# 8 The Data-Driven Computer Defense Lifecycle

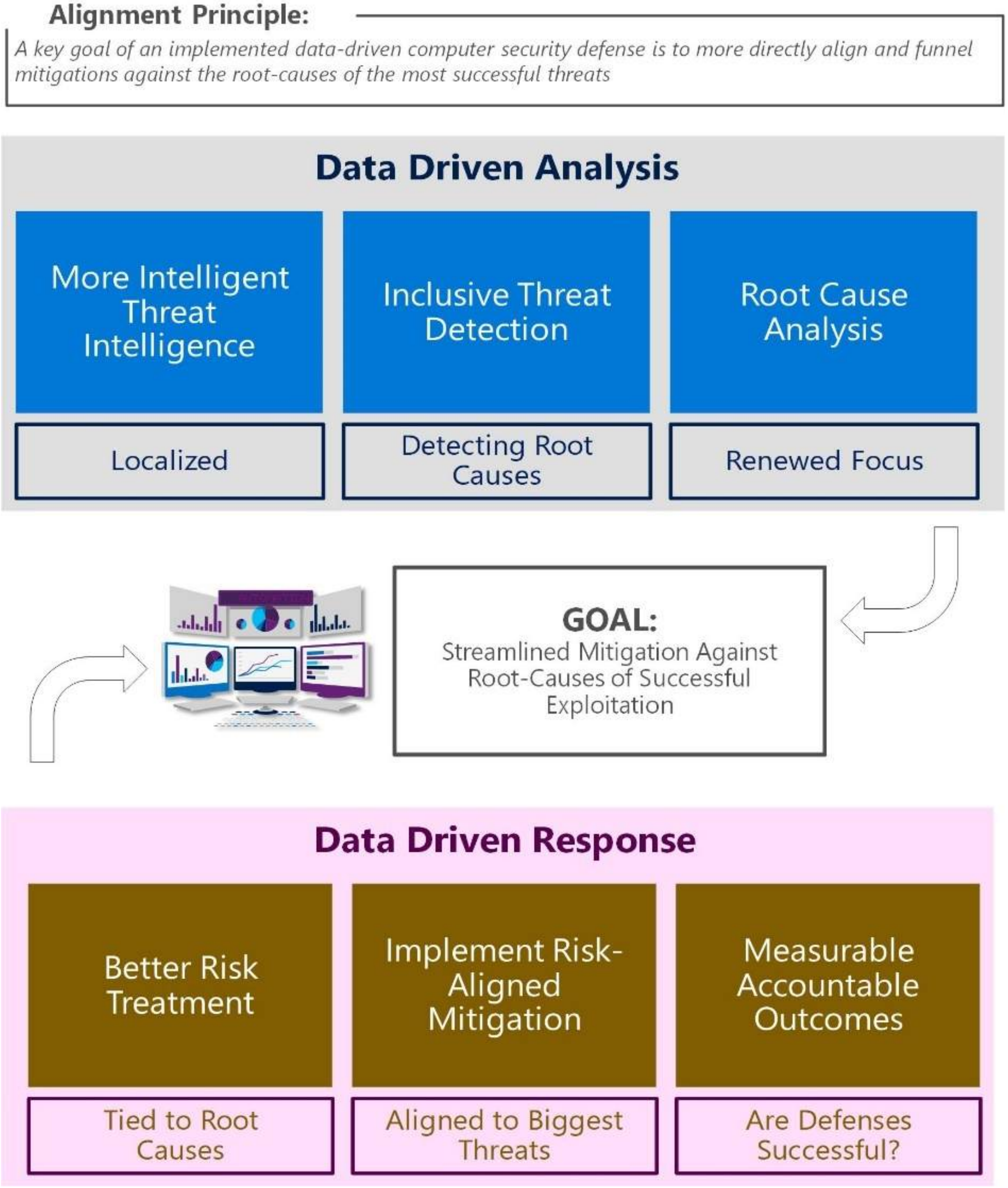
Chapter 8 examines the Data-Driven Computer Defense lifecycle and discusses other defense-in-depth defenses that should be included.

> "*I know of no more encouraging fact than the unquestionable ability of man to elevate his life by a conscious endeavor*."—Henry David Thoreau

# Recap

As summarized in the figure below, a Data-Driven Computer Defense uses data-driven analysis to better identify the most damaging current and future most likely successful threats and uses them to more efficiently align mitigations.

*A data-driven analysis leads to a data-driven streamlined response.*



**Alignment Principle:**

*A key goal of an implemented data-driven computer security defense is to more directly align and funnel mitigations against the root-causes of the most successful threats*

**Data Driven Analysis**

More Intelligent Threat Intelligence

Inclusive Threat Detection

Root Cause Analysis

Localized

Detecting Root Causes

Renewed Focus

**GOAL:**

Streamlined Mitigation Against Root-Causes of Successful Exploitation

**Data Driven Response**

Better Risk Treatment

Implement Risk-Aligned Mitigation

Measurable Accountable Outcomes

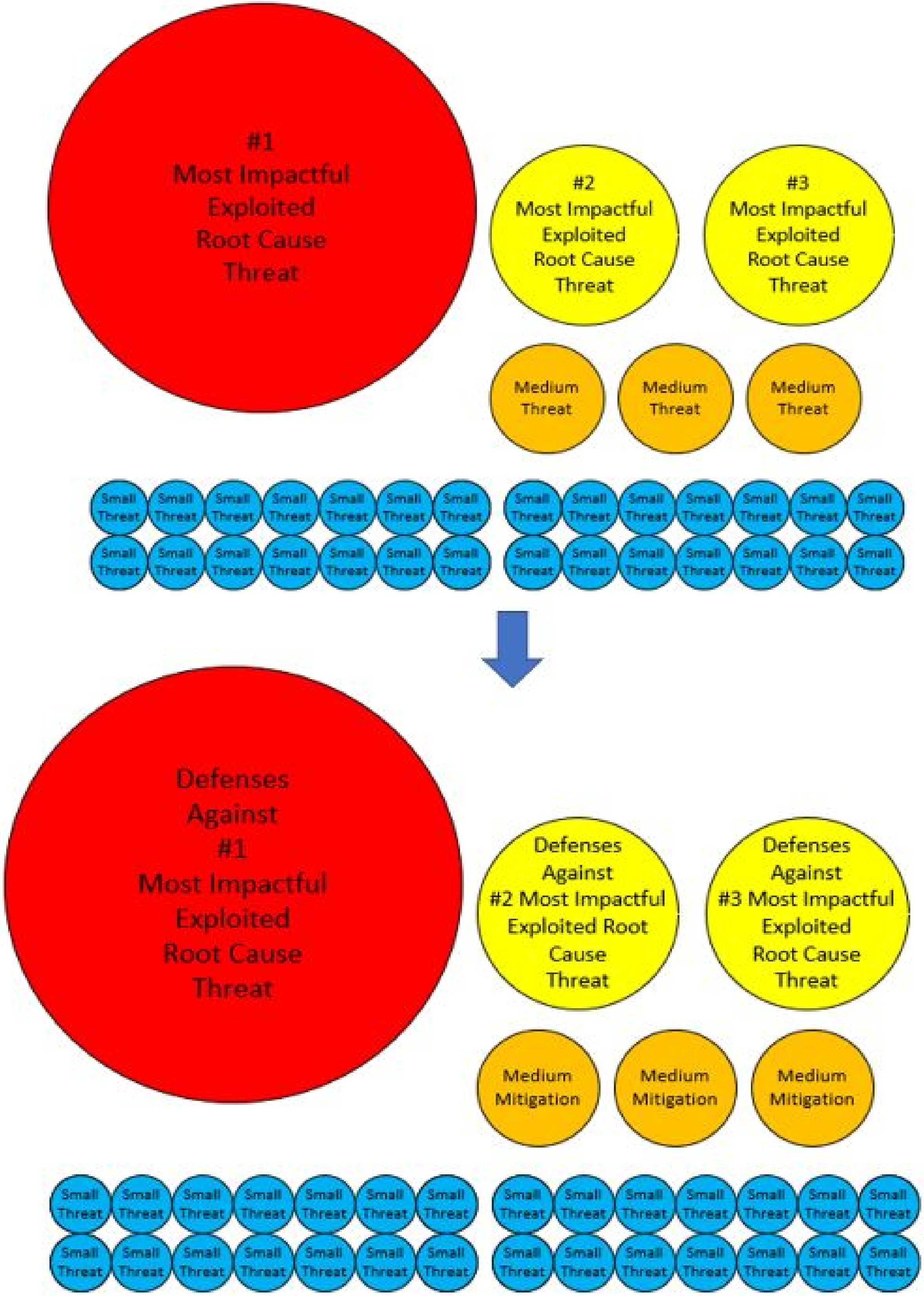
Tied to Root Causes

Aligned to Biggest Threats

Are Defenses Successful?

Instead of expending energy on remote and less likely to be successful threats, defensive efficiency is improved by focusing on local threat intelligence data, root causes of initial exploitation (e.g. unpatched software, social engineering, etc.), and improving threat detection through gap analysis review. These steps help defenders make a better risk assessment for each individual threat, compare the risk of different threats against each other, and identify the most likely damaging successful threats. After choosing which threats to focus on, data-driven defenders can better apply the right mitigations in the right places in the right amounts at the right time to achieve measurably lower risk and accountable outcomes. This results in more efficient mitigations against the top threats and root causes (as shown in the figure below).

*Risk-ranked threat perception leads to risk-ranked defenses*


#1 Most Impactful Exploited Root Cause Threat
#2 Most Impactful Exploited Root Cause Threat
#3 Most Impactful Exploited Root Cause Threat
Medium Threat
Medium Threat
Medium Threat
Small Threat
Defenses Against #1 Most Impactful Exploited Root Cause Threat
Defenses Against #2 Most Impactful Exploited Root Cause Threat
Defenses Against #3 Most Impactful Exploited Root Cause Threat
Medium Mitigation
Medium Mitigation
Medium Mitigation
Small Threat

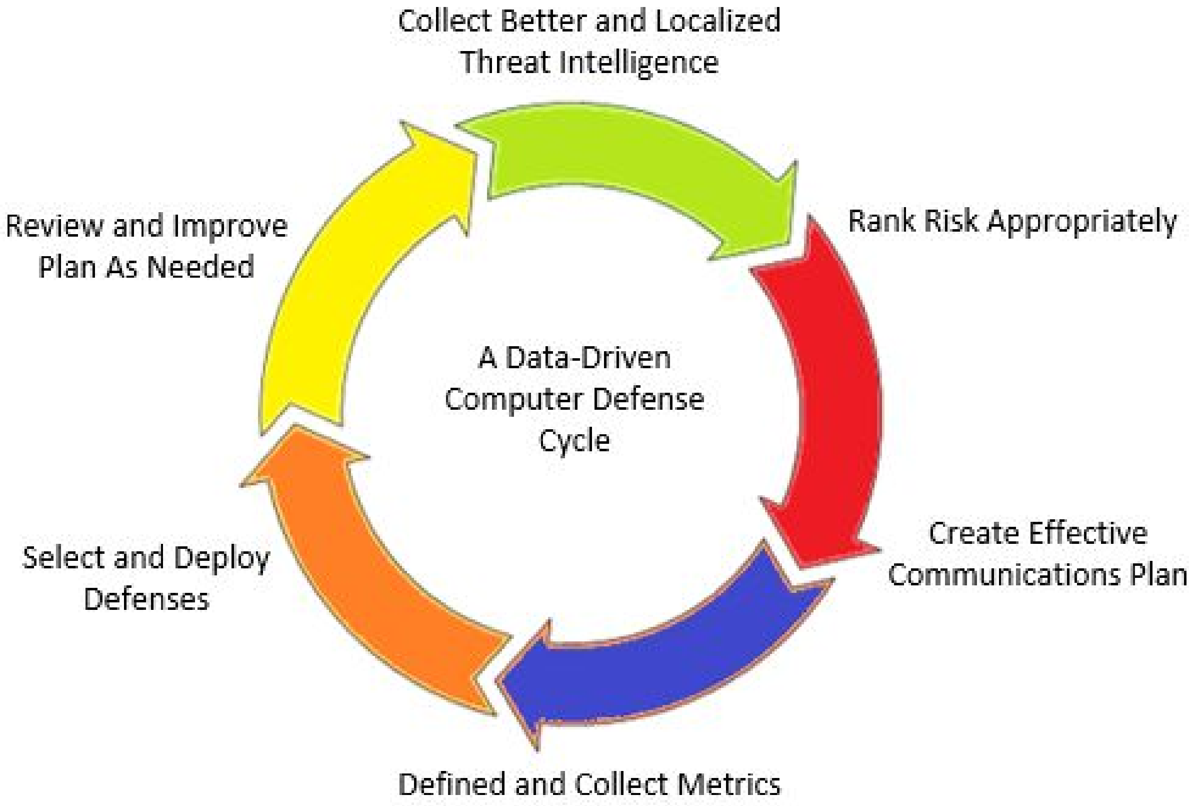
May decide that the cost of defending against small threats is not a good business decision

In order for the complete benefits of a data-driven defense to be fully recognized, it has to become part of the organization’s culture and be broadly implemented.

## The Data-Driven Computer Defense Lifecycle

A Data-Driven Computer Defense has a lifecycle (summarized below), beginning with collecting better threat intelligence, ending by holding existing mitigations accountable against expectations, and then starting the process all over again.

*The Data-Driven Computer Defense lifecycle*


Collect Better and Localized Threat Intelligence
Rank Risk Appropriately
Create Effective Communications Plan
Defined and Collect Metrics
Select and Deploy Defenses
Review and Improve Plan As Needed
A Data-Driven Computer Defense Cycle

Let’s explore each stage of the lifecycle further.

## Collect Better and Localized Threat Intelligence

Nothing is more important to a computer security defense than the first step of improving threat intelligence. More accurate threat intelligence that focuses on current, local, most successful threats first, followed by the most likely successful future threats, helps better define what the top threats (by damage) are. Improving local threat intelligence and detection is crucial for making an efficient data-driven defense plan.

## Rank Risks Appropriately

Once all the biggest threats are known, they can be more clearly ranked against each other. You’re going to give less emphasis to the old way of ranking threats by blindly accepting a vendor’s pronouncement that a particular threat is a “high priority”. Instead, you’re going to use the improved threat intelligence of your own organization’s experience to drive your ranked threat list.

## Create an Effective Communications Plan

Once all the top threats are known and ranked, they should be communicated across the organization, in accordance with their actual local risk. In most cases, you'll have a number one top threat, perhaps followed by one, two, or a handful of other big threats. These are usually followed by dozens of threats that all together don't usually account for the risk accumulated by one of the top threats.

You need to clearly communicate what the top threats are to each group in the organization, according to the level of detail and strategy needed for each group. For example, the CEO will have one level, and the CIO and the CSO will have another. Those positions want to hear about your long-term strategy for how you are dealing with the top threats, along with compliance issues, money, resources, and other C-level concerns. A front-line employee, not in the IT department, probably needs less strategic detail and more education about how the threat will be impacting their position specifically (e.g. technical defenses coming their way or new education, etc.) and how they can help. An effective communications plan gives the right education in the right places, all focused on eliminating top threats.

## Define and Collect Metrics

Once the top threats are identified, everyone across the organization should come together to identify potential metrics and discuss how to collect them or make existing ones more accurate (if needed). Metrics help drive the well-oiled machine that is a Data-Driven Computer Defense. Gut feelings and experience are backed up or replaced by good data. The mantra for this component should be “If you can’t measure it, you can’t do it.”

## Select and Deploy Root Cause Defenses

A Data-Driven Computer Defense plan focuses on root causes (e.g. unpatched software, social engineering, misconfiguration, human errors, etc.) to create mitigations. You cannot defeat a car thief by worrying about the brakes after the car is stolen. You cannot defeat malware by worrying solely about the accuracy of your antivirus software, which will never be perfect. Every threat uses a root cause exploit method to break into an organization. Only be reducing root causes will you reduce the threats. Stop one malware program, and you stop one malware program. Stop one root cause exploit avenue, and you stop every malware program (and hacker) that might otherwise have used that root cause to be successful.

## Review and Improve the Plan as Needed

The entire Data-Driven Computer Defense process is a cyclical journey from start to finish and back again. At its core, it’s a moving, constantly changing plan, much like the threats it is trying to minimize. The key question at any given point in the cycle is “Are there any deficiencies that need to be improved?”

You can begin that process by asking the following questions:

- Is threat intelligence accurate about the top current and future most likely SUCCESSFUL threats?
- Is threat detection of the top threats accurate? Are there too many false-negatives or false-positives? Are there some top threats that you are missing altogether?
- Are emerging threats being seen and dealt with faster?
- Are root causes being identified and acted upon?
- Are communications focusing on the right things and communicating them across the organization? Can all employees name the top successful threats?
- Are the right mitigations being applied, and how do they succeed?

These questions should be asked all the time, and at the very least they should be formally addressed and documented once per quarter. Significant deficiencies should be discussed and rectified, if possible, and be cost-effective.

## The Perfect Data-Driven Computer Defense

In a perfect world where a data-driven defense is pushed to its fullest efficiency, as much of the process would be automated as possible. Data analytics would be used to process local threat intelligence, which would drive automatic risk ranking. Relative risk rankings, along with artificial intelligence (AI), would drive defenses and help create communication plans. Results in the deployed defenses would be used to drive the next cycle of reviews. Any product or service that helps you to automate a part of the data-driven defense lifecycle should be given serious consideration.

## Defense-in-Depth

A Data-Driven Computer Defense does not mean you stop doing all the other defense-in-depth things (such as credential hygiene, the Security Development Lifecycle, improved authentication, “Assume Breach” defenses, etc.) that are necessary for a complete, encompassing, computer security defense. When car manufacturers were working to significantly improve car safety in the 1970s and 1980s, it didn’t mean that traffic safety stopped being improved.

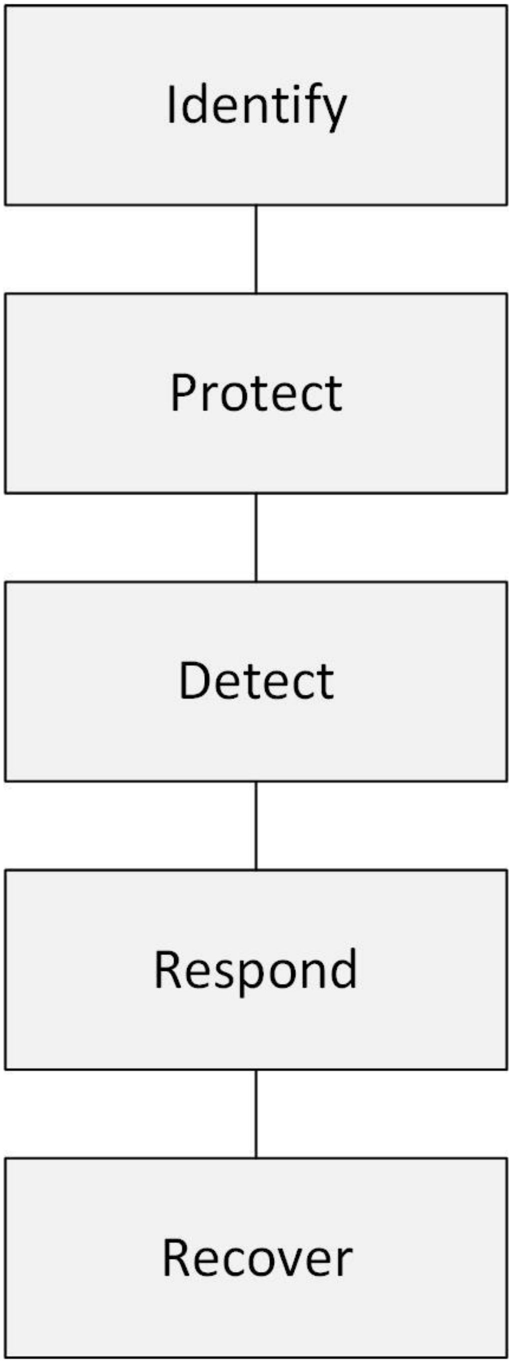
To be clear, with a data-driven defense you absolutely need to focus on doing the right things in the right amounts first and then doing everything else. A data-driven defense attempts to stop people from focusing on all the other stuff first to the exclusion of the biggest, most successful threats and best mitigations. A data-driven defense means you don’t concentrate on implementing smartcards when you should instead focus on badly patched software or anti–social-engineering training, but that doesn’t mean that you altogether ignore smartcards as a part of a good computer security defense, particularly if they can help reduce a significant root exploit cause.

## Part of the Big Picture

A Data-Driven Computer Defense is a conceptual framework and methodology for helping to more efficiently align mitigations against the top threats. Although I believe a data-driven defense plan's key focus is revolutionary, the general processes it follows are not. There are many computer security frameworks, each with similarities and differences, but they are all focused on trying to minimize cybersecurity risk.

For example, the NIST Framework for Improving Critical Infrastructure Cybersecurity (https://www.nist.gov/sites/default/files/documents/cyberframework/cybersecurity-framework-021214.pdf) components, summarized in the figure below, map fairly consistently to the data-driven defense approach.

*The NIST cybersecurity framework components*



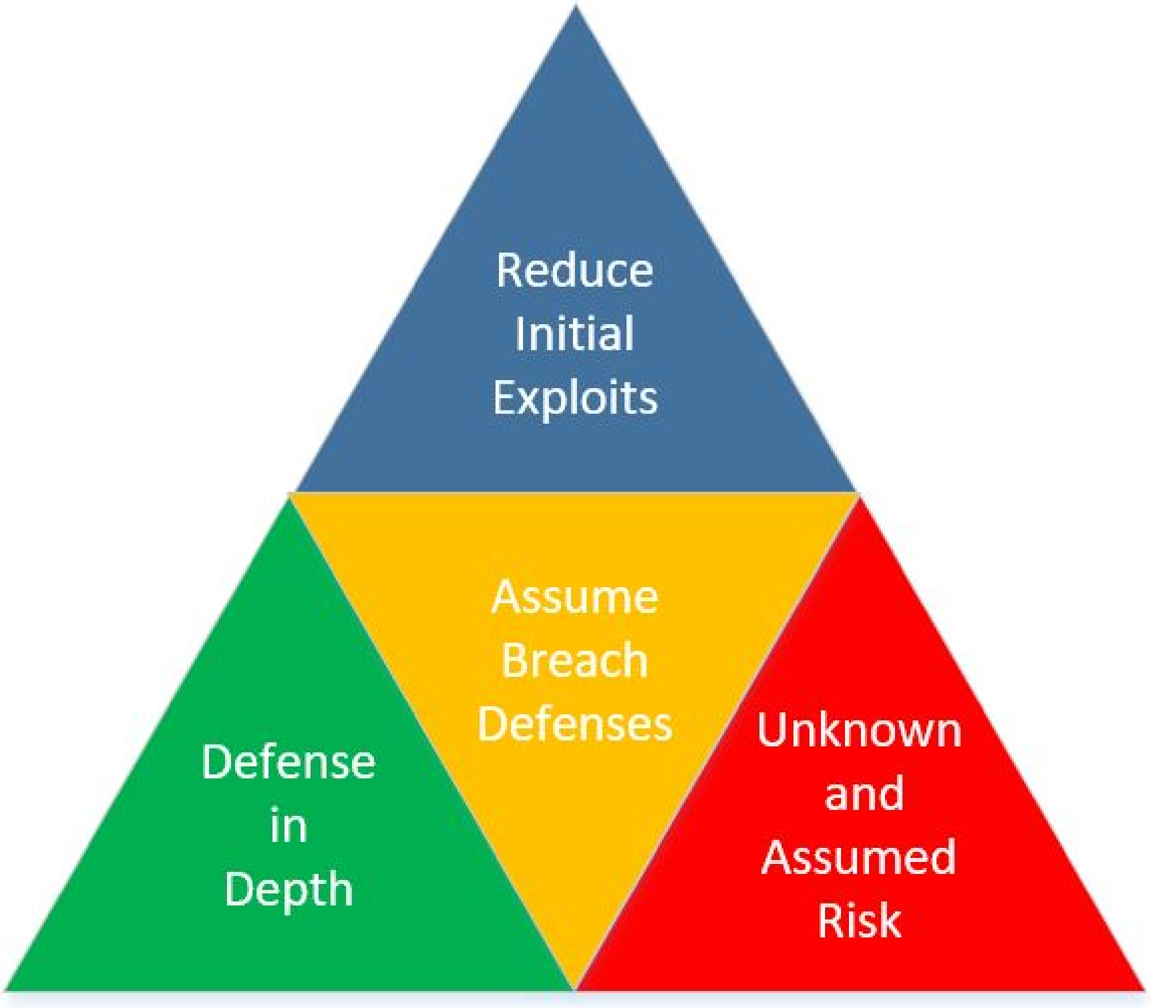
The last two components, *Respond* and *Recover*, refer more to traditional incident response and minimizing resulting damages, which a data-driven defense plan does not cover. This doesn't mean an organization should not include those components in their overall risk framework. It just means it's not something that a Data-Driven Computer Defense plan specifically focuses on because those components are already well covered and agreed upon by most computer security practitioners. Use a data-driven defense plan to better perform the three first NIST framework components, *Identify*, *Detect*, and *Protect*.

> Note: If you're interested in additional, more inclusive IT governance and operational frameworks (more than just cybersecurity), check out ISACA's COBIT (http://www.isaca.org/COBIT/Pages/default.aspx), ISO/IEC 27001: 2013 (https://en.wikipedia.org/wiki/ISO/IEC_27001:2013), or Information Technology Infrastructure Library (ITIL) (https://en.wikipedia.org/wiki/ITIL).

## Four Computer Security Defense Pillars

A Data-Driven Computer Defense uses local experiences and data to drive all the defenses and responses. The overall defense memes can be broken down into four defense pillars (shown below).

*The four computer security defense pillars*



As discussed throughout this book, a computer data-driven defense focuses on exploit root causes and does this specifically to prevent initial breaches. If you can't stop the initial breach, then you will never significantly minimize risk to the organization.

## Use Assume Breach Defenses

Right now, most organizations are so porous that they must assume that they are either currently breached by hackers or malware or easily could be. This is the case with 99.999% of organizations not disconnected from the Internet and not running on a classified network. There is occasionally hacking of organizations that are disconnected from the Internet and running on classified networks, but it is far less common than hacking of organizations that are not.

Use the data-driven defense concepts to measure Assume Breach threats and create right-aligned mitigations to assist. For example, if you assume that a hacker can always capture your local administrator password stored on a single computer, you can slow down the attacker's movement in the organization by not using the same administrator password on any other computers. Or you can implement defenses that make it significantly harder for an intruder to retrieve privileged credentials even if they have admin access to a computer (such as is done by Microsoft's Windows 10 Credential Guard™ technology).

While a data-driven defense absolutely wants you to work the hardest on minimizing initial breach exploits, it recognizes that most organizations can't prevent them 100%, and so a data-driven defense must be done inside and out. You want to try to prevent intruders from getting inside your networks and slow them down if they do get inside.

Most defenders are already operating this way, and just because they deploy very strong network perimeter firewalls doesn't mean they leave a "soft, chewy center" for any attacker or malware that makes it past the perimeter defenses. Instead, use an Assume Breach defense until you can use data to show that it isn't something your organization needs to worry about anymore (and if so, congratulations!). Use Data-Driven Computer Defense concepts for both initial and subsequent malicious actions.

## Individual Defense-in-Depth Recommendations

There are dozens of defense-in-depth models and recommendations for what we all should be concentrating on. The biggest difference between them and the data-driven defense plan promoted in this book is that they often base their recommendations on a society's global experience and recommendations for success, while a data-driven defense plan says that it is best to first use your organization's local, most timely experience.

This doesn't mean the other models have no value. It just means that you should take them for what they are, global recommendations without any idea of what your local, biggest, successful threats are. They have no idea what your defenses are, what your organization already defends well against, and where the gaps are.

If you do not have any data of local experiences, you can start with any of the popular defense-in-depth recommendations from highly respected organizations. Here are some respected and popular recommendations:

General Security Control Recommendations

- Center for Internet Security Top 20 CIS Controls™ (formerly known as the SANS Top 20) (https://www.cisecurity.org/controls/)
- OWASP Top 10: The Ten Most Critical Web Application Security Risks (https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf)
- U.S. Department of Homeland Security Continuous Diagnostics and Mitigation (https://www.dhs.gov/cdm)
- NIST Special Publication 800-53, "Security and Privacy Controls for Federal Information Systems and Organizations" (http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf)

Specific, Detailed Configuration Controls

- Defense Information Systems Agency, or DISA (http://disa.mil/Cybersecurity/Secure-Configuration-Guidance)
- U.S. Government Configuration Baseline, or USGCB, (https://csrc.nist.gov/Projects/United-States-Government-Configuration-Baseline)

- Center for Internet Security CIS Benchmarks™ baselines (https://www.cisecurity.org/cis-benchmarks/)
- Microsoft Windows Security Baselines (https://docs.microsoft.com/en-us/windows/device-security/windows-security-baselines)

A Data-Driven Computer Defense is easy to map to these other control frameworks and specific configuration recommendations.

## Focus, Focus, Focus

As much as I like the frameworks and best practice recommendations discussed in the previous section, the key missing distinction with them is focus. We already know that the vast majority of risk to most organizations is from social engineering and unpatched software. These other recommendations often mention these two top-tier threats, but they don't stress the true importance of doing them well. The average framework document mentioned in the previous section runs from 80 to 100 pages. But the text devoted to patching software and providing security awareness training is probably 5 to 20 sentences. At the same time, items that deal with far less real-world risk have many pages devoted to them. These documents are good, but they treat all threats either equally or disproportionately to actual risk (like bubbles in a glass of champagne).

Additionally, even when these documents cover the most important mitigations, such as patch management and security awareness training, they don't go into very much detail. They say something like "Make sure to patch all critical vulnerabilities in a timely manner." Although that is a good philosophy to follow, it treats all critical vulnerabilities as having equal weight in mitigating risk, and we know from Chapter 5, "A DDD Example", that not all patches are alike.

Every recommended mitigation must be scrutinized to determine which ones have the best chance of diminishing the most risk the fastest. Use and follow respected organizations' frameworks and guidelines, but implement their best practice recommendations in a way that ensures you get the best bang for your buck in reducing risk first.

## My Personal Top Defense Recommendations

I’ve been a computer security consultant for over 30 years, and my clients are among the most secure out there, not only because they follow a data-driven defense, but because I have long been pushing what is THE best bang-for-the-computer-security-defense-buck. Even though every client is different, most share many common issues. Here is the list of the most common computer security defense recommendations I give on a regular basis:

- Improve anti–social-engineering training.
- Improve patch management, focusing first on the applications most exploited in your organization.
- Admins should always use secure admin workstations with very strong security controls, including not allowing access to or from the Internet. See https://msdn.microsoft.com/en-us/library/mt186538.aspx for more details.
- Implement Assume Breach defenses, like preventing privileged credential theft.
- Improve monitoring and alerting.
- Use honeypots. Honeypots are “fake” computers and devices that exist only to detect and alert on anomalous activity.
- Minimize or eliminate permanent elevated group membership. Instead use “just-in-time” and “just-enough” mechanisms.
- Use smartcards and/or two-factor authentications. (It’s hard for hackers to steal passwords when you don’t have them.)
- Implement “always on” encryption for both storage and transferring data.
- Use application control “whitelisting”.
- Use a secure OS following the vendor’s best practice recommendations.
- For the serious security professional, use Qubes OS (https://www.qubes-os.org/) or something like it.
- If you write or develop software, use the Security Development Lifecycle or something similar to decrease security bugs.
- Save all critical data to a secure, less hackable location, like offline storage.

If you don’t see one or two of your favorite recommendations, like firewalls or antivirus software, it might be because I just don’t see them as top, necessary defenses. Or I might have overlooked one when I was writing this book. If you feel I have missed one, send it to me at roger@banneretcs.com, and if I agree, I might possibly add it to the next edition of the book.

A data-driven defense has a lifecycle starting from better threat intelligence and ending with properly aligned, accountable mitigations, which are frequently reviewed. A Data-Driven Computer Defense plan is part of a larger IT security framework, which is part of a larger IT governance framework. Chapter 9 examines more Data-Driven Computer Defense implementation examples.

# 9 More Implementation Examples

Moving to a Data-Driven Computer Defense can only be accomplished on a case-by-case basis. Every organization is different and has a different starting point. Chapter 9 highlights many Data-Driven Computer Defense examples that have been used by some of the world's largest organizations and can be used to spearhead your new projects.

> "*In theory, there is no difference between theory and practice. In practice, there is*."—First appeared in print in the 1986 book *Pascal: An Introduction to the Art and Science of Programming* by Walter J. Savitch

# Moving to a Data-Driven Defense

In an existing organization, it is difficult to move wholesale from a non-data-driven defense to a Data-Driven Computer Defense in a single step. Moving to a data-driven defense usually requires adopting data-driven tenets on a case-by-case, project-by-project basis, starting with the easiest opportunities. Implementers should begin by picking “proof-of-concept” (POC) projects where the data-driven concepts are easy to execute, the data and metrics can be gathered more easily, and the benefits are more readily seen. The summaries in the following sections describe real-life deployments of Data-Driven Computer Defense concepts.

## The Microsoft Backstreet Project

I was working in India as a Microsoft employee teaching a Windows Vista computer security class to other Microsoft employees. During my first day of classes, an IT security employee came in and announced that he had to check each of the class computers for the Conficker malware program (https://en.wikipedia.org/wiki/Conficker).

He had a USB key drive with an antivirus program that he could plug in, and it would automatically run. There were about 50 computers in the classroom, and every computer he tested was infected with Conficker. As he got to about the twentieth computer in a row that was infected, I asked to see his USB key. I disabled the autorun functionality on my computer, plugged in his USB key, and ran my antivirus on the plugged-in drive. Sure enough, it was infected. The IT security employee had unknowingly infected his USB drive on one of the former computers, and he now subsequently infected every single computer that he plugged the USB drive into. It was a self-fulfilling prophecy.

For those defenders who had to fight the Conficker worm, it was one of the hardest malware programs to fully eradicate from any corporate environment. It would go from one or two computers to many more, and you would find infected USB key drives for years. It disabled automated backups (making it harder to recover from), caused network slowness, locked out user accounts, disabled services, stopped antivirus programs from working, and prevented Microsoft patches from being downloaded.

Conficker first appeared in November 2008 and ended up infecting tens of millions of computers over many years. It spread using at least three different exploit vectors, including exploiting a patched Windows vulnerability (MS08-067), password guessing against weakly password-protected NETBIOS drive shares (using a list of 100 very simple but common passwords), and exploiting the USB autorun modality. It went from being just a nuisance to causing significant operational issues for many organizations. It would die down in popularity only to reemerge and start to spread again.

Right from the start, Microsoft told customers what to do to avoid getting infected and what to do if they got infected. Microsoft aggressively made sure that everyone had the MS08-067 patch applied. They co-founded an industry working group to study and eradicate Conficker. Microsoft even announced an unprecedented reward of $250,000 for information leading to the malware author’s identification and arrest (although I think that went unawarded). Still, no matter what they did, Conficker continued to spread.

Finally, with a project internally called Backstreet, Microsoft decided to take a more data-driven approach to Conficker. Because Microsoft customers download hundreds of millions of patches each month, Microsoft’s patch install routine had the telemetry ability to check each participating computer for Conficker and also for each of the three possible Conficker vulnerabilities. The data was reported back to Microsoft and the Backstreet project.

To the project leader’s surprise, the USB key autorun method was the most popular attack vector, and not the missing patch as most had previously assumed. It was amazing to some that a method that required more human intervention in order to succeed was infecting more computers than missing patches or weak passwords, which were automated methods that worked over a network.

Given this new data, Microsoft, decided to take a fairly drastic step and disable autorun from running on removable media drives by default. This was a pretty big deal at the time. It was going to remove a decades-old default behavior that users loved for its convenience. This doesn’t happen all that often in computer security, especially when the result risks making hundreds of millions of customers unhappy. Obviously, Microsoft decided that the people upset with the ongoing Conficker disruptions were going to outnumber the people upset about the disabling of autorun. It only takes a few more clicks to run waiting media, but it can take hours to clean a Conficker infection.

In April 2009, Microsoft decided to modify the forthcoming beta version of Microsoft Windows 7 so that autorun was turned off by default on removable media. Microsoft expected that this would make Windows 7 significantly less likely to be infected by Conficker, and after a few months, the collected data confirmed the theory.

Initially, Microsoft didn't want to force older versions of their OS (i.e. Microsoft Windows XP™ and Microsoft Windows Vista™) to have changed functionality, but Conficker was still a huge problem on those platforms. So, in February 2011, Microsoft sent out a new patch to all customers downloading the free monthly Microsoft patches (in the Windows Update program) that disabled the autorun feature when dealing with removable media. Microsoft did publish ways for admins and users to bypass the autorun patch, although the disabling of autoruns would become the default behavior if the user or admin did nothing to stop it.

After the patch downloaded and ran across hundreds of millions of XP and Vista machines, Conficker's infection rate immediately plummeted from many millions per month to maybe a few hundred thousand. From that point forward, not only did Conficker continue to decrease in popularity, but so too did any malware program that used the USB autorun vector (and, due to Conficker's success, there were quite a few malware programs exploiting USB autorun by that time).

I wasn't a part of the Microsoft Backstreet Conficker eradication team, but I knew about it and, along with a few other employees, really took to heart what data analytics could mean to computer security. It was the genesis of what became my Data-Driven Computer Defense philosophy. I figured if such an approach worked across one malware program and all the others like it, why couldn't it work on all defenses? It turns out that it can.

After a few more intermediate experiments, I began my life's work dedicated to a Data-Driven Computer Defense. Wherever I could, I espoused the data-driven computer defense philosophy and implemented it as much as possible. Rarely would a company allow me to do a full implementation. Most of the time I had to prove the data-driven defense concept by starting with more limited POC projects. As the smaller POCs proved successful, I was allowed to do larger and larger projects. Whether they were small or large projects, they all shared one thing in common: If the questions were right and the data was good, they significantly reduced risk. You, too, will likely start with more limited POC examples.

# Data-Driven Examples

The following sections provide some more data-driven defense examples to gain inspiration from or adapt to your needs.

## Mean-Time-to-Detect

Most organizations are happy enough with antivirus reports summarizing how many malware programs were detected and removed by their antivirus software in a given time period. But that metric doesn't mean anything super useful about the increase or decrease in malware risk over time. A detected and removed malware program is no risk to the environment. Accuracy and detection speed are more important.

If an antivirus program detects malware at the same instant that it is trying to enter a computer, that's an example of the antivirus software completely eliminating the risk. The problem is that no stand-alone antivirus software that I'm aware of has a report telling you how long it took from the time the malware program first appeared on the computer until the antivirus program detected and removed it. The antivirus vendor probably sees no benefit to its own interests in providing such information.

The time that it took for the antivirus program to detect the malware program after the malware entered the computer is the most real risk (i.e. mean-time-to-detect). During that time, the malware program can do anything it was designed to do, limited only by the security context the malware program is running under.

> Note: How long a malware program or hacker is actively exploiting something before detection and removal is known as "*dwell time*".

In a very large company, we designed a scheme to figure out the mean-time-to-detect for every malware program across every computer. All the computers already had a running application control whitelisting program, which in this case was Microsoft AppLocker™ in audit-only mode. This meant that it was silently recording any newly installed or executed program that differed from each computer's original baseline configuration. The users of this company didn't even know that AppLocker was running. It simply logged newly appearing programs in the Windows event log and did nothing else.

We decided that every time the antivirus software detected and removed a malware program, we would look to see when the malware program first installed or executed. We did this by always extracting all AppLocker and antivirus log event messages to a database while extracting only the minimum needed log event message details to get an accurate answer.

We started on a limited, proof-of-concept basis and quickly moved it to a full-scale rollout. In a very short time after setting up this data-driven metric, we had lots of useful data. The vast majority of malware was getting detected and removed before it was executed. This was great news. There were, however, times when it took up to three days for the antivirus program to detect and remove particular malware family classes. We were able to take our data to the antivirus vendor and ask why it was taking longer for certain malware classes. The vendor responded, giving us good technical answers for why we were seeing what we were seeing, but at the same time we noticed that even these malware family classes started to be detected faster over time. The vendor had apparently made some changes that decreased the mean-time-to-detect.

The project was a huge success. Not only did it measure the organization's mean-time-to-detect risk over time, and we could see it dropping, but it also helped with other computer security activities.

## Hosts Also Determine Risk

Most vulnerability reports list vulnerabilities as high-, medium-, or low-risk threats regardless of the type of computer they get exploited on. But the computer an exploit lands on has a big impact on the amount of overall risk to the organization. If the same exploit or malware program is on the CEO's computer or a public kiosk computer, that creates different types and amounts of risk. Using a data-driven defense, you can assign different levels of risk to different computers and let that data impact the rest of the risk calculation and defense.

The same organization mentioned in the previous example classified many computers and users as critical risks, meaning that compromises of their computer or software could lead to significantly elevated risk to the organization. The critical-risk systems included critical application software, supporting network devices, infrastructure servers (e.g. DNS, DHCP, etc.), and people who were administrators on critical software systems, payroll and accounts receivable software, as examples. All-in-all, more than 15,000 employees (out of over 200,000) and their computers and devices were classified as being a critical risk. The company also identified over 100 critical "line-of-business" software programs that ran the business.

When a device ranked as a critical risk became infected with malware, if the company's mean-time-to-detect was more than 20 seconds, they sent the operator an automated email explaining the issue and risk, gave details, and asked them to self-report if they felt that they were compromised while processing secrets (e.g. data or logon credential) or viewing confidential data. Although they started with a 20 second baseline, that was eventually changed to 60 seconds and then 120 seconds because of timing issues that created some false-positives and a reevaluation of the real risk in such a short time period.

The idea is that not all computers, users, and devices are the same risk to the organization, and so they should not be treated the same. Malware that goes undetected for days on a database server or the CFO's computer represents more potential risk than if it went undetected on a stand-alone cafeteria computer.

Similarly, if malware was found on some super-critical computers involved in sensitive operations, regardless of the mean-time-to-detect value, the server was automatically cut off from the network and an emergency forensic response investigation was ordered. For example, if a pass-the-hash toolkit was found on a domain controller or the CEO's computer, the organization didn't care if it was removed instantaneously. Its mere presence was enough to initiate a stronger, more immediate response.

All that was needed for these host rating scenarios was a single criticality rating saved to an already existing inventory database for all users, computers, devices, and software. Then when malware was detected and removed, the records from the inventory program were used to indicate host criticality, which then could be used to initiate a particular response. An organization's *identify-detect-respond* processes should reflect that understanding. Does your antivirus detection software and related data help you prioritize risk according to device, software, and user?

## Using Inventory to Calculate Risk

Many organizations I consult with about a data-driven defense love the theory, but they tell me their organization is nowhere near mature enough to begin to implement it. As I've stated in previous chapters, figuring out your biggest risks and their root causes is the beginning point for the rest of the data-driven defense lifecycle. Many organizations tell me that they wish they had some useful data to begin testing data-driven concepts on. I often respond by asking them if they have an accurate software and hardware inventory. I've never met an organization that didn't. With that you can do a lot.

Take every computer that was detected as having malware and compare it to its hardware and software inventory, in aggregate. You want to look for trends that indicate higher or lower exploitation rates for particular hardware or software configurations. Here are some sample questions to answer:

- What hardware and software configurations are exploited the most (percentage-wise)? Which are exploited the least?
- What departments are exploited the most and the least? What are their locations?
- What browser and version are running when most computers are or aren't infected?
- What day of the week or time of day are the most people's devices or software infected?
- What day of the week or time do most people type in the wrong logon information?
- If you have mean-time-to-detect information, what hardware and software computer configurations make the best and worst times?

With just two data sets that likely already exist in your organization, you will be able to point out software and hardware attributes that were more or less prevalent on computers that were compromised versus not compromised, and you can use that data to try to determine risk and additional mitigations.

For example, it’s common to see substantially more exploitations on computers with the following attributes:

- Unpatched software, especially unpatched Internet browser add-on software. But what software? What versions?
- Older operating system versions.
- Older browser versions.
- 32-bit systems (versus 64-bit systems).
- Systems running non-current versions of anti-malware software or none at all.
- Higher exploitation percentages (in particular geographic regions throughout the world).
- Non-domain joined computers.
- User Account Control (UAC) is disabled.

Microsoft frequently reports these types of statistics for global Microsoft customers in their quarterly Security Intelligence Reports (http://www.microsoft.com/sir). The information is usually shown by Windows operating system version and often by country or region.

An organization can compare its own software and hardware inventory to its own rates of exploitation to determine what device traits seem to lead to higher risk. For example, one browser version or another may lead to more exploitations, or perhaps risk can be lowered by moving from 32-bit to 64-bit systems.

Newer, more up-to-date hardware and software usually means fewer exploitations (although not always). If this is true when you’re analyzing your own data and you can put a dollar value on the average exploit recovery event, you might be able to argue, with the data to back you up, that moving everyone to a newer computer or software version will be cheaper in the long run.

With a good inventory and detection of successful exploitations, any organization should be able to determine relative risks for different software and hardware configurations.

## Root Cause Report

A Data-Driven Computer Defense plan includes focusing on root causes of initial exploitations. Minimize root causes, and you kill entire classes of malware and hackers.

One company I worked for understood this better than most. They didn't initially collect or store any root cause data. After my data-driven defense presentation, they did gap analysis and changed many of their processes and tools to better account for root causes.

In some cases, they realized that their software and methodologies were already tracking root causes, and they just had to look for the data. In other cases, they had to buy new software that had the explicit ability to look for and document root causes.

One quick method to generate a lot of root cause data to compare the top malware programs detected by your anti-malware program against the most likely root causes of those malware programs. First, generate a report listing the most popular malware programs detected in your environment and pull those results to a small database or spreadsheet. Next, research the most common ways that those programs can exploit a computer.

Most malware programs only exploit using one or a few predefined exploitation methods. For instance, most malware exploiting across the web comes from "web exploitation kits" (https://www.f-secure.com/en/web/labs_global/exploit-kits), which are predefined to check for and use a few specific exploits. For example, some exploitation kits only look for a few Microsoft vulnerabilities. Others only work with particular browsers and their plug-ins, such as Java, Adobe Acrobat Reader, and Flash. In the database with the top found malware programs, add a cross-tab section that lists all the possible ways the malware could have broken in, if defined. Then you can run a report to list the top root cause methods for the top found malware programs.

The company discussed in this section developed an employee interview process where, after any malware was detected on an employee's computer, they asked the employee how they thought it might have been infected. At first, the company collected data by just emailing or calling the employee and asking the question. This transitioned to an automated email that was sent as part of the detection report process. The interview process was promoted as part of a larger employee awareness campaign. They went from collecting no data on root causes to collecting gobs of it.

From there they were able to create a report that revealed how much overall exploit damage could be avoided by completely fixing a particular root cause. (A simplified example is shown in the table below.)

| Defensive Mitigation | % of Threats Mitigated by Defense |
|---|---|
| Better social engineering training against email phishing attacks | 94% |
| Better patch management of two software programs | 52% |
| Two-factor authentication | 35% |
| Longer and more complex passwords | 2% |

Note: % of Threats Mitigated by Defense will often add up to more than 100%, as several defenses will often mitigate the same threats.

They also calculated how much it would cost them to minimize the root causes, so they could say something similar to “For $98,000 in additional cost to better patch two programs, we can expect to remove $198,000 in resulting exploitation damage.” And so on.

Their goal was to directly identify how much of their current threats would be removed by applying particular and specific mitigations and attach costs to both the problem and the potential remedies. The example above is a simplified representation of a table used in the real production environment, but the concept is the same. When you can collect better data, you can better assign risk and costs to mitigate.

## Data-Driven Security Awareness Training

Security Awareness Training is getting better, although not enough companies are using it or using enough of it, if you look at the data and how often social engineering is involved in doing significant damage to an organization. Companies should review the most damaging social engineering techniques against their company and then create or buy targeted, quality, end-user education (and products) to fight it.

I don’t have a preference of whether the end-user training is created internally or purchased externally as long as it is targeted to the most damaging types of social engineering and is quality education that excels at accomplishing its objectives.

## Working with External Education Companies

External end-user education companies, like KnowBe4 (https://www.knowbe4.com/), offer a wide variety of commercial quality end-user training services. If you select an external company, select products that most closely match your organization’s specific biggest threats. The vendor may already have something available that matches exactly what you need, or you can work with them to create custom, more targeted, courses. Whether your training is made internally or externally, the data-driven key is to use your own local data and experience to determine exactly what type of training should result in the biggest decrease in risk to your organization.

## Simulated “Fake” Email Phishing Campaigns

Email phishing is a huge problem in most organizations. A growing number of companies now regularly conduct simulated (“fake” phishing) email test campaigns against their employees on a regular basis. Test phishing campaigns are a GREAT way to educate the end-user masses about the techniques and risks of email-based phishing attacks. The fake phishing campaigns should match as closely as possible the real-life phishing campaigns that have been most damaging to your company. KnowBe4’s long-term data shows that most organizations can take their users’ “phish prone” rate from about 30% to 2% with a combination of security awareness training and simulated phishing test campaigns (https://info.knowbe4.com/2018-phishing-by-industry-benchmarking-report).

These simulated phishing campaigns are great ways to collect data on which users, groups, and locations are more susceptible to email phishing, and then use that data to provide education specifically to those people. Companies like KnowBe4 make the follow-up education an automatic part of the campaign. Just make sure the education is targeted specifically to your organization's needs. Data-driven defenders should also look for opportunities to see up or down trends for groups of users and try to find out what is or isn't working, and why.

## Spearphishing-Education Example

Social engineering encompasses more than just email phishing. It also includes web site-based phishing attacks, instant messaging spam, tech support scams, physical presence hacks, phone call cons, CEO wire fraud, swatting attacks, and essentially any medium that allows unauthenticated content. A data-driven defense enumerates the top social engineering threats that are currently being the most successful against your organization and the most likely future ones, and it trains employees against those.

One company I worked with had experienced multiple advanced persistent threat (APT) spearphishing attacks that had resulted in the loss of intellectual property and partial reputation damage. They responded by requiring every employee to take 30 minutes of generic social-engineering training each year. But it did not work effectively enough to significantly reduce the success of phishing against the company.

So, they started to test different types of training and compare the success of one type versus another over time. They found out one type of training seemed to have the most success and another type, for reasons that are still unknown, seemed to actually increase the odds of someone falling for a social engineering attack. They got rid of that latter training class.

The training that they kept and pushed out to the rest of the company was a highly personal and customized anti-spearphishing training video (created using an external training company) that shared the true story of a well-known, well-liked, very smart, prominent co-worker who had been successfully spearphished. He narrated the video detailing what he was doing (i.e. working on the weekend) when he received an unexpected email from “a co-worker”. The email referenced a confidential project they were both working on and instructed the employee to click on an encrypted PDF document to see something similar to their project that one of their competitors was working on. The victim wasn’t expecting the email, and it purportedly came from the sender’s personal account, which was not normal, but it referenced information “only the sender” could know. With that, he clicked on the link and accidentally infected his computer.

The respected co-worker even shared how even though he was fairly sure he had just been successfully spearphished, he was embarrassed to call the company’s help desk right away to report the incident. He tried to handle it himself. He didn’t call until the next day. When he called, he was surprised to learn that what had happened to him had also happened to other senior leaders and that an incident response team task force had been set up to help recover and protect the company. He shared that his personal embarrassment and the prospect of possibly getting in trouble delayed his calling the help desk. He closed by saying “If it can happen to me, it can happen to anyone. But you can be smarter than me by not opening a strange external email in the first place, calling the purported sender to verify first, and calling the help desk sooner if you think you’ve been hacked. I regret my slow response, but I’m glad to share my experience here to help others.”

The video was a huge hit. The following year, successful spearphishing attacks plummeted. Another data analysis showed that 60 minutes of anti–social-engineering training, instead of just 30 minutes, helped significantly reduce all social engineering risk. Security awareness training is one of the best ways to see a data-driven defense in action and to be able to see measurable results.

## Tracking Attacker Histories

A new class of computer security defense now exists that can help any organization better track hacker actions and even proactively notify you when hackers or malware have made it into your environment. It does this by monitoring existing data sources or creating new sources and alerting on anomalous activity.

Some of these products/services track known hacker and malware "command-and-control" centers (usually by IP addresses of known malicious hosts or network segments), and if they find traffic heading out of your internal network to a known hacker site, then they alert you. This is sort of an extension of the spam email "blackhole" methods of yesteryear, where email servers sending spam would be marked as such and all the other email servers would simply stop accepting email from them. Only these new services apply to all network traffic and not just email protocols.

Other companies offer services that are very good at telling you not only that they've detected a malicious action, but also when the malicious origination point first entered your system, where it went, and what it did, in detail, since the beginning of its intrusion until the current time. The graphic information you can get and drill down into is incredible. It's a forensic tracker's dream. I could easily write several chapters about these types of services. They are the epitome of a data-driven defense. And they are everywhere.

I was onsite at a customer's facility one day when the CEO reported that they thought he had been successfully spearphished. The company's security operations center (SOC) quickly brought up a virtual representation of his laptop on a huge console screen. They were quickly able to see the dropped malware program, its name, what it did, what other malicious processes it spawned, and where it moved from there. Very quickly, they could see the same malicious processes existed across over 100 computers and that most instances of them had been created in the last few hours. With a few clicks of the mouse, they froze the malicious processes across all machines, halting more immediate damage. It literally took 10 minutes to see the threat, analyze it in detail, and stop it. Data is an amazing thing.

You, too, can take advantage of such systems. Theoretically, any company should be able to create a similar system within their own company. The logs and data necessary to accomplish similar detection and discovery information exist in nearly every company. All the organization has to do is create a system to aggregate, analyze, and alert on the collected data. Unfortunately, this is usually easier said than done, and for that reason, most companies buy an external vendor’s product or service. Either way, whether developed internally or externally, these sorts of malware and hacker detection services and products are great examples of a data-driven defense being used to its best end.

## Reprioritizing Criticality Rankings

Nearly all organizations run vulnerability scans on their hosts to ferret out existing vulnerabilities. One company I worked for ran Tenable Nessus™ for general vulnerability scans, Qualys™ for web site vulnerability scanning, HP Fortify™ for code scanning, and a host of other vulnerability analysis tools. Each server and application in their environment had to be completely scanned by each tool before it went live to their production network, and again monthly after that. Nearly every newly scanned server, application, and web site scanned ended up with a list of dozens of found issues to fix. Many computers had over 50 top priority recommendations found (many were repeats for the same issue found in multiple places), and if you believed their risk ranking alone, they all had to be fixed immediately.

You cannot fix 50 “top priorities” all at once. You can only fix a few at most at the same time. This problem begged the question about which of the top issues were really the top issues.

Using data-driven concepts, we analyzed the top current and most likely successful threats against the company, and then we modified each vulnerability scanning tool so that the real top priorities were vulnerabilities that involved those top successful threats. The number one threat was hardcoded authentication secrets (e.g. passwords, private keys, etc.). So, in this company, if your application was found to have hardcoded secrets, fixing that single issue became the number one problem to fix, followed by everything else.

In phase 1 of this project, we modified the vulnerability tools’ priority rankings by hand. Most vulnerability scanning tools allowed the admin to set custom priorities. We were able to insert our prioritizations into new fields, which we then indexed and reported out in the vulnerability reports. But the process of updating the custom criticalities in the scanning tools and reporting mechanisms was done manually by hand.

In phase 2 of the project, we automated the reranking of the criticalities, so that when the latest data-driven threat/risk dataset came in to reveal the existing and most likely expected threats, the scan tool criticalities were automatically updated. This project was the epitome of a data-driven defense, and I’ve since repeated it many times.

## Driving SDL Requirements

Companies that create software can reduce risk by training their programmers in Security Development Lifecycle (SDL) practices, requiring the use of SDL-enabling tools, and enforcing SDL requirements. For those interested in more detail on SDL, Microsoft has the most free information and tools dealing with SDL of any company: https://www.microsoft.com/en-us/sdl.

Microsoft has long performed data-driven analysis of the issues and practices that cause the highest number of security bugs in the software that it develops. For example, many years ago, a high percentage of bugs was found to be related to older, legacy, long ago unrecommended programming language features/functions that were known to be very exploitable, such as *strcpy*. Since then, programmers have been trained to avoid using known exploitable programing functions, and Microsoft's code development tools and "security safety checks" explicitly look for them. In Microsoft, it's known as the "banned function calls". See https://msdn.microsoft.com/en-us/library/bb288454.aspx for more details.

Unfortunately, checking for banned code functions after the code is programmed is a little late in the development pathway. Microsoft decided to put in earlier code checks so that coding weaknesses, including banned code functions, would be checked for and blocked at the time the developer tries to "check in" the code into the centralized code repository. No longer would a developer have to wait until someone did a code security review to find those flaws. Now, the developer cannot check it in as delivered code if it has a recognized flaw. This not only finds flaws sooner, but helps incentivize and educate developers faster. It took looking for and collecting data around what was involved in most code bugs in order to automate the process that prevents them.

## Individual Personal Behavior Examples

A Data-Driven Computer Defense can be driven at the corporate level, but it can also be done at the individual level, even if not officially recognized by any other corporate structure. The following sections provide some personal real-life examples from my career.

## Training

My boss's boss came up to me and asked if I would get involved in a computer security training program that would cover multiple topics using video training. His exact words were "I know you like to do training and education, and we need someone to lead this effort. Will you do it?"

I looked over the list of five training topics and said "Yes." Then I asked which one should be done first? The five topics were completely unrelated and basically looked like a list of pet projects from different senior managers. He said "They are all important. Do whatever one you want to do first."

This, of course, bugged me. I needed data to back the prioritization. The list of topics was wide ranging from social engineering to cloud security. So, I pushed back and said that I wouldn't begin creating the videos until I had a data-driven understanding of which topic needed to be covered the most first.

The organization already had a fairly good understanding of their top threats, using data, and what came back that could improve the organization the most was improved security awareness training followed by improved Security Development Lifecycle (SDL) training.

Having this data motivated not only me, but the entire team that was making the training videos. Nothing helps motivate a team more than feeling like you're all working toward a common goal that will have the greatest possible impact on the company. It motivates even better than money.

## Stopping Red Team Attacks

I discussed this in Chapter 6, “Asking the Right Questions”, but I’ll address it again. I was meeting with a CSO to discuss the benefits of a data-driven defense. He understood the theory, but he wasn’t so sure it could be implemented in practice. In frustration, he handed me a recent list of the top 20 ways his red team had broken into his organization’s servers. He asked “Which should I do first?” I responded “Exactly!” Without the data we did not know which of the 20 things needed to be done first. But I asked an even better data-driven question: “Which of these things are what the real hackers do the most in your environment?” It’s not enough to simply have data, you have to ask the right questions and get the right data.

In the list of 20 things the red team had done, I saw maybe a handful of attack vectors that were common in the real world. Red teams are good at finding vulnerabilities, but a great red team helps you find and fix the vulnerabilities most likely to be accomplished by real-world threats against your company. Who cares if a red team member, with superior knowledge, creates a new zero-day attack to break into something? I’d rather be told of how they captured the crown jewels of the company using ordinary, very common, most-likely-to-be-used-in-real-life hacker techniques. A data-driven defender cares about the current and most likely future successful local threats more than anything else.

After collecting and reviewing the data, it was great to give the CSO a ranked, prioritized, list of what we needed to fix first, based on current and future most likely attacks. He saw the value and required that all future unranked “fix it” lists be ranked with data before bringing them to him or his team for review.

## Implementing a Thousand Security Controls

One customer wanted to implement every Microsoft security best practice configuration, which at the time numbered over 1,400 separate configuration controls. Most of them were defaults and highly likely to be already implemented. But a few hundred of them were likely to be "net new" in the environment, and it was expected that some far lower number of new controls, say less than two dozen, could be expected to cause some amount of operational interruption.

I was called in to assist with figuring out which controls were likely to cause the most problems, so we could troubleshoot them, try to minimize the pain, and get all the security controls implemented. My first question was "Which security controls will have the most value to the organization in minimizing the most current and future most likely successful threats?" Again, this organization already had a data-driven culture and could answer what the biggest threats were: Unpatched software and social engineering.

I performed an analysis of the 1,400 plus security controls and found only three that would have any bearing whatsoever on the top threats. I found that almost all the rest would provide nearly zero value for the organization in reducing current risk. They either would be pushing out values that were already the confirmed defaults or would have zero impact to the organization's security risk against the real threats they faced. So, instead of helping to push 1,400 controls, I convinced them to push the three controls that would give them the most bang for their buck. And only pushing three controls resulted in less unplanned operational interruption.

Further, I helped them realize that overall, even these security configuration controls would have very little value for their organization if they did not first get their unpatched software and social engineering threats under control. What is the value of pushing out something that will cause operational interruption if its net result is not measurably lower risk? That is what a data-driven mindset thinks about all the time.

The overall idea is that unranked lists of tasks and items should offend you. Replace unranked lists of tasks and items with a list of prioritized ones that are selected using good data that accurately reflects the organization's worst current and future most likely threats. And there are some projects that should not be done in their current form because they don’t really help significantly minimize risk, especially if the larger risks are not already mitigated. You must think with a data-driven mindset all the time. Otherwise, you are more than likely being inefficient with your time and resources and those of your organization. Don’t volunteer to be part of the bad army that was described in Chapter 1.

Chapter 9 described many Data-Driven Computer Defense examples that highlight company- and individual-level responses and projects. Use them to help move your organization to a better Data-Driven Computer Defense plan. Chapter 10 will help you sell your data-driven defense plan and mindset to the rest of the organization.