POLICY BRIEF

ISPI

Policy Brief no. 224, October 2013

# A PRACTICAL METHOD FOR MINIMIZATION OF ATTACK SURFACES IN INFORMATION WARFARE

Daniel Shoemaker

This article discusses a practical mechanism to guard against cyber attacks by reducing the application's exposure to hostile intentions. The article provides the underlying assumptions and theory of attack surface reduction. It then relates practical steps that the user can take to that theory. The outcome of those steps will be a more robust and secure application environment that will increase the effectiveness of defensive measures

***Daniel Shoemaker****, University of Detroit Mercy.*

The opinions expressed herein are strictly personal and do not necessarily reflect the position of ISPI.
The ISPI online papers are also published with the support of Cariplo

*Hence that general is skilful in*
*attack whose opponent does not*
*know what to defend;*
*And he is skilful in defense whose*
*opponent does not know what to*
*attack.* (Sun Tzu, 496 BC)

### Defensive Warfare in a Virtual World

Not much has changed in the 2500 years since Sun Tzu wrote those words. From Chancellorsville to Normandy the aim of any successful attacker is to "hit them where they don't expect it". So the goal of defensive planning has always been to ensure that the attacker's only option is a thoroughly prepared defense. Tactically, that is best accomplished by constraining an adversary's knowledge of the actual form and disposition of the preparation. Clausewitz summed this strategy up in his Principles of War;" we must at every instant be on the defensive and thus should place our forces as much under cover as possible" (1). Two thousand years earlier Sun Tzu summarized the outcome of that tactic when he wrote, "weakness comes from having to prepare against attacks, strength comes from compelling our adversary to make those preparations against us" (2).

*So the goal of defensive planning has always been to ensure that the attacker's only option is a thoroughly prepared defense. Tactically, that is best accomplished by constraining an adversary's knowledge of the actual form and disposition of the preparation*

Nonetheless, the fundamental challenge of attacks in cyberspace is that there are no physical limitations and so attacks can originate from an unlimited number places and be of an uncharacterizable range of types. As a result, cyberspace is the perfect venue for asymmetric warfare. In practice, cyberdefenders have tried to address the problems of asymmetric attacks by applying another one of Clausewitz's principles, "not to bring all our troops into combat immediately" (1), which is known in military circles as "defense in depth". Sun Tzu sums up the general conduct and aims of maintaining defense in depth when he wrote that, "The good fighters of old first put themselves beyond the possibility of defeat and then waited for an opportunity of defeating the enemy" (2).

Defenses based on defense in depth have always worked well in the physical universe. However the problem with defense in depth in cyberspace is that electronic attacks take place in nanoseconds, so they are finished before any human response can be deployed. Therefore, the only possible counter to a cyber-attack is an automatic response. Nevertheless, the need to have a pre-programmed response in place prior to an attack raises an obvious question, which is "how do you pre-program a response to an event that you can't anticipate"?

This paradox has to be resolved however, since Congressional testimony estimate that it would only cost about $5 million dollars for any nation state or individual attacker to destroy major power grids, eliminate water and sanitary service, Induce mass flooding, release toxic/radioactive materials, or bankrupt any business (11). The resolution that we propose here mirrors the advice of both Sun Tzu and Von Clausewitz, which is to present the narrowest front possible, both in the system's interfaces as well as in the underlying code (9). In effect, in tactical terms we are proposing ways to shorten the defense in an environment where that front is entirely virtual.

### Minimizing Attack Surfaces

In cyberspace the attack surface is the part of the system that is accessible to unauthenticated users (4). That includes mechanisms from application interfaces to generic operating system services (8). The boundaries of the attack surface can encompass any input, operation, or service request that can be performed from the system interface (4). Specifically, an attack surface is vulnerable if there are no "specific separations, or dedicated functional controls for a given attack vector" (3).

*In cyberspace the attack surface is the part of the system that is accessible to unauthenticated users. That includes mechanisms from application interfaces to generic operating system services*

In order to fit attack surfaces into the context of the discussion we need to set the boundaries of the problem. The attack types we are talking about involve the adversarial actions of any unauthenticated entity against the defensive perimeter of the system (8). The other category of attack is one that is launched by trusted people from inside the system. Those types of attacks also fit in the context of a discussion of attack surfaces but there are so many differences in the tactics that would be employed to defend against them that a separate conversation is required.

Since the category of attack being discussed here originates by definition from outside the perimeter the most sensible option for the defense is to reduce the size of that perimeter (8). By limiting the scope of any feasible attack to a few accessible points, each of those points can then be rigorously controlled and normal defense in depth measures can be implemented (8). This fundamental principle of restriction and rigorous control of accessible points provides the justification for the attack surface limitation strategy that we are suggesting here.

Being able to limit the attack surface to a defensible front enables a number of good things. First the creation of an optimum attack surface allows the defense to marshal its forces to protect only those places that can possibly be attacked, such as defined portals and interfaces. That will ensure that the maximum defensive reserve is available at the point of attack (8). Second, a limited and well defined attack surface allows the monitoring systems at the access points to build up a picture over time of what constitutes "normal" traffic across the perimeter (7). Knowing what constitutes normal makes it a lot easier for the system to identify anomalies at machine speeds, specifically where those anomalies might represent the signature of new or unique attack (7). Third a limited attack surface ensures that any successful attack can be better contained at the point of concentration, which will allow the organization to locate its most valuable assets furthest from any conceivable point of attack (8). That is the concept that best justifies a defense in depth strategy.

## Mapping the Terrain

Because cyberspace is intangible, the first practical step in building an effective defense is to define a specific perimeter to defend. Practically speaking this boundary has to be drawn in a way that will allow all stakeholders to see and agree on its form. Drawing that line would seem like an impossible task given cyberspace's lack of substance. But fortunately all current computers have two universal features, which can be utilized to ensure a substantive definition, which is the simple fact that all electronic data processing systems do nothing more than process data. Accordingly, cyberattacks can only target two things the processing, or a data component of the system.

*Cyberspace is intangible, the first practical step in building an effective defense is to define a specific perimeter to defend. Practically speaking this boundary has to be drawn in a way that will allow all stakeholders to see and agree on its form*

Given that basic provision, the specific requirements for proper defense requires the development of an explicit understanding of how each of those two global entities, data and processing, can be attacked. That consideration needs to take place within the constraints of a given situation. Michal Howard proposes that attacks comprise three distinct elements targets, enablers and channels (5). According to Howard, targets are the specific processes or data resources that an adversary aims to control, while enablers are all of the other processes and data resources that the attacker might employ to obtain that control (5). In effect, the attacker gets control over those enabling resources by exploiting vulnerable communication channels or protocols (5).

*Given that basic provision, the specific requirements for proper defense requires the development of an explicit understanding of how each of those two global entities, data and processing, can be attacked. That con-sideration needs to take place within the constraints of a given situation*

Targets are just that. They are a specific end that the attacker seeks to achieve. In practice targets are usually data of value, or system processes that the attacker wishes to control (5). Enablers are any of the other processes running within the system that the attacker could use to achieve their selected purposes (5). Moreover, the only characteristic that differentiates an enabler from the target is that the enabler is by definition not the target (5). Finally, because computers rely on communication channels to access both enablers and targets. These three elements, targets, enablers and channels give us a starting point to characterize the potential attack vectors against any system.

The practical steps that can be taken to limit an attack surface should start with the identification and control of communication channels within the system.  Channels are enabled by system processes. They have to be created, in code and they have to be configured to operate as intended. Thus, channels are an intentional part of the system design process. Properly designed each channel should be directly associated with the data that it has been created to access. Likewise, given the fact that channels exist to enable access it becomes possible to construct a map of the system's terrain for the purpose of planning a virtual defense by focusing on the channels and their access rules.

Given the ability to characterize channels and their enabling processes, the specific approach to minimizing the attack surface involves mapping potential targets for attack to the communication channels that access them and then characterizing the enablers within those channels. Tactics for optimum defense in depth against any known attack will flow from that knowledge. However since successful attacks by their nature usually involve doing something that a conventional defender wouldn't think of, it is important to trace each potential attack vector while considering every conceivable way that attack can be successful.

### Three Principles and a Corollary

Since it is reasonable to assume that an attacker has a specific aim the first step in the process is the identification of all reasonable targets in the system. All channels and protocols that access those targets must then be identified and mapped. All enablers on those channels are then be identified down to the machine level processes and any known or potential vulnerability should be characterized and countered by an explicit mechanism to ensure proper control.

*All enablers on those channels are then be identified down to the machine level processes and any known or potential vulnerability should be characterized and countered by an explicit mechanism to ensure proper control*

Generally, because those targets will have differing value they can be prioritized. Prioritization of attacker goals is important because defense in depth demands it. That is, in the pragmatic world it is impossible to secure everything. Therefore, the things that are the most important have to have greatest amount of assurance and so on down the priority list to items that might be left entirely outside of the defense because the effort to secure them cannot be warranted. It is in this final respect that the steps that are taken to control attack surfaces have a business justification. That is because in order to be effective the defense can only provide optimum protection within available resources. The planning that underlies the limitation of attack surfaces can ensure that those resources are concentrated at the points that justify the greatest amount of protection.

The approach to attack surface limitation in cyberspace is geared around three fundamental principles. The first is the reduction of the amount of code available to unauthenticated, users (4). The basic logic for this principle just assumes that the more ways that an unknown, or untrusted user can interact with a piece of software, or a system, the more vulnerable that software or system will be (8, 7). That simple rule of thumb applies to unauthenticated user interactions at both the entry and exit points of the software or system item. As a result, adding up the number of points where data passes into or out of a program or system should provide a rough estimate of the degree to which that program or system is vulnerable (8). And accordingly the practical steps for attack surface minimization in the case of this principle would be to ensure that the data exit and entries are purposely restricted to an optimum number of rigorously controlled access points.

For example if the access points in an application are restricted it is logical to assume that the two most common weaknesses in the Common Weakness Enumeration (CWE), SQL Injection and OS Command Injection, would be reduced. That is because both of these weaknesses require specific programming steps to ensure that potentially malicious arguments are escaped, or filtered. The problem occurs where the developer fails to embed a sufficiently rigorous whitelist in the programming, or fails to create strict boundaries between the input process and the operating system. Both of these conditions take time and effort to create. So if a large number of access points have to be ensured, then the likelihood of either incorrectly configuring, or worse yet forgetting, an access point is commensurately increased. If, on the other hand, the number of access points is limited, programmers can concentrate on making each access point optimally secure.

*If the access points in an application are restricted it is logical to assume that the two most common weaknesses in the Common Weakness Enumeration (CWE), SQL Injection and OS Command Injection, would be reduced. That is because both of these weaknesses require specific programming steps to ensure that potentially malicious arguments are escaped, or filtered*

The second principle for attack surface limitation involves restricting the default privileges of any running application to the lowest practical level (4). This action enforces the universal cybersecurity principle of "least privilege" which in the physical universe denotes the fact that the user, or client, is given privilege sufficient to allow them to carry-out whatever tasks they have been assigned. The problem is that program and operating system default settings are often not attuned to the security requirements of the real-world. Worse, these settings might be designed to ensure that maximum degree of access across the perimeter. As a result, the systematic characterization and appropriate restriction of each program's default privileges can go a long way toward reducing the attack surface and by so doing ensure an overall defensible system (8, 7).

If the default value isn't used, privilege settings for applications that cross the system boundary have to be defined by the system's administrator. That creates another time and effort problem that can be resolved if the number of applications that can cross the perimeter is limited to the correct set. The identification and control of those applications is, in effect, a tactic for attack surface limitation. The outcome of that limitation would be that applications are not allowed unauthorized access due to the assignment, or accumulation of improperly elevated privileges. This is not a technical exercise as much as it is one of good system administration so it can be easily implemented in most systems.

Finally attack surfaces can be reduced by simply restricting the amount of running code within the system. On the surface this might seem like an impractical suggestion since people ordinarily assume that any code that is running in a system is there for a purpose. The fact is that the features that are running in most systems might be there for all sorts of reasons, none of which have anything to do with utility. Consequently a careful inventory of the various programs and processes that are active in the system's operating stack can reveal unnecessary or irrelevant services (3, 4). Examples of this would be features that are utilized by relatively few users, or which continue to be loaded simply because they were once needed (4). The elimination of each unnecessary service will reduce the amount of running code and thereby reduce the attack surface of the system. Even so, the elimination of running processes should be done with some careful preparation and thought since users would not be happy if the application they were using just disappeared one day, so an alternative to eliminating an application that is important to a small number of users would be to highly restrict the ability to run that program to only those few users who are authorized to use it (4).

*Attack surfaces can be reduced by simply restricting the amount of running code within the system. On the surface this might seem like an impractical suggestion since people ordinarily assume that any code that is running in a system is there for a purpose*

The corollary to these principles is the definition of privileges to access an enabling process. Definition of privilege is a classic in formation assurance chore that is a critical step in ensuring effective attack surfaces. Logically, the extent of the attack surface is going to be directly proportional to the number of enablers and channels it encompasses. In computing, the definition of privilege is the all-purpose mechanism for controlling access to enablers and channels. Therefore privilege definition is, in effect, the process that substantively defines the form of the attack surface. In practical use, privileges traditionally apply to three fundamental actions within the system, read, write and run. Therefore decisions about the degree of privilege assigned to an untrusted user for any channel allow explicit control over access to any given target within the system. The target itself can be a data item of vale, or it can be a process that accesses data, of value or any process that indirectly leads to the target. So it should also be kept in mind that enabling processes can be targets, when mapping the defensive terrain.

Of course controlling the access privileges for all accounts in something like a Windows 7 environment is not a trivial or easy task (10). That is because there are always a complex set of trust variables involved in the assignment of rights in the conventional world of business and all of those potential variations in trust have to be factored into the assignment of privileges for each entity requesting access (10). Nevertheless, the optimum restrictions have to be applied in each case to ensure the most defensible attack surface possible. Thus the real-world process of defining privilege can be both time consuming and costly. But, the outcome of the time spent thinking through an effective set of restrictions will be a system that is optimally prepared to defend itself against the types of unconventional attacks that characterize asymmetric warfare.

## Conclusions

This paper has proposed a specific approach to building a robust defense against asymmetric attacks. It centers on the restriction of the attack surfaces across an organization's systems. A practical method for creating and enforcing limitations to the attack surface of the organization was provided here. This method is based around target and attack enabler identification and limitation of access rights through the enabling channels. The specific aim of the approach is to only enable access through the system perimeter at a limited number of well defended interface points. It is implicit that if the limitation process is correctly executed the defender will be able to provide a robust defense in depth at each of the designed points of access.

*The specific aim of the approach is to only enable access through the system perimeter at a limited number of well defended interface points. It is implicit that if the limitation process is correctly executed the defender will be able to provide a robust defense in depth at each of the designed points of access*

Because the limitlessness of cyberspace poses unconventional challenges this approach centers on restricting the ways an untrusted outsider can approach the system. In general, those limitations are implemented through relatively simple system configuration and maintenance steps. However, in order to ensure a highly controlled environment it is necessary to pay attention to the details of system operation and how those operations impact the critical assets within the system. It is then assumed, that if all priority targets can be identified and the channels and protocols that enable access can be characterized it will be possible to take the necessary system administration steps to ensure that access along those channels is properly controlled by specifically designating and monitoring the read, write, or run privileges for all users.

Nonetheless, the question still remains, why are the actions outlined here something that an organization should consider, especially given the time and cost required to prepare and maintain an effective, substantive defense? The potential for asymmetric and unconventional attacks is a reality in cyberspace. That is because the nature of the environment lends itself to that kind of warfare. Whether the adversary is a nation state, or a jihadist with an uplink to the internet it is presently far too easy to cause serious harm though the inadequately defended systems that underlie our national cyber-infrastructures. Given the swiftness of technological change it is excusable that organizations might not have sufficiently prepared themselves to defend their priority assets. It is inexcusable however to know that those attacks are likely to occur and stand idly by without doing anything about the situation. This paper provides one suggested way of doing something about that problem.

## References

1. Gatzke, Hans (ed.), *Clausewitz, Principles of War*, Military Service Publishing Company, 1942

2. Giles, Lionel (trans. 1910), *Sun Tzu on the Art of War 2010*, at http://classics.mit.edu/Tzu/artwar.html

3. Herzog, Pete, *OSSTMM Open Source Security Testing Methodology Manual*, Institute for Security and Open Methodologies (ISECOM), 2010

4. Howard, Michael, *Mitigate Security Risk by Minimizing the Code you Expose to Users*, MSDN Magazine, November, 2004

5. Howard, Michael, Jon Pincus, and Jeannette M. Wing, *Measuring Relative Attack Surfaces*, Proceedings of Workshop on Advanced Developments in Software and Systems Security, Taipei, December 2003

6. Manadhata, Pratyusa K., and Jeannette M. Wing, *An Attack Surface Metric*, IEEE Transactions on Software Engineering, Vol. XX, No. X, 2010

7. Manadhata, Pratyusa K., Kymie M.C. Tan, Roy A. Maxion, and Jeannette M. Wing, *An Approach to Measuring A System's Attack Surface*, CMU Technical Report CMU-CS-07-146, August 2007

8. Northcutt, Stephen, *The Attack Surface Problem*, 2011, at http://www.sans.edu/research/security-laboratory/article/did-attack-surface

9. Pincus, Howard, M.J., Wing, J., *Measuring Relative Attack Surfaces* in Proceedings of Workshop on Advanced Developments in Software and Systems Security, 2003

10. Rowe, David, *Windows 7: Take Back Control by Managing Windows Access Rights*, «TechNet Magazine», January 2011

11. Ruus, Kertu *Cyber War I: Estonia attacked from Russia*, «European Affairs», FindArticles.com. 9 February 2012.

12. Wang, Ke, "Anomalous Payload-Based Network Intrusion Detection", International Symposium on Recent Advances in Intrusion Detection Vol. 3224, No.7, Valbonne (France), 2004, pp. 203-222