION

Breaches of private data held by companies and other types of organizations have been occurring at an alarming rate. Consider the following sampling of recent breaches [1]:

- August 21 – September 5, 2018: British Airways, 380,000 customers affected; card payment information stolen; the airline's website and app were hacked.
- January 1, 2016 – December 22, 2017: Orbitz, 880,000 customers affected; payment card information and personal data (billing addresses, phone numbers, emails) stolen; the company's website was hacked.
- May 1, 2015 – July 4, 2018: SingHealth, 1.5 million users affected; names and addresses in the Singapore government's health database, and some histories of dispensed medicine were stolen; also, the prime minister of Singapore was specifically targeted; hackers orchestrated a deliberate, well-planned attack.
- August 20, 2018: T-Mobile, about 2 million users affected; a group of hackers accessed T-Mobile servers through an application programming interface and stole encrypted passwords and personal data, including account numbers, billing information, and email addresses.

Apparently, the attack surface for the data that was breached, or the number of ways that the data could be accessed and stolen, was sufficiently large and attractive to the attackers.

Given the rate of recent data breaches, it is clear that more needs to be done to reduce the probability of a data breach occurring. The objective of this work is to derive methods for reducing the attack surface of private data held in online (i.e., connected to the Internet) computer systems of organizations. The methods are obtained from consideration of the definition of the attack surface, which in turn is based on how an attack happens. This definition also leads to a straightforward formula for calculating the size of the attack surface, which can be used to verify that use of the methods does indeed reduce the attack surface. The methods focus on reducing the attack surface by altering the system architecture, rather than the deployment of add-on security appliances, such as firewalls and intrusion detection systems. The methods are meant to be applied at the early stages of design within a software development cycle, as part of the Design for Security toolset.

This paper is organized as follows. Section II explains private data, attacks, attack surface, and how to calculate the size of the attack surface. Section III derives methods for reducing the attack surface based on its definition. Section IV illustrates the methods using an application example. Section V describes related work, and Section VI presents conclusions and future work.

II. PRIVATE DATA, ATTACKS, AND ATTACK SURFACE

A. Private Data, Attacks, and Attack Surface

Private Data (PD), also known as personal data, is data about an individual, can identify that individual, and is owned by that individual [2]. For example, an individual's driver license number, passport number, or credit card number can each be used to identify the individual and are therefore considered as private data. The individual's privacy then refers to his/her ability to control the collection (what personal data and collected by which party), purpose of collection, retention, and disclosure of that data, as stated in the individual's privacy preferences [2].

 ISBN: 978-1-61208-746-7

DEFINITION 1: An *attack* is any action carried out against an organization's computer system that, if successful, results in the system being compromised.

This work focuses on attacks that compromise the PD held in the online systems of organizations. The attacker who launches an attack may be internal (inside attacker) or external (outside attacker) to the organization. This work applies to both types of attackers. An internal attacker usually has easier access to the targets of his/her attack and he/she may hide his/her attacks in the guise of normal duty.

Salter et al. [3] give an interesting insight into what enables a successful attack: "Any successful attack has three steps: One, diagnose the system to identify some attack. Two, gain the necessary access. And three, execute the attack. To protect a system, only one of these three steps needs to be blocked." Thus, an attack surface must contain a target that the attacker deems worthy of attack (suit his/her purpose for the attack) and that target must be accessible to the attacker. For this work, the target that is potentially worthy of attack is the PD that is accessible to attackers. In a computer system, this PD is either moving (travelling from one location to another), at rest (stored), or being used (by some process). This leads to the following definition of attack surface:

DEFINITION 2: The *attack surface* for private data contained in an online computer system is the set of all locations in the system that contain attacker accessible PD in the clear, where the PD is moving, at rest, or being processed.

Figure 1 shows an example attack surface.

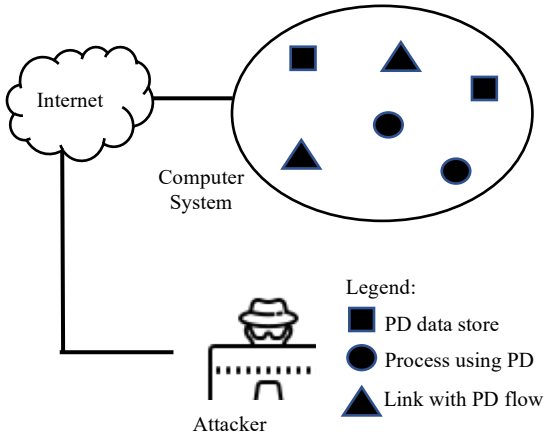

Figure 1. Example attack surface consisting of the set of all 6 attacker accessible locations in the system that contain PD in the clear.

In Definition 2, "attacker accessible PD" means that the attacker is able to exfiltrate the PD using some agent of attack, such as malware against stored PD and PD being processed, or a man-in-the-middle attack against a link containing moving PD.

An alternative definition of attack surface for PD contained in a computer system is the set of ways the attacker has to exfiltrate the PD. However, given the complexity of computer systems and the fact that the tools available to the attacker to use in his/her attacks are unknown to us, it is next to impossible to determine this set. On the other hand, locations that contain attacker accessible PD are easier to identify. Since an exfiltration must be from a location that contains PD, the set of such exfiltrations depends on the set of such locations. The larger the set of locations, the larger the set of exfiltrations. The smaller the set of locations, the smaller the set of exfiltrations. Therefore, Definition 2 in a sense includes this alternative definition, but in addition, is more easily applied.

As mentioned above, in the first step of a successful attack, the attacker diagnoses the system to identify the attack [3]. A smaller attack surface will make this step more difficult for the attacker. Therefore, a smaller attack surface corresponds to higher security, which is why we wish to reduce the attack surface. Definition 2 also gives rise to this conclusion: a smaller attack surface means a smaller number of locations that contain PD, which in turn means fewer opportunities for exfiltration of the PD, or in other words, higher security.

Definition 2 is consistent with the intuitive understanding of an attack surface, which is "the set of ways in which an adversary can enter the system and potentially cause damage" [4]. Each "way" corresponds to a location in Definition 2 that in turn corresponds to methods for exfiltrating PD from the location.

B. Calculating the Size of the Attack Surface

It would be useful to have a numerical value for the size of the attack surface, since then we could a) compare attack surfaces at different stages of development to see if the system's security is getting better or worse, b) compare attack surfaces of different systems when choosing a system for purchase, and c) easily see if actions taken to reduce the attack surface have indeed reduced it.

As mentioned above, private data held in a computer system can be in the following three states: moving, at rest, or being processed. These states correspond respectively, within a computer system, to PD that is moving along a link, PD that is stored in a data store, and PD that is being processed. Thus, the locations in Definition 2 refer to links, datastores, and processes that contain attacker accessible PD. Definition 2 then leads naturally to the following formula for calculating the size of the attack surface for private data.

Let N be the size of the attack surface for PD. Let m, n, and k be the number of links, data stores, and processes, respectively, that contain attacker accessible PD in the clear. Then

$$N = m + n + k \quad (1)$$

Equation (1) says that the size N of the attack surface is found by adding up the number of attacker accessible locations in the system that contain PD, namely: the number m of links, the number n of data stores, and the number k of processes, all of which contain attacker accessible PD. This equation follows directly from Definition 2, by simply replacing "attack surface" with "size of the attack surface" and

 ISBN: 978-1-61208-746-7

"set" with "size of the set" in that definition. Applying this equation to Figure 1 gives an attack surface of size N = m + n + k = 2 + 2 + 2 = 6.

III. REDUCING THE ATTACK SURFACE

A. Methods for Redusing the Attack Surface

Equation (1) implies that the attack surface will decrease if and only if any or all of the quantities m, n, or k decrease. Therefore, the attack surface may be reduced by the following methods, where each method decreases m, n, or k:

a) Make a PD location useless to the attacker.
b) Combine two or more PD locations into a single PD location.
c) Deny the attacker access to a PD location.
d) Remove a PD location from the system.

The following explains these methods in greater detail and describes how they may be carried out.

a) Make a *PD Location Useless to the Attacker*

As mentioned above, in the first step of a successful attack, the attacker diagnoses the system to identify an attack, or in our case, the PD target for the attack. In this diagnosis, it is reasonable to assume that the attacker will ignore any target that he/she finds useless for his/her purposes. Such targets may be removed from the attack surface. Some ways to make a PD target useless to an attacker are:

- Obfuscate (e.g., encrypt) the PD at the location. The attacker will not want to exfiltrate PD that cannot be read. The computer system will need to able to de-obfuscate the data securely for its own purposes.
- Anonymize the PD at the location. Again, the attacker will not want PD that cannot be linked to individuals, since it is this linking that adds value to the data, e.g., for advertising purposes. The computer system will need to be able to de-anonymize the data securely for its own purposes.

To illustrate, obfuscating one data store and one process in Figure 1 results in Figure 2, where the obfuscated data store and the obfuscated process have been removed from the attack surface. It can be seen that the attack surface in Figure 2 is reduced (size 4) relative to the attack surface of Figure 1 (size 6).

b) Combine Two or More PD Locations into a Single PD Location

This method will decrease the number of PD locations and reduce the size of the attack surface per (1). Additional data links may need to be implemented in the system's design to carry PD that was previously carried by links to/from the locations that were combined. In addition, changes to the software logic may be needed for data stores or processes that were combined to accomplish reading or storing the data in the combined location (for combined data stores), or new processing of data in the combined location. (for combined processes). To illustrate, combining two data stores into one data store and combining two processes into one process in Figure 1 also results in the reduced attack surface shown in Figure 2.

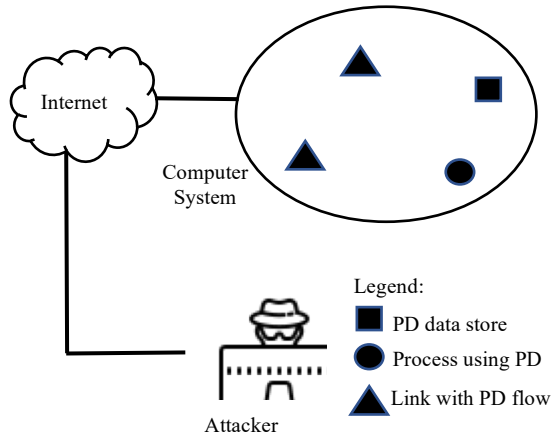

Figure 2. Resulting reduced attack surface of size 4 after obfuscating or combining locations in Figure 1.

c) Deny the Attacker Access to a PD Location

It may be possible to have some PD locations offline, thus denying the attacker access to these locations. For example, this may be possible for certain self-contained processing, such as analytics, that can be done using PD that is offline. In this case, all data stores, processes, and data links involved solely in such offline processing may be removed from the attack surface of the system to which these locations originally belonged, and re-constituted into an offline system. It may be necessary to update the offline PD data stores periodically using data from the system that is online. This update will need to be done in a secure fashion, perhaps by transferring the data manually using disks, after making sure that no malware can infect the offline system via this transfer. Although the destination locations are offline, it may still be possible for transferred malware to exfiltrate the offline data, e.g., hiding the data in the disks that are used for transfer and then transmitting the data once the disks are on the online part of the system.

d) Remove a PD Location from the System

Another way to reduce the attack surface is to remove a PD location from the system by deciding that the PD in the location is no longer required. For example, a company that stores the credit card information of its customers for their convenience may decide to stop storing this information, and instead, ask the customer for their credit card information every time the customer goes through checkout. This is in general a good decision, to avoid storing PD that may get compromised, at the cost of a little inconvenience. In this case, the associated credit card PD datastore would no longer be needed, and would be removed from the attack surface. Another example is the removal of a process that periodically sends customers the status of their order. The process uses PD consisting of the customer's name and email address to send the status. Suppose that this process is no longer necessary because the customer can now use a new Web interface to

 ISBN: 978-1-61208-746-7

check order status. Removal of this process from the system removes it from the attack surface. Interestingly, removal of a PD location can also result in removing other PD locations that are connected to the location that is removed. For example, the removal of a PD data store or a process that uses PD can result in also removing connected PD locations, such as the links that carry PD, or a PD data store that the removed process was exclusively using. Thus, removing a PD location not only removes that location from the attack surface but can also lead to removing other PD locations further reducing the attack surface.

B. Applying the Methods

Since the above methods operate on attacker accessible PD locations, it is recommended that they be applied in the second phase of two phases, where the attacker accessible PD locations are identified in the first phase. These phases are carried out on an architectural representation of the online system, such as a Data Flow Diagram (DFD) [5] (see the application example in Section IV). The phases are as follows.

- Phase 1: Identify PD locations by tracing the flow of private data in the online computer system, looking for where PD enters the system, where PD flows (links), where it is stored (data stores), and where it is used (processes). Identifying the PD locations by tracing the flow of PD in the system implies that there are paths to the PD that an attacker can use to exfiltrate the PD. We therefore conclude that all PD locations found in this manner in an online system are attacker accessible PD locations. Given the ingenuity of attackers (the exfiltration could even be aided by an insider of the organization that owns the computer system, through social engineering), this conclusion is valid.
- Phase 2: Apply the above methods to the attacker accessible PD locations found in Phase 1, where possible, while considering the potential negative effects on the following aspects of the system:
  - Performance
  - Reliability and dependability
  - Ease of maintenance
  - Implementation cost

  For example, encryption or anonymization incurs extra overhead, combining data stores may introduce a performance bottleneck since the newly combined data store will now need to additionally support data accesses that were originally shared among the data stores that were combined. Combining PD locations in general may reduce modularity and lead to extra effort needed to maintain the system. A general guiding rule is to look for opportunities to apply the methods where the potential negative effects mentioned above are minimal. It may be more efficient to consider method a) last, since the other methods can add/delete links that are candidates for method a).

Carrying out the above phases clearly requires knowledge of the computer system in terms of identifying the PD locations. Some basic knowledge of security would also be advantageous. These skills should be found within the software development team responsible for developing the system, perhaps with a little security training if needed.

IV. Application Example

This section illustrates how to apply the methods for reducing the attack surface for private data using an example computer system for an online seller of merchandise (e.g., Amazon.com). Suppose this system is at the beginning stages of development and that the development team has produced a DFD showing how both private and non-private data will flow, be stored, and used in the system. This DFD is shown in Figure 3.

The system in Figure 3 allows the customer to enter his/her "name", "address", "email", "item selected" for purchase, and "credit card info" for payment. These comprise the PD for this example. Five processes cooperate to provide the functionality for the system. One datastore stores the customers' private data; another datastore contains inventory data, i.e., what items are in stock. The system is an online system since it is for an online seller. The PD locations will be found by tracing the flow of PD in the system (described below). Thus, all PD locations in the system are attacker accessible PD locations, as noted above in the description of Phase 1.

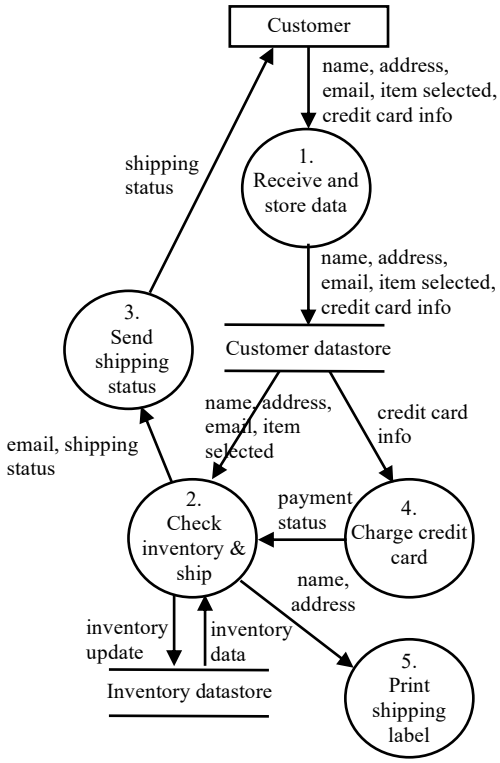

Figure 3. DFD for online seller system, showing how data flows, are stored, and used.

Applying Phase 1 in Section III B, we trace the flow of private data from the point where the data enters the system at

process 1. From there, the PD passes through process 1 and is stored in the customer datastore. After this datastore, the PD is split up with the "credit card info" going to process 4 to be used, and the "name", "address", "email", and "item selected" going to process 2, where the "item selected" datum is used, and "name" and "address" are passed to process 5 to print the shipping label, whereas "email" is passed to process 3 to send the customer the shipping status. Thus, we can identify the PD locations as links, datastores, and processes through which the PD passes, is stored, and used. These attacker accessible PD locations are shown in Table I.

TABLE I. ATTACKER ACCESSIBLE PD LOCATIONS IN FIGURE 3

	Links	Datastores	Processes
1	link into process 1	customer datastore	process 1
2	link out of process 1		process 2
3	link from customer datastore to process 2		process 3
4	link from customer datastore to process 4		process 4
5	link into process 5		process 5
6	link into process 3		

Table I shows that there are 6 attacker accessible PD link locations, 1 attacker accessible PD datastore, and 5 attacker accessible PD processes. For Figure 3, prior to the application of the above methods, (1) gives the size N of the attack surface for private data as $N = m + n + k = 6 + 1 + 5 = 12$.

Applying Phase 2 in Section III B, we first use the above methods on the attacker accessible locations in Table I, as follows:

- Using method b), combine process 3 with process 2; this was seen to have negligible impact on performance and an acceptable reduction in modularity.
- Using method b), combine process 5 with process 2; this was also seen to have negligible impact on performance and an acceptable reduction in modularity.
- Using method d), remove the customer datastore from the system; it was decided that storing customer PD was not needed (customer purchase history can be stored securely on the customer's device by the seller's app and later retrieved by the seller's website).

These changes result in the DFD shown in Figure 4. Table II gives the attacker accessible PD locations corresponding to Figure 4.

Table II shows that there are 3 attacker accessible PD link locations and 3 attacker accessible PD processes. For Figure 4, (1) gives the size N of the attack surface for private data as $N = m + n + k = 3 + 0 + 3 = 6$. Thus, the application of methods b) and d) have reduced the attack surface from 12 to 6.

We can further reduce the attack surface as follows:

- Using method a), obfuscate (encrypt) the links in Table II; the impact on performance due to the extra over head is deemed acceptable.
- Using method a), obfuscate (encrypt) the PD in the processes shown in Table II; here, the impact on performance and the cost involved for extra code to handle encryption/decryption were considered unacceptable, and this reduction step was not done.

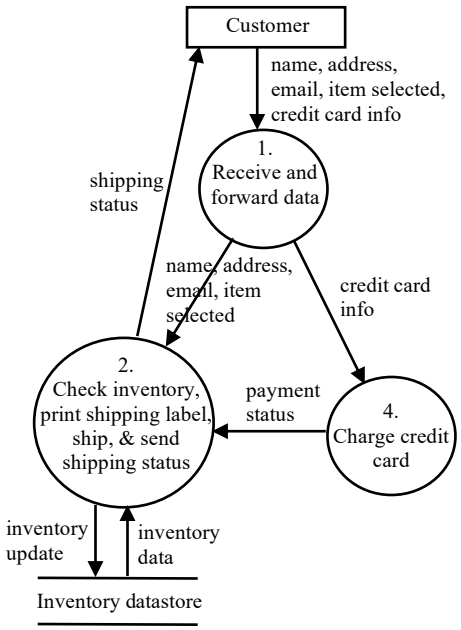

Figure 4. DFD for online seller system after combining processes and removing the customer data store.

TABLE II. ATTACKER ACCESSIBLE PD LOCATIONS IN FIGURE 4

	Links	Datastores	Processes
1	link into process 1		process 1
2	link from process 1 to process 2		process 2
3	link from process 1 to process 4		process 4

Table III shows the remaining attacker accessible PD locations after applying method a) to the links in Table II. The new attack surface is of size $N = m + n + k = 0 + 0 + 3 = 3$. The application of the methods in Section III has improved the security of private data in the system by reducing the size of the attack surface from 12 to 3.

Comparing Figure 4 to Figure 3, reducing the attack surface requires the following architectural changes to the system: a) reducing the number of processes from 5 to 3 by eliminating processes 3 and 5, b) changing the functionality of processes 1 and 2, and c) eliminating the customer database. As noted above, the implications of these changes were accepted by the development team.

TABLE III. REMAINING ATTACKER ACCESSIBLE PD LOCATIONS IN FIGURE 4 AFTER OBFUSCATING THE LINKS IN TABLE II

	Links	Datastores	Processes
1			process 1
2			process 2
3			process 4

 ISBN: 978-1-61208-746-7

The size of the attack surface obtained by applying the above methods depends on which methods were applied and the order in which they were applied. In particular, it may depend on the available opportunities for applying method b). For example, by using only method a) (obfuscation) on the locations in Table I and assuming that it is not advisable to apply method a) to the processes due to unacceptable impacts on performance and costs, we obtain an attack surface of size 5 (for the remaining 5 processes since the obfuscated links and datastore would have been removed from the attack surface), which is larger than the attack surface of size 3 obtained above by opportunistically first applying method b). This is the rationale for the comment made in the description of Phase 2 above, that it may be more efficient to consider applying method a) last.

V. Related Work

Most closely related to this work is this author's previous work on reducing the attack surface [6]. However, this previous work differs from the current work in at least the following ways: a) the previous work deals with sensitive data (including private data) whereas the current work focuses on private data, b) the previous work proposes a graphical model with which to identify the attack surface whereas the current work does not require any such model, c) the previous work reduces the attack surface by requiring the developer to learn and modify the graphical model whereas the current work has no such requirement.

Some of the following related works deal with attack surface identification and reduction at the code or binary levels, whereas this work deals with it at the architectural level. A few of these works reduce the attack surface by removing unnecessary code or features similar to the removal of PD locations in this work. A. Kurmus et al. [7] look at reducing the attack surface of commodity OS kernels by identifying code that is not used and removing it or preventing it from executing. T. Kroes et al. [8] investigate reducing the attack surface through dynamic binary lifting, removal of unnecessary features, and recompilation. R. Ando [9] presents work on attack surface reduction through call graph enumeration in which attackable call graphs are removed. S. N. Bukhari et al. [10] propose reducing the attack surface corresponding to cross-site scripting by employing secure coding practices. G.V. Neville-Neil [11] writes that "the best way to reduce the attack surface of a piece of software is to remove any unnecessary code". M. Sherman [12] looks at attack surface identification only and investigates attack surfaces for mobile devices. This author claims that mobile devices exhibit attack surfaces in capabilities, such as communication, computation, and sensors, that are generally not considered in current secure coding recommendations.

Some works propose to increase security through attack surface expansion rather than attack surface reduction. For cloud services, T. Al-Salah et al. [13] propose three attack surface expansion approaches that use decoy virtual machines co-existing with the real virtual machines in the same physical host. They claim that simulation shows that adding the decoy virtual machines can significantly reduce the attackers' success rate. For enterprise networks, K. Sun and S. Jajodia [14] propose a new mechanism that expands the attack surface, so that attackers have difficulty in identifying the real attack surface from the much larger expanded attack surface. Note that these works do not contradict reducing the attack surface to improve security, since the real attack surface is not expanded. The attack surface only appears to be expanded due to the addition of decoys.

VI. Conclusions and Future Work

This work has presented methods for reducing the attack surface for private data held within an online computer system. The methods are intended to be applied at the architectural level early in the development cycle prior to coding, as part of the Design for Security toolset.

Applying the methods does not require developers to learn a new model or a new coding language. Apart from the methods themselves, which are straightforward, a minimal level of security knowledge is needed, in order to understand the concept of attack surface, the purpose of the methods, and how they work. Knowledge of the computer system is the major requirement, but developers already have this knowledge. Although the methods themselves are straightforward, applying them can be challenging in terms of their impact on performance, ease of maintenance, and so on, as mentioned above.

Future work includes refining the methods from developer feedback, obtained perhaps through workshops and trials. Other future work consists of investigating new methods for reducing the attack surface and looking at tools that could indicate a method's impact on such aspects as performance, reliability, ease of maintenance, and implementation costs.

References

[1] Business Insider, "The 21 Scariest Data Breaches of 2018," [retrieved: Sept., 2019] https://www.businessinsider.com/data-hacks-breaches-biggest-of-2018-2018-12

[2] G. Yee, "Visualization and Prioritization of Privacy Risks in Software Systems," International Journal on Advances in Security, issn 1942-2636, vol. 10, no. 1&2, pp. 14-25, 2017, [retrieved: Sept., 2019] http://www.iariajournals.org/security/

[3] C. Salter, O. Sami Saydjari, B. Schneier, and J. Wallner, "Towards a Secure System Engineering Methodology," Proceedings of New Security Paradigms Workshop, Sept. 1998, pp. 2-10.

[4] P. K. Manadhata and J. M. Wing, "An Attack Surface Metric," IEEE Transactions on Software Engineering, vol. 37, no. 3, pp. 371-386, May/June, 2011.

[5] T. DeMarco, Structured Analysis and System Specification, Prentice Hall, May, 1979.

[6] G. Yee, "Modeling and Reducing the Attack Surface in Software Systems," Proceedings, 11th Workshop on Modelling in Software Engineering (MiSE'2019), May 2019, pp. 55-62.

[7] A. Kurmus, A. Sorniotti, and R. Kapitza, "Attack Surface Reduction for Commodity OS Kernels: Trimmed Garden Plants May Attract Less Bugs," Proceedings of the Fourth

 ISBN: 978-1-61208-746-7

European Workshop on System Security (EUROSEC '11), April 2011, article no. 6 (no page number available).

[8] T. Kroes et al., "BinRec: Attack Surface Reduction Through Dynamic Binary Recovery," Proceedings of the 2018 Workshop on Forming an Ecosystem Around Software Transformation (FEAST '18), October 2018, pp. 8-13.

[9] R. Ando, "Automated Reduction of Attack Surface Using Call Graph Enumeration," Proceedings of the 2018 2nd International Conference on Management Engineering, Software Engineering and Service Sciences (ICMSS 2018), January 2018, pp. 118-121.

[10] S. N. Bukhari, M. A. Dar, and U. Iqbal, "Reducing Attack Surface Corresponding to Type 1 Cross-Site Scripting Attacks Using Secure Development Life Cycle Practices," Proceedings of the 4th International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB-18), February 2018, pp. 1-4.

[11] G. V. Neville-Neil, "Reducing the Attack Surface," Communications of the ACM, vol. 61, issue 2, pp. 27-28, February 2018.

[12] M. Sherman, "Attack Surfaces for Mobile Devices," Proceedings of the 2nd International Workshop on Software Development Lifecycle for Mobile (DeMobile 2014), November 2014, pp. 5-8.

[13] T. Al-Salah, L. Hong, and S. Shetty, "Attack Surface Expansion Using Decoys to Protect Virtualized Infrastructure," Proceedings of the 2017 IEEE International Conference on Edge Computing (EDGE), June 2017, pp. 216-219.

[14] K. Sun and S. Jajodia, "Protecting Enterprise Networks through Attack Surface Expansion," Proceedings of the 2014 Workshop on Cyber Security Analytics, Intelligence and Automation (SafeConfig '14), November 2014, pp. 29-32.

 ISBN: 978-1-61208-746-7