DRIVING DEVOPS SECURITY

SCALABLE CYBERSECURITY BEST PRACTICES FOR SCALABLE TEAMS

GARRETT GILLAS
MICHAEL WAXMAN

tripwire

DRIVING DEVOPS SECURITY

By Garrett Gillas & Michael Waxman

TABLE OF CONTENTS

1. TEAMS IN TRANSITION 3

DevOps vs. Systems Administrators 3

Why Move to a DevOps Model? 4

The Benefits of a DevOps Model 5

The Five Vital Steps to A Successful Transition 6

2. UNDERSTANDING THE METHODOLOGY 9

Problems You Can Avoid 9

Step 1: Create a Vision 12

Step 2: Start at the Top 12

Step 3: Systems Thinking 13

Step 4: Tools Strategy 14

Step 5: Milestones 15

Step 6: Invest in People 16

Step 7: Collect Metrics 16

Step 8: Embrace Feedback 16

3. ASSESSING THE STACK 18

Ansible 18

Docker 18

Chef 19

GitHub 19

Jenkins 20

JIRA 20

New Relic 21

Tripwire Enterprise 21

SolarWinds 22

Splunk 22

Visual Studio 23

4. IDENTIFYING RISK 25

Inventory and Control of Hardware Assets (Control 1) 25

Inventory and Control of Software Assets (Control 2) 27

Continuous Vulnerability Management (Control 3) 27

Controlled Use of Administrative Privileges (Control 4) 28
Secure Configurations for Hardware and Software on Mobile Devices, Laptops, Workstations and Servers (Control 5) 28
Maintenance, Monitoring, and Analysis of Audit Logs (Control 6) 29
Malware Defenses (Control 8) 30
Limitation and Control of Network Ports, Protocols and Services (Control 9) 31
Data Recovery Capabilities (Control 10) 31
Boundary Defense (Control 12) 32
Data Protection (Control 13) 33
Controlled Access Based on the Need to Know (Control 14) 33
Wireless Access Control (Control 15) 34
Account Monitoring and Control (Control 16) 34
Implement a Security Awareness and Training Program (Control 17) 35
Application Software Security (Control 18) 35
Incident Response and Management (Control 19) 36
Penetration Tests and Red Team Exercises (Control 20) 36
Security Skills Assessment & Filling the Skills Gap 37
5. BEST PRACTICES FIRST, WORST CASE SCENARIOS LAST 38
Foundational Security Controls 38
Continuous Vulnerability Management (Control 3) 39
Maintenance, Monitoring, and Analysis of Audit Logs (Control 6) 40
Advanced Security Controls 41
Data Recovery Capabilities (Control 10) 41
Secure Configuration for Network Devices, such as Firewalls, Routers, and Switches (Control 11) 41
Controlled Access Based on the Need to Know (Control 14) 42
Penetration Tests and Red Team Exercises (Control 20) 44
6. CONTINUOUS IMPROVEMENT 45
Accepting Feedback 45
Reaping the Benefits 46

1. TEAMS IN TRANSITION

The DevOps model is an exciting and relatively new approach to streamline processes towards increasing your teams' collaborative efforts and overall productivity. The word "DevOps" itself comes from combining the "Agile Operations" industry (which involves applying Agile and Lean approaches to operation management) and the broad concept of collaboration between development and operation teams as a whole.

Amazon defines DevOps as "The combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. This speed enables organizations to better serve their customers and compete more effectively in the market."[1]

In a traditional model, the development team will produce code and then give it to the operation team to deploy it. This can lead to confusion and change management challenges ("It worked when I tested it!") that ultimately slows down launches. On the other hand, A DevOps model puts both teams working together, keeping their eyes on customer-facing goals for the speediest and streamlined process possible.

Using the DevOps model promotes a mature and productive working environment, while also improving the end product for the customer.

DevOps vs. Systems Administrators
While DevOps and System Administrators (SysAdmins) often have areas of overlap, they are by no means identical. System administrators are highly focused on implementation and operations while the DevOps teams work on progressive software integration between different internal teams of the company.

The easiest way to understand the differences between these two positions is to look at the definition of their corresponding job roles:

- The job role of a DevOps Engineer is defined like so: “In this role, you'll work collaboratively with software engineering to deploy and operate our systems. Help automate and streamline our operations and processes. Build and maintain tools for deployment, monitoring, and operations. And troubleshoot and resolve issues in our dev, test and production environments.”[2]
- The job role of a System Administrator is defined like so: “System administrators typically install, upgrade and monitor software and hardware… They usually maintain the essentials such as operating systems, business applications, security tools, web-servers, email, PCs, local and wide area networking both hardware and software and mid-range server hardware.”[3]

One major source of confusion behind the incorrect usage of (and dependency on) DevOps engineers is that, about a decade ago, automation began taking tasks away from System Administrators. Automation also made many positions in IT redundant or obsolete, such as testers, developers, database administrators, and so on.

This left System Administrators with less on their plate, leading managers to redirect them so that they now spend more of their time supporting developers, helping end users, and making processes more efficient. Eventually, this led to the DevOps position, and while similarities remain between the two positions, no organization can reasonably expect any one person to have such a broad skill set that they can interchange the two.

They're two separate positions, each with a different purpose. Organizations that understand the difference have a better chance of succeeding.

Why Move to a DevOps Model?

In a traditional waterfall software development model, developers are given a goal and a timeline. As one example, you may be giving your developers a six-month deadline to launch your next product, which has to do A, B and C for your client. In a traditional model, developers will usually use every day of that six-month timeframe to work on their software.

In a DevOps environment, your developers know that they don't have 6 months to code your next product. Rather, they have four months to code, and then two months for the operations team to work with them on testing it, perfecting it, and ensuring the security of the product. The product will be launched by the six-month deadline.

In a traditional environment, the development process will always be drawn out. Projects are divided into phases on a workback schedule and stakeholders tend to collaborate at certain touchpoints. If any single phase of the project goes over schedule, the end product will be delivered late. With traditional waterfall-driven projects the development process will be slow, bumpy, and drawn out.

A DevOps environment streamlines the work process--it's that simple. That's because your developers aren't just working on the code. The operations team is helping them keep the end-user in mind by constantly considering the code in the environment it will ultimately be deployed in.

The Benefits of a DevOps Model

Operations continue to grow increasingly more important as our world moves to be more service-oriented, and that means implementing a DevOps model is a critical move for most software companies. The 2017 State of DevOps report provides insight into this cultural shift and paves the way for further opportunities into this field. "Over the past six years and more than 27,000 State of DevOps survey responses, we've found clear evidence that DevOps practices yield remarkable results for IT teams and organizations… Today, DevOps is viewed as the path to faster delivery of software, greater efficiency, and the ability to pull ahead of the competition."[4]

While each report released so far has continuously confirmed the value of implementing a DevOps model, each report has also brought about new perspectives that companies can greatly benefit from. The most recent report has done just that, delving into topics of automation, version control, integration, and more.

Here's what it could bring to your company:

- **Technical Benefits**: Decreased complexity for management, quicker problem-solving within your organization, and continuous software delivery.
- **Cultural Benefits**: Increased employee engagement, additional and improved professional development opportunities and more productive teams able to enjoy a streamlined workflow.
- **Business Benefits**: Quicker delivery of new features, a stable operating environment, more time for innovation (rather than time spent fixing and maintaining), improved communication, and improved collaboration.

The Five Vital Steps to A Successful Transition
Transitioning to a DevOps model is a multi-step process, but one that your organization can tackle if you take the right approach. The first step is looking at the current state of collaboration at your company. What's working, and what isn't?

1. Figure Out Your Starting Point
Collaboration is already happening in your organization to at least some degree. Before you try to change things for the better, you must consider what's currently being done well. For instance, which groups are already working tightly together?

Focus on what your company is already doing right and look to expand and maximize those approaches to better benefit your business.

2. Create a Plan for the Transition
The best way to come up with a workable plan is to test the transition on a smaller scale. You can do this by locating a small team within your organization receptive to the idea of implementing a new model and moving them to DevOps. Your plan should be largely influenced by what you learn throughout the process of working with the initial team.

Some team members can be resistant to change. Gather input from your most productive team members and shape your plan based on their input. Stakeholders, who will be there to support the process, will play a critical part during the transition. Working with them will be the easy part. The

hardest part about change is convincing other people within your organization to get on board when they may not be too receptive to the change.

As you scale the process and tools up across your organization, you also must remember the importance of training staff on how to properly function within the new process. This means both group training and individual follow-up training for team members that require additional instruction.

3. Make Security a Priority
Security is one of the biggest concerns and investments in organizations today. The majority of organizations realize that a skills gap can lead to security flaws inside of their core systems.

Because traditional environments treat the development process as separate steps (code, test, deploy) there is very little communication between your key teams. Your developers are focusing on coding something that works, and then operations are focused on testing it so that it's ready for the end user.

Even if it's a consideration during the process, security is unlikely to be a priority. The limited collaboration, the arduous processes, and the fast-approaching deadlines will push it further down on the list and out of focus as workers try to complete the end goal. In addition, there's a good chance that your teams don't have a person well-versed in security.

Having this in mind, about 98% of organizations say that they favor integrating security into DevOps, according to Jason Sabin from DigiCert. "An overwhelming majority of companies believe an integrated security and DevOps team makes sense, with 98% of survey respondents saying they are either planning to or have launched such an effort."[5]

This could also prove to be an intuitive move for your organization to ensure that a skill gap doesn't leave you vulnerable.

4. Budget Enough Time
While it's important to always be optimistic and have everything finished in a timely manner, for your company to fully transition to this new

model, you'll need to budget plenty of time for the transition and integration process.

Out of the 150 responses received on the DigiCert survey, respondents estimated the process would take between 7–11 months to complete. In actuality, it took an average of 1–2 years before the companies had completely transitioned to a DevOps-security integrated model. [5]

5. Measure the Progress

As with any change you make in your company, you must use metrics to keep track of your progression towards the end goal: finishing the transition and making work processes more progressive. Plan out a general timeline so you and key stakeholders can revisit it throughout the transition and make sure everything is on track.

It's rare for a timeline to pan out exactly as you planned it, so be open to adapting and re-shaping your plan as you go along to accommodate for new aspects you didn't consider before. There will be bumps along the way. Just stick to it and don't give up, it will pay off.

2. UNDERSTANDING THE METHODOLOGY

Prior to starting the transition process, you'll want to sit down with key stakeholders and propose the new model to them. Hear their feedback on DevOps and how it can have a positive impact on your company and demonstrate how you think it will improve how things are done in your organization. Once you have them on board with the idea, then you can look into the first step of the process.

Everyone needs to be on the same page. This means you and the people who will play a huge role in the transition must understand this transition will cause disruption. You should not expect a smooth or easy transition or a fast one. In the previous chapter we noted, integrations can take between 1–2 years to complete. During this timeframe, your employees must learn new workflow dynamics, new collaboration techniques, and new tools and software.

It's important that you budget for the time needed in each step so the transition is a success. You also need to support your employees through this transition and be patient in solving the problems that will inevitably come up.

Problems You Can Avoid
Problems will occur no matter how perfect you think your plan is, but many common problems can be avoided during the transition. These result from incorrect planning or a poor vision of the process.

Problem 1: Not Determining Success Criteria
Why are you implementing DevOps, anyway? What will it mean for *your* company? If you have no specific end goal in mind, there's no way you can measure progress or call the transition a success. You'll also have a difficult time trying to figure out what needs to be changed during the transition.

To do so, you must have your Development, Operations, and QA team members answer a questionnaire that asks:

- How does your team define DevOps? What's the process like?
- What privileges are your personnel willing to lose?
- What tasks are your personnel willing to delve into?
- What boundaries should there be between the newly developed DevOps system and the rest of the organization?

Problem 2: Forgetting About Culture
"Some, eager to start on the DevOps path, begin by researching which tools they should buy. More important than specific tools, though, are the technical practices that enable you to achieve the very things that most people turn to DevOps for."[4]

In reality, using DevOps tools is only about 25% of the model. The other 75% is about cultural changes that must happen within your organization. If you don't focus on making these cultural changes, the transition process won't do your organization any good. You might be able to train staff on how to use their new tools, but if they still lack communication and collaboration skills on the human side of things, it's going to remain a disjointed and unproductive setup.

Problem 3: Failing to Reorganize Management
There is a reason managers are resistant to change: if it fails, it falls on them. There is little room for error in most organizations today, and managers know that a failed transition could cost them their job, even if it's not their fault. Managers within the delivery pipeline also know something else: Their positions are likely to be removed altogether in a DevOps model.

While they may try to bring up the, "Why change if it works?" paradigm to keep things the way they are, they must realize something very important. Even though the legacy model is working now, it is not as efficient as it could be. And, the modern, fast-paced world will only continue modernizing and speeding up, putting the legacy system further and further behind.

Having employees ask multiple managers for multiple approvals as work goes down the delivery pipeline slows down the process, so these hurdles should be removed. You must determine who's going where, and why.

Make sure that each manager is informed so they can support the process without feeling lost or forgotten.

There is another important point to consider and that is that "…Removing the need for multiple approvals does not mean removing the need for following the security and compliance restrictions. It simply means the DevOps should be responsible for the consequences of their actions and have the power to correct errors if all hell breaks loose."[6]

Prior to starting the transition process, you must reorganize managerial positions.

Problem 4: Not Properly Preparing for Risk
Risk accompanies every major change you make within your organization, and while the risk associated with a DevOps transition can be easily mitigated, you should plan and prepare for it.

"Continuous delivery, automated testing, continuous integration, building immutable infrastructure as code—all of these DevOps benefits lead to drastically shortening the time to market and feedback loops. However, the same actions can lead to significant losses, should the things go awry. Thus said, the DevOps engineers should be trained to remove as many error-generating factors out of the equation as possible."[7]

Typically, the solution is to automate your organization's testing procedures. Testing and staging environments need to be used at every stage of development to catch problems early because letting them get through to later stages of development will lead to significant costs in time and man-hours and even burnout for your teams. Your system must be able to catch issues as early-on as possible.

Problem 5: Failing to Measure
If there are no measurable statistics to compare, no comparison can be made. This is the biggest problem companies face once they get into the transition process. They think they're making progress, but how can they be sure? Every business's end goal is the same: Make a profit.

To determine whether a transition to a DevOps model succeeded, you need to have a way to measure how it affected the bottom line. This

requires that you audit your infrastructure to gain key insights into how much development is costing you and how long it takes to complete. The post-transition infrastructure then must be audited in the same way and compared.

Your organization also must look at improvement over time, so you can present the progress to key stakeholders within the organization.

The Critical Steps for Success
Every organization is different. Each has unique operations with a different-sized team, different output goals, and a different company culture. This will greatly affect how you approach the transition process, and how long it takes to complete. There is no one-size-fits-all guide to the transition because it depends on these factors, however, the simple outline below will help you plan.

It's best to start small. Rather than trying to blindly write a transition plan and then stretch it over your entire organization at once, write one and implement it on the small scale first. Use the mistakes and lessons learned with that implementation to revise and perfect the plan before scaling it up to be full-sized. However, there are eight basic steps.

Step 1: Create a Vision
Your DevOps "vision," or scenario, is vital to the success of the transition. That's because the scenario must include more than just the vision of a perfect DevOps model, it must be a vision of a DevOps model that supports your business's needs. Basically, your vision needs to justify the implementation of a DevOps model into your organization. One of the primary goals of Tripwire customers moving to DevOps is to improve their security. Do you have this goal? Does it justify the change?

If you have no justification, your implementation will be doomed before it even starts.

Step 2: Start at the Top
Key stakeholders *should* require you to present and prove this justification before getting on board, and it's important that you have their support. Even if you believe fully in transitioning to a DevOps

model, it should not be a grassroots campaign. The top positions in your organization need to be behind you on it.

This means the next step in the process of transitioning to DevOps is heading up to the top executives' suite and convincing them on why you need this change. The good news is that their main goal is for the business to work better so profits can keep growing. If you can show them that a DevOps model will help with that, it shouldn't be hard to get them on board.

You need to walk them through it though, because it will cause disruption, and disruption can mean lost efficiency.

Step 3: Systems Thinking

"DevOps often begins with a software development group that has embraced agile discipline. That's a great start because it introduces the group to thinking about creating software as a system rather than simply a process."[8]

Remember that you must begin the transition process on a small scale. Your most receptive staff should be involved in the initial "test run" of the transition so you can take their valuable input and feedback to revise the plan ultimately applied to the entire organization. This team should be ready to use "systems thinking," meaning they understand how everything is part of a system.

"Systems thinking is a management discipline that concerns an understanding of a system by examining the linkages and interactions between the components that comprise the entirety of that defined system."[9]

The systems thinking of this small-scale team also must be expanded to the entire organization. This doesn't mean everyone needs to get on the Agile train, although that would be great. Rather, you must focus on looking at what is currently happening with application development, deployment, and usage. Use that information to figure out how the entire system can be made more effective and less wasteful.

Which Type of Thinking is Happening in Your Organization?

Before moving on, it's important that you solidly understand what systems thinking is. Far too often, it is confused with "holism," the process of "seeing things in the context of their whole."[10]

Picture an assembly line. Each person on the assembly line has one small task to complete. The person at Position A may be laying a mold onto the conveyor belt while the person at Position F may be placing a screw into the hole that Position E drilled. Each person has one task to focus only on. This is a "reductionism" model, which is the practice of: "breaking things down into their constituent parts."

The opposite of Reductionism is Holism, but there's a problem that happens when objectively looking at either model:

"No matter how holistic one gets, there is always a larger whole to act as context, and one is therefore always surgically removing the whole under consideration from the whole it is a part of, which is one of the chief criticisms with reductionism …

Therefore, to break things down into parts (reductionism) is an act of systems thinking as much as seeing things in the context of their whole. A healthy part-whole balance is a necessary aspect of systems thinking."[10]

Successful organizations will find that it is important to distinguish when to use holistic and reductionist thinking. For example, daily standups always involve breaking down tasks into their smallest parts and solving them in a reductionist way. Less frequent sprint planning meetings involve looking at the organization's needs holistically and then deciding which tracks of work to start development on.

Step 4: Tools Strategy
Many tools will be involved in the new DevOps model, and you'll also be employing tools for the transition itself.

Here's a short list of what you will need to select tools for to start with:

- Version Control

- Project Management Tracking
- Build/Deploy Process
- Security Assessment Process
- Change Management

You will also find tools that support your processes, and therefore your systems, as you move through the transition. Don't focus on finding tools from a single vendor, or other small aspects. Doing so will only cause delays and issues in your selection process. Your biggest priority should be: What role will this tool play?

Be certain that your tools are serving the business goals, and not the other way around. That is how you develop a beneficial tools strategy. We'll get into some important tools in the next chapter.

What Is Change Management?
Change Management is technology that monitors and detects changes in files that may indicate a cyberattack. Most changes are intended to make improvements or to correct problems. However, just because a change is scheduled does not mean that it was actually made. Confirmation that a change was correctly applied is critical.

Tripwire's file integrity monitoring empowers IT professionals to enforce change and configuration management policies. This can ensure compliance with internal governance, external regulatory requirements, and industry best practices.

Step 5: Milestones
The transition is disruptive. That's a fact that simply cannot be understated. It's important that you have "short-term wins" or milestones along the way so your organization can continue measuring progress and ensuring that you are headed in the right direction.

People are uncomfortable with change, and uncomfortable people are more likely to panic. That's why you need to do a lot to keep everyone calm, onboard, and receptive to the changes you will be making. Setting milestones and making short-term wins are good ways to do this.

To help keep people comfortable, plan for a small victory every so often in which people can look up and say, “Hey, it’s working!” This is what will keep everyone motivated and confident.

Step 6: Invest in People
Your accounting department might be rather unhappy with the money thrown at investing in new software and paying to rearrange systems. Don’t try to appease them by cutting corners on your employees’ training. The people at your organization are the most vital part of making things work.

Remember, this transition is 75% cultural. It’s 75% *people.* Put the necessary time and money into your employees if you want the transition to be a success.

Step 7: Collect Metrics
Metrics are a vital part of showing executives that the transition is working and that it was a good idea. It's also necessary to use metrics to measure your progress through the transition. But where do you get these metrics from? The tools that you are using for the new DevOps model.

Not every tool you use can generate the metrics you need, but at least some should be able to give you some input on how things have changed since the transition and, hopefully, improved.

You need to audit your current systems so you have something to compare these metrics to once you make changes. One question to ask is: When a feature request is put in, how many days does it take for that feature request to be fulfilled? Ask that same question when you begin making changes and measure the progress.

It’s vital that your comparisons and metrics are accurate. Don’t ask this question to the legacy system about Complex Product A, and then ask the new system this question referring to Simple Product B.

Step 8: Embrace Feedback
Tools will help you continue to gather feedback on the changes happening within your organization, and the effect they are having on

efficiency. However, you also must give employees a way to pass feedback along to you so your organization can continue growing and improving. Lacking such a communication channel or making it too complex to get feedback to you, will only result in disgruntled employees who aren't willing to go through the trouble of trying to inform you about what they're seeing. This leads to "It was better the old way," complaints, and this can then hurt the culture.

And, as you now know, culture is 75% of the change.

So, it's important that you embrace the arrival of feedback continuously, throughout the transition process and beyond. Regularly check-in and ask for feedback if it isn't openly being presented to you. Gather this feedback, sift through it, and make changes accordingly.

3. ASSESSING THE STACK

DevOps engineers are always looking for new tools to improve their work processes. The biggest issue is that there are so many tools to choose from, making it difficult to decide which ones are worth the time and which ones aren't. Here's a look at some of the most common tools specific to DevOps team.

Ansible
"Every business is a digital business. Technology is your innovation engine, and delivering your applications faster helps you win. Historically, that required a lot of manual effort and complicated coordination. But today, there is Ansible—the simple, yet powerful IT automation engine that thousands of companies are using to drive complexity out of their environments and accelerate DevOps initiatives."[11]

If you do any amount of reading about DevOps, it's hard not to come across Ansible. It's one of the most popular tools available because it is surprisingly easy to implement, and it has near limitless potential. However, a couple widely used alternatives to Ansible include Salt and Foreman.

By taking on the repetitive work that your current employees dread, this type of software saves time, money, and raises morale and productivity across the board. It's certainly a tool that's worth looking into.

Docker
"Docker is the company driving the container movement and the only container platform provider to address every application across the hybrid cloud. Today's businesses are under pressure to digitally transform but are constrained by existing applications and infrastructure while rationalizing an increasingly diverse portfolio of clouds, data centers and application architectures. Docker enables true independence between applications and infrastructure and developers and IT ops to unlock their potential and creates a model for better collaboration and innovation."[12]

Modernize your existing apps, hop onto the hybrid cloud, and make continuous integration and deployment simple with Docker's refined technology. Docker uses lightweight containers that can be designed separately and then combined in the tens or hundreds, allowing them to be output as a single application. While Docker is the industry leader in container-based application environments, some teams are using Kubernetes-based orchestrators for their applications. If your organization has anything to do with microservices, Docker should be high on your list of considerations.

Chef

"Our mission is to help the most enduring and transformative companies use Chef to become fast, efficient, and innovative software-driven organizations."[13]

From cloud migration to infrastructure automation, Chef is designed to turn legacy systems into modern, fast-paced, and disruptive new approaches that can take any company to the next level of innovation. Besides removing redundancy, taking care of repetitive tasks, and lowering complexity, Chef promises that implementing the software can decrease failure rate by 50%.

In case that wasn't impressive enough, Chef also works to lower risk by allowing you to deploy compliance audits with a click. Chef includes everything from the building and management stages to collaboration and deployment. The team behind it is confident that Chef will be your "recipe for success". Salt, Fabric, and RunDeck are all viable alternatives.

GitHub

GitHub is already a well-established and highly-favored development tool for a multitude of reasons. Some alternatives include Gitlab and Bitbucket. The reason for using Git-based platforms is that they are flexible, fast, and can adapt to just about anything. It can also be branched out as needed.

GitHub and similar platforms are indispensable tools for DevOps because of its organizational effectiveness and code management capabilities.[14]

Jenkins
Jenkins is central to developers and has a great appeal thanks to its cloud-based design. Jenkins is both a CI (Continuous Integration) and CD (Continuous Delivery) solution.

"The idea of CI is to merge code from individual developers into a project multiple times per day and test continuously to avoid downstream problems. CD takes this a step further to ensure that all merged code is always in a production-ready state."[15]

It's also popular because there are over 1,000 plug-ins that organizations can use to extend the functionality and integrate Jenkins with other software. Since it is so easily pluggable, it is quickly growing to become one of the most popular DevOps solutions on the market. BuildBot and Codeship work in a similar fashion.

JIRA
Created by parent company Atlassian, JIRA is a three-piece lineup of software solutions that can greatly benefit any organization using the DevOps model.

JIRA Software is made for project and issue tracking while JIRA Service Desk is designed as a complete solution for your IT department to handle departmental issues and customer service. JIRA Core is designed for essential business management tasks.

Atlassian claims that JIRA Software is the number one development solution used by Agile teams. It's designed to be used by every member of your development team so everyone can work together while they plan, track, and release great software. The software is perhaps the most intuitive of all, with the ability for you to create user stories and issues, and have your developers work to solve them.

You can also distribute tasks across multiple teams for easy delegation and prioritize as needed. Plan sprints, encourage and enable collaboration, and speed up the time of development by using the Release and Report modules. It comes with out-of-the-box, pre-designed workflows, or you can make your own for your team.

If you're worried about integration, it syncs with over 3,000 apps. You can also check out the alternatives, which include Trello and Asana.

New Relic
Over 16,000 organizations use New Relic's intuitive software. Similar to JIRA, they also have a three-product lineup with New Relic Insights, New Relic APM, and New Relic Infrastructure.

Insights works to put all of your organization's data into one place where team members can organize, visualize, and evaluate it. It offers in-depth analytics for a better understanding of the end-to-end impact software has on your business.

APM, Application Performance Magic, allows your team to build and maintain cutting-edge apps in a single interface. It works with any language and any environment, and it's one of their most popular solutions.

Infrastructure gives you a precise picture of what's going on with your dynamic systems, allowing you to scale quickly and easily and deploy new software intelligently. New Relic Infrastructure aims to help you run your business with maximum optimization and higher visibility. Sentry and Rollbar can achieve similar results.

Tripwire Enterprise
Tripwire® Enterprise is an excellent tool to be integrated into a DevOps model.

"Tripwire provides the most comprehensive file integrity solution for the largest enterprises. Years have been spent honing Tripwire's ability to detect and judge change and prioritize security risks with integrations that provide high value, low volume change alerts. Tripwire delivers a robust file integrity monitoring (FIM) solution, able to monitor detailed system integrity: files, directories, registries, configuration parameters, DLLs, ports, services, protocols, etc. Our enterprise integrations provide granular endpoint intelligence that supports threat detection and policy and audit compliance. With Tripwire, you get the continual assurance of

the integrity of security configurations and complete control of all change to your IT environment."[16]

Tripwire ExpertOps℠ is also a great alternative for this who wish to get expert monitoring and administration in a single subscription. This new SaaS-based offing is ideal for those who don't have enough resources to implement and run their own security controls.

SolarWinds

SolarWinds isn't just popular because they allow companies to try any product for free. It's because once a company tests out their product, it's almost a sure thing that they're going to become a repeat customer. From the Network Performance Monitor to Service & Application Monitor, Database Performance Analyzer, and more, SolarWinds offers a variety of software that can help organizations stay at the cutting-edge of performance.

Their Security & Compliance software, which acts as a log and event manager, is one of their top-performing products. It allows your IT department to quickly spot potential security issues by normalizing log data. It also has out-of-the-box reports and rules for simple implementation and ease of compliance.

But, that's only the tip of the iceberg. All of SolarWinds' software products help your teams solve problems quickly, even before you buy their software. Their "Fix It Now" approach allows you to download anything for free, deploy it, and then pay later. You can solve a problem you're facing in about an hour, which proves the ease of setup, reliability, and usability of their systems.

They have over 250,000 customers around the world, including large corporations like Chevron, Lockheed Martin, Nielsen, and more. They also service government organizations and smaller startups, and scale well to fit the needs of just about any organization. MotaData and Netcrunch offer alternatives to SolarWinds if you're looking at all of your options.

Splunk

Splunk is another highly sought-after software company that's perfect for any DevOps model. Splunk helps organizations turn machine data into

answers in real-time. It's also scalable to meet modern data needs, and it handles complexity with ease. Machine learning helps sort all data you collect and gain insight into the opportunities and risks your business is facing.

“Thousands of organizations rely on Splunk as the single source of truth to help drive better, faster security decisions.”[17] Risk mitigation, incident response, and compliance are all built-in to Splunk's high-velocity security setup. Real-time log management can search, diagnose, and report on security issues while an analytics-driven approach enables your organization to view security information in new ways.

At last count, over 85 of the Fortune 100 companies were using Splunk, including Coca-Cola, AAA, and Zillow. Smaller companies will also find great value in the software thanks to it maximizing IT operations, presenting application performance analytics, enhancing security and collecting business analytics. Logstash and Datadog are two popular alternatives that work in much the same way. When it comes to security, this type of software can help notify you in the event of an issue and help you understand and mitigate security risks.

Visual Studio
Visual Studio, presented by Microsoft, is without a doubt one of the best-in-class tools for developers. With an App Center, team services, and scalable solutions, you may well find your organization favoring Visual Studio over other software.

Real-time collaborative development is made simple thanks to the Visual Studio Live Share. Meanwhile, the App Center allows for continuous integration, delivery, and learning. Visual Studio software is available for both Windows and Mac environments, and developers can create applications for Android, iOS, Mac, Windows, the web, and the cloud-all through one interface.

Some of the highlighted benefits of Visual Studio include the ability to write code quickly, debug with ease, diagnose problems in a snap, test regularly, release with confidence, extend/customize, and collaborate effectively.

The all-inclusive approach that Visual Studio takes enables teams to be more productive, and ultimately ship code out faster, thanks to the “continuous everything” setup that allows for cloud-based development, easy testing on real devices, and simple deployment. Atom and NetBeans are two alternatives to look into.

4. IDENTIFYING RISK

Identifying potential security risks is by far one of the most crucial proactive steps an organization can take to help prevent issues before they even occur. In this chapter, we'll get into mitigating common security risks.

The Center for Internet Security (CIS) maintains a list of the top 20 security controls that every organization should implement, called the CIS Controls™ (previously called the Critical Security Controls, and the SANS Top 20 when they were maintained by SANS Institute)[18]. Some steps are obvious, while others you may not have thought about, but they're all critical to protecting your organization from security issues.

Inventory and Control of Hardware Assets (Control 1)
Your organization must deploy an automated tool that can take inventory of all systems connected to your public and private networks. The intent of this first requirement is to establish a baseline inventory of your organization's assets. NMAP is a good tool to start with to get a quick and accurate account of what is currently running on your network.

"[One] requirement underscores the fact that, without an accurate and precise understanding of assets under control, the rest of what your Information Security Management System could be considered suspect."[19]

In this control, the CIS is emphasizing the importance of using both active tools and passive tools to scan your network and identify hosts by analyzing traffic. This provides redundancy, ensuring maximum effectiveness and helping prevent something from being missed or skipped.

An on-going requirement regarding such a scan will be mentioned in a later requirement on the list. This requirement asks organizations to deploy DHCP Server logging and to utilize a system that helps improve the process of maintaining an asset inventory and detecting unknown systems with DHCP information. If your data center doesn't use DHCP,

this otherwise valuable method probably won't be of much help to you, however, any organization already leveraging DHCP will find using it very useful.

Since most enterprises have DHCP services, and since most DHCP services can generate logs if configured a certain way, it becomes an easy and reliable source of discovery information. If you have it, use it!

As new devices get approved and are connected to the network, equipment acquisitions should automatically update the inventory system.

“This seems like a no-brainer, except for the one pesky word: ‘automatically.' I haven't been in ops for a while, but the last time I was, there was no integration with an authoritative asset inventory and the asset acquisition process. Perhaps things have changed, but if they have, the changes don't seem to have made their way to the security solution space just yet."[19]

This requirement calls for a robust change control process. This process should also be usable to evaluate and approve every new device being added to your infrastructure. While startups have little to no time for security, not to mention change management, it's important that effort is made for robust change management solutions to be implemented.

This one in particular may feel directed towards larger enterprises, but it's important for companies of all sizes. An asset inventory must be maintained that contains the information of all systems connected to the network, along with (at the very least) network addresses, machine names, the purpose of every system, and an asset owner who takes responsibility for each device. The department associated with any given device should also be listed.

Every system on the network with an Internet Protocol (IP) address should be included in the inventory. This includes desktops, laptops, servers, routers, switches, firewalls, printers, storage area networks, multi-homed addresses, Voice Over-IP telephones, virtual addresses, and so on.

In certain instances, this requirement may not apply to your organization. It assumes that your network is an IP network and that it leverages TCP and/or UDP protocols. While this is likely the case, a SCADA system, for example, may be in place instead. Instead of IP routing, an organization using a different system likely uses point-to-point protocol.

"I would still classify a rail switch as an asset, and therefore something that should be kept in the inventory—especially if it is, in some way, routable or reachable by remote means."[19]

Inventory and Control of Software Assets (Control 2)
Implementing this control should coincide with Control 1, covered above. There are multiple similarities between this control and the previous control, and it all starts with devising a list of authorized software. Make a list of software required at your organization for each type of system. This should include servers, workstations, laptops, etc.

Next, you must use file integrity checking software to be certain that the software on the list has not been modified. Following that, your organization must perform regular scans and generate alerts if an unapproved software is installed on any computer within your organization.

You must implement a strict change control process to track and control any changes that may be made to systems on the network. You also need to deploy whitelisting technology that only allows systems to run approved software. It should prevent the execution of any other software not on the whitelist.

A software inventory then must be employed, which tracks the type of operating systems in use for all your equipment. "The software inventory system should track the version of the underlying operating system as well as the applications installed on it."[20]

Continuous Vulnerability Management (Control 3)
The primary requirement of this control is that your organization runs "automated vulnerability scanning tools against all systems on their networks on a weekly or more frequent basis using a SCAP-validated

vulnerability scanner that looks for both code-based vulnerabilities (CVE) and configuration-based vulnerabilities (CCE)."[21]

You very likely need to acquire SCAP-validated scanners if you're like most organizations and don't have them deployed already. When feasible, you should be scanning for vulnerabilities daily. If a vulnerability is identified, remediate it promptly, fixing anything that is critical within a maximum of 48 hours.

Your event logs should be correlated with the information gathered through vulnerability scans. This works to fulfill two goals: First, personnel can verify if the activity of regular vulnerability scanning tools itself is being logged. Second, personnel can correlate attack detection events earlier using the results of your vulnerability scans to determine if the exploit was used against a vulnerable target.

Controlled Use of Administrative Privileges (Control 4)
The first step in implementing this control is to use automated tools to keep an inventory of all administrative accounts. This automated tool should validate each person with administrative privileges on any system, including desktops, laptops, and servers. Each person with these privileges should be authorized by a senior executive.

All administrative passwords must be complex, meaning they contain letters, numbers, and special characters. No dictionary words should be allowed in a password. They also must be of sufficient length, and all administrative-level accounts should have to update passwords regularly depending on the complexity of the password.

All default passwords for applications, operating systems, wireless access points, routers, firewalls, and other systems should be changed to a secure password before being deployed. The passwords for any system should be stored in a well-hashed/encrypted format. Weaker formats should be eliminated.[22]

Secure Configurations for Hardware and Software on Mobile Devices, Laptops, Workstations and Servers (Control 5)
Before you can get this control underway, you're going to have to get started on implementing some portions of Controls 1 and 2. If you were

to only focus on one thing within the CIS Controls, this control should be it. Security Configuration Management, which is what this control focuses on, is vital to your organization.

“Strict configuration management should be followed, building a secure image that is used to build all new systems that are deployed to the enterprise.”[20]

If a system becomes compromised at some point, it should be re-imaged using a secure build. Regular updates also need to be integrated into the organization’s change management process, and images should be created for workstations and servers alike.

There needs to be documented security settings for system images, and these settings must be tested and approved prior to development. These settings should be registered with a central image library, and then validated and refreshed regularly to update the security configuration as new vulnerabilities and attack vectors come about.

“Standardized images should represent hardened versions of the underlying operating system and the applications installed on the system.”[20]

It's important that this hardening removes unnecessary accounts and disables/removes unnecessary services as well. It should also configure non-executable files and assets while involving the application of patches, the closing of unused network ports, and implementing intrusion detection.

When working with Virtual Machines, the master images also must be stored securely themselves, placed on configured servers that are monitored using integrity checking tools and change management. This helps to ensure that only authorized changes can be made to the images.[23]

Maintenance, Monitoring, and Analysis of Audit Logs (Control 6)
“Each organization should include at least two synchronized time sources (i.e., Network Time Protocol or NTP) from which all servers and network

equipment retrieve time information on a regular basis so that timestamps in logs are consistent."[24]

Audit log settings should be validated for each hardware device. Your systems should record logs in standardized forms, like those outlined by the Common Event Expression initiative. Log normalization tools should be used if your systems cannot generate logs in a standardized format.

All systems that store logs should have adequate storage space so your log files won't fill up between log rotation intervals.

Malware Defenses (Control 8)
Automated tools should be deployed to continuously monitor your servers, workstations, mobile devices, and any other endpoints. Alerts should be sent out immediately regarding malware issues. Active, up-to-date, anti-malware protection systems should be in place alongside anti-virus, anti-spyware, host-based IPS functionality, and personal firewalls.

Malware detection should be instantly sent to enterprise event log servers and anti-malware administration tools. Administrators should push updates to machines daily, and after an update has been applied, automated systems should verify that each machine has received the update.

And workstations, laptops, and servers should be configured so they do not auto-run content from USB drives, CDs, DVDs, or other external connections. If any device is not required for business, it should be disabled rather than allowed to continue sitting open or left running.

However, "in most cases, disabling these external sources of media will be difficult, or they will cost you in unanticipated ways … There may be a process optimization in a very secure environment that uses a USB media device, but what would the risk of exfiltration from an insider be under those circumstances? Is the cost savings due to the process optimization worth the potential loss?"[25]

The answer to that question will be specific to both your organization and the risk analysis results.

Limitation and Control of Network Ports, Protocols and Services (Control 9)
Your organization must limit and control network ports, protocols, and services. In doing so, this means you are turning off any service that is not needed for a 30-day period. Following that 30-day period, it should be uninstalled from your system. Additionally, any host-based firewalls or filtering tools must be applied to end-user systems, with a “default deny” rule that will drop all traffic except what is explicitly allowed. An alert should be generated regarding any change not listed on the organizations’ approved baseline. It should then be reviewed carefully.

All services must be kept up-to-date. Unnecessary components must be uninstalled and removed from the system. A service only turned on for limited engagement should be turned off again no longer needed.[26]

Data Recovery Capabilities (Control 10)
Every system must be backed up automatically at least once per week. This should happen more often is systems are storing sensitive information. To ensure the ability of your personnel to rapidly restore a system from a backup, there is certain information that should be included in the backup procedure, including operating system application software, and data from the machine.

Your backup policies should comply with official and regulatory requirements. And data on backup media should be tested regularly by performing a test-run "data restoration" so you know the backup is working. Key personnel should be trained on the backup and restoration protocol and be ready in case a major incident occurs.

In case key personnel are not available in the event of an incident, alternative personnel should be trained. Physical security and encryption should properly protect backups where they are stored, and when being moved across networks.

If you are using cloud-based infrastructure, the backup process will be slightly different but much simpler.[27]

Secure Configurations for Network Devices, such as Firewalls, Routers, and Switches (Control 11)
Compare the security configurations of your firewall, router, and switch against the standard security configuration defined by your organization for each type of network device. Document, review, and approve the security configurations of each device with your change control board.

"All new configuration rules beyond a baseline-hardened configuration that allow traffic to flow through network security devices, such as firewalls and network-based IPS, should be documented and recorded in a configuration management system, with a specific business reason for each change, a specific individual's name responsible for that business need, and an expected duration of the need."[28]

Every device on your network should be managed using two-factor authentication, and encryption during every session where possible. The system must be capable of identifying changes made to routers, switches, firewalls, and other systems on your network. These changes should be logged, including deletions, additions, and changes to any part of the device configuration.

The official master image database should verify any changes made to any system to ensure security. Tripwire's SCM capabilities can help your organization fulfill these requirements.[29]

Boundary Defense (Control 12)
All known malicious IP addresses should be gathered and black-listed so your systems deny communications with them. Access should be limited to trusted sites only, which should be on the whitelist.

"Tests can be periodically carried out by sending packets from bogon source IP addresses (un-routable or otherwise unused IP addresses) into the network to verify that they are not transmitted through network perimeters."[30]

You can locate bogon addresses on the internet for testing purposes. Lists of bogon addresses are publicly available on the Internet from various sources, and indicate a series of IP addresses that should not be

used for legitimate traffic traversing the Internet. This means they can be employed for public use, like testing.

SPF, Sender Policy Framework, should be implemented to lower the chance of spoofed e-mail messages. IDS sensors should also be placed on internet and extranet DMZ systems/networks that scan for unusual attack mechanisms.[31]

Data Protection (Control 13)
Deploying approved hard drive encryption software to any system that holds sensitive data, and to mobile devices, is the first requirement to fulfill when implementing this control.

An automated tool then must be deployed to monitor for sensitive information, keywords, and characteristics that will help you discover unauthorized attempts to exfiltrate data across network boundaries.

This system should also block these transfers and alert security personnel. Periodic scans should be conducted of server machines through automated tools. Data should only be moved using secure, authenticated, and encrypted mechanisms.[32]

Controlled Access Based on the Need to Know (Control 14)
For maximum security, it's important that your organization controls access based on the "need to know" facts. Any sensitive data being stored should be located on separate VLANs, with a proper firewall filter in place. The communication of information over less-trusted networks should be encrypted.

Detailed audit logging should be enforced for access to non-public data, and sensitive data should require special authentication. Host-based data loss prevention, or DLP, should enforce ACLs, even when data is being copied off a server.

Your system should be able to detect any attempt to access sensitive files on a local system along with network-accessible file shares. The system must generate an alert for administrative personnel as soon as possible when an attempt is made by someone without appropriate privileges. The

primary issue here is sample size. Account for the amount of data that these detections will produce, and plan accordingly.[33]

Wireless Access Control (Control 15)
Every wireless device connected to your network should match an authorized security profile and configuration. This profile and configuration should contain a documented owner and a defined business need for the connection. If there is not an authorized configuration or profile, the organization should deny access to the wireless device.

All wireless access points should be managed using enterprise management tools. Network vulnerability scanning tools also must be in place to detect wireless access points. WIDS, Wireless Intrusion Detection Systems, also need to be used to identify rogue wireless devices. They will also help detect attack attempts and notify personnel of successful compromises.[34]

Account Monitoring and Control (Control 16)
All system accounts should be regularly reviewed. Any account not associated with a business process *and* owner should be disabled. Your systems should generate a report each day that includes locked-out accounts, accounts with passwords older than the maximum password age, and disabled accounts. Accounts with passwords that never expire should also be in the report.

This report must be sent to the associated system admin, in a secure fashion.

A process also needs to be put in place for revoking system access. This should involve disabling accounts immediately when needed, such as when an employee is terminated.

Users should also be automatically logged off if they remain inactive for too long. CIS and DISA both provide benchmarks if you are looking for a standard time to use for your system. This should apply to logged in users, and screensavers. Account age should determine dormant accounts. If an account hasn't been used for so long (for example, 45 days) the user or their manager should be notified.

After a longer period, like 60 days, this account should be disabled to prevent breaches. The files associated with the disabled account should be encrypted and moved to a secure server, where they can be analyzed by management personnel.

"What's the purpose of this analysis? From a security perspective, you can see whether this particular user was up to no good. From a management perspective, you can see if they had any critical information that would have otherwise fallen through the cracks. As an employee, by the way, realize that your employer will do this analysis, which is another reason to use organizational machines, not for personal use."[35]

Implement a Security Awareness and Training Program (Control 17)
While the CIS Controls go in-depth into multiple facets that your organization must cover, there are also some additional risks not included in the standards you will need to cover during security implementation. While it may seem overwhelming, implementing security protocols as you transition to a DevOps model is by far the smoothest and most efficient way to go about the change.

"Companies that have begun to integrate security into their DevOps are already seeing a 50 percent decrease in the time spent fixing security issues."[4]

Bug tracking and security testing must be implemented early on if you want the DevOps model to speed up anything at your organization. Automation is one of the biggest pieces of the DevOps puzzle, and it's also essential to your security testing. Manual security configurations simply aren't scalable, so implementing automated security testing will go hand-in-hand with moving to DevOps.[36]

Application Software Security (Control 18)
To protect web applications, you must deploy web application firewalls (WAFs). These WAFs should inspect all traffic flowing into any web application. Your protection must include cross-site scripting attacks, command injection attacks, and directory traversal attacks. Specific application firewalls should be deployed for any application that is not web-based.

“If the traffic is encrypted, the device should either sit behind the encryption or be capable of decrypting the traffic prior to analysis.”[21] If neither solution is appropriate, use a host-based web application firewall.

Explicit error checking should be the minimum for all input. Moreover, the size and type of every variable created in the source code should always be determined. "When the input is provided by the user it should be verified that it does not exceed the size or the data type of the memory location in which it is stored or moved in the future.”[24]

Using automated remote scanners, every in-house developed web application should be tested for common security weaknesses before its code is deployed to production.[37]

Incident Response and Management (Control 19)
To respond to incidents most efficiently, there needs to be written incident response procedures. These procedures should include a definition of personnel roles for handling incidents, and they should also define the phases of incident handling.

You also must assign job titles and duties to specific individuals who will be handling computer/network incidents. Next, define your management personnel who will support the incident response and management. This person will be taking action in key decision-making roles.

Finally, organization-wide standards must be devised for the time required for system admin and other personnel to report events to the incident handling team. The mechanisms for such reporting, and what information must be included, should also be part of the standard.[38]

Penetration Tests and Red Team Exercises (Control 20)
To identify vulnerabilities and attack vectors that could exploit your systems, you must be regularly conducting external and internal penetration tests.

“Penetration testing should occur from outside the network perimeter (i.e., the Internet or wireless frequencies around an organization) as well as from within its boundaries (i.e., on the internal network) to simulate both outsider and insider attacks.”[39]

Carefully control any user or system accounts utilized to perform penetration testing. You must be certain that they are only being used for legitimate purposes. Periodic red team exercises also must be performed to test your organization's readiness to identify and stop attacks.

If you discover a systematic problem, you must track and mitigate those issues.

Security Skills Assessment & Filling the Skills Gap
A security gap analysis should be performed to see which areas of security your current employees are not capable of performing. This should be a basis for creating your awareness program. Devise a periodic assessment for employees and contractors at least annually to make sure they understand information security procedures and policies.

Each person also must understand the role they play in these procedures, and security awareness training should be developed for various personnel job descriptions. This should include specific, incident-based training scenarios to demonstrate potential threats and proven defenses.

5. BEST PRACTICES FIRST, WORST CASE SCENARIOS LAST

While the shift to cloud-based infrastructure has posed new challenges, it has allowed for improved integration and automation opportunities that will ultimately allow your organization to transition to a modern, fast-paced work model.

As you dig into the transition process, keep the implementation of automated security basics at the top of your list. In this chapter, we'll cover those basics. Then, we'll get into the more advanced security controls and how you can implement them.

Foundational Security Controls
From an operational standpoint, implementing many of the CIS Controls may look a lot easier said than done. While it will take significant time and planning, following these controls is essential to a successful and scalable company. Fortunately, you do not need to go at all 20 of them at once.

Rather, you should start with what are considered the "basic" controls that will provide your organization with a solid and secure foundation that you can build upon later.

Inventory and Control of Hardware Assets (Control 1)
Implementing this control will go hand-in-hand with implementing Control 2. The key is to not try and do everything all at once. Start with Controls 1 and 2, and then move forward from there.

Now, before you can implement Control 1 with your organization, you must take these requirements to your vendors. If your tool vendors aren't aware of these requirements, integrating data between your business processes will be a burden due to disparate tools. For that reason, you should look for standard data formats supporting in your tools.

The tools you acquire in the future should also support the standard data formats to ensure compatibility. Start small and start with the basics.

Inventory and Control of Software Assets (Control 2)
As you'll see once you implement Controls 1 and 2, asset inventory is complex and time-consuming, especially when you look at software. Don't try to do everything at once. Going too big too soon will just lead to issues, so start small and work your way out to cover everything.

Again, take the requirements of this control to your vendors and see if they even know of them. If they aren't aware of them, interoperability functionality will be difficult due to non-standardized design. But, it's still possible and equally important that you work to implement this control.

Continuous Vulnerability Management (Control 3)
This control is very focused on the time it takes to accomplish a specific task rather than the number of specific results.

"For example, this Control wants you to measure how quickly you're applying available patches and does not care how many you've applied. Another example, this Control wants you to prioritize application of patches based on vulnerability criticality without concern for how many there might be. This Control, in other words, is all about the *process* of continuous vulnerability management."[21]

Asset management, alerting, and ticketing systems are the three most obvious points of integration. Integration opportunities with LDAP are also present and should be taken advantage of when implementing this control.

Controlled Use of Administrative Privileges (Control 4)
Automation will be your best friend with this control. There is simply no way for you to manually double-check your user base every day, or even once a week. Automate every aspect you can.

The important part of this control is that you don't break the rules for anyone. If employees are having trouble remembering their random 16-character password that they have to change four times a year, use a password manager such as 1Password or Password Safe to mitigate the problems. Don't bend the rules, as they compromise security.

Two-factor authentication is suggested in this control by CIS for administrative users, but consider it for everyone.

Secure Configurations for Hardware and Software on Mobile Devices, Laptops, Workstations and Servers (Control 5)
In the previous chapter, this control was the "If you only do one thing, do this" suggestion. However, remember that Controls 1 and 2 are a prerequisite. These controls will go together as you start implementing. It's suggested that you start with Controls 1 and 2, and then take this control from there.

Look at past years' breach reports to help you find misconfigurations (configuration vulnerabilities) that act as breach enablers. You should also prepare for incidents because this control has a direct relation to your Incident Detection and Response processes, no matter where they're at right now (really great, or almost non-existent).

Again, also take these requirements to your vendors, but also take these requirements to your developers. This control should act as a source of requirements for anything being design in-house.

Maintenance, Monitoring, and Analysis of Audit Logs (Control 6)
For this control, you must enable centralized logging, and review logs regularly. This means more than just opening them, reviewing means using a tool that will help you make sense of the information being shown to you.

You must locate a vendor who offers something that can easily be customized to meet your needs because, once again, this isn't a manual process. Take the time to set this one up right.

Malware Defenses (Control 8)
When implementing this control, it's important that you realize how much of a great friend automation is becoming for your organization. You can have no anti-malware system in place without automation because the tools must be able to *automatically* update signatures, learn behavioral analysis, and interoperate with your other tools.

Your Configuration Assessment tools should also cover your anti-malware systems. Now, when looking at the requirements of this control, you're likely to cause an uproar if you took them at face value. Imagine what your employees would have to say when you consider blocking out personal email, social networks, or instant messaging.

So, start the process with the most common attack vector. This would likely be web traffic or e-mail. Start small and work up. Anti-malware tools cannot catch everything because nothing is perfect, but you can increase security significantly.

Advanced Security Controls
After you have successfully begun the transition and implemented the security basics covered previously, then you can move into the area of more advanced security controls covered here.

Data Recovery Capabilities (Control 10)
Be aware of process dependency when focusing on this control. Your incident response team should be trained in backup and recovery protocols and encrypt things wisely.

"If you're doing encryption on your backups, be sure you're generating keys for the right purpose—long-lived confidentiality and integrity. NIST has some documentation you can read through on cryptographic key management—it's good stuff. The basic thing to remember is that keys are generated for different purposes and not all key generation methods are suitable for all purposes. It's complicated."[8]

Has Word ever crashed on you and you lost an hour's worth of work? Everyone knows the frustration of losing data on a small-scale. That's why this control can absolutely not be understated. Imagine losing your entire system and not having a backup.

Secure Configuration for Network Devices, such as Firewalls, Routers, and Switches (Control 11)
If you aren't already implementing this control, get on it right away. If you can secure the design of your network from the outset, everything else at your organization will be so much easier to manage, and a lot more secure. If you have been operating for a while already without this, you

must take it slow. Begin with qualified humans designing and maintaining a network with security as a top priority.

Set a roadmap, execute, and plan to change. Additionally, get Configuration and Asset Management under control if you want to react quickly.

Data Protection (Control 13)
Once again, start small with this one. With detection, you can do quite a bit without having a DLP (Data Loss Prevention) system in place. While you're likely to need one later, don't implement a DLP system until you first look at your proxies, audit logs, and SIEM solutions. Leverage the tools you already have at your disposal and make sure they're performing at their best.

Controlled Access Based on the Need to Know (Control 14)
When implementing this control, you'll want to start with the obvious and branch out from there. What's most important to your system? Cordon that information off and then take it from there. Data should only be transmitted over a secure channel and appropriate classification identifiers should be tagged to sensitive data.

Everything that happens around or near these "crown jewels" should be audited, and that part goes back to Maintenance, Monitoring, and Analysis of Audit Logs.

Wireless Access Control (Control 15)
Use common sense and be practical when deciding how to treat wireless devices. Many control frameworks, include the CIS Controls, treat wireless as something special. This seems to result from the lingering "newness" of the technology, but it's really just another thing that needs securing.

Wireline and wireless requirements should be married within your organization. Otherwise, you'll be treating the requirements of device authentication and access controls as something separate for these two types of devices. This equates to operating less efficiently.

Account Monitoring and Control (Control 16)
If you do not already have an account management process established, you must begin by establishing one. Accounts should be actively managed, and then you should make a checklist of the things your account management process needs to achieve based on the requirements of this control.

You must pay attention to the on-going monitoring of this requirement. Each time you revisit your credential policies, you should be re-briefed on state of the art attacks like password cracking.

Implement a Security Awareness and Training Program (Control 17)
One key to implementing this control is outsourcing. Look into a security awareness provider who can step in to help you get past your lack of human resources. It's often advisable to not spend your department's valuable time on menial or highly-specific tasks when you can outsource this to someone who specializes in the field.

"This can be *very* boring work, maintaining links between policy and implementation—especially the security awareness piece. Still, this is some of the most important work you can perform. Why? Consider what might happen when "that day" comes and your organization suffers a material breach. The organization is subsequently sued. You land in court and are asked: Did you train your people appropriately—as others in your industry have? Would you be able to answer that well?"[36]

Application Software Security (Control 18)
When implementing this control, a Software Development Lifecycle (SDLC) should be deployed. Security attributes should be attached to the SDLC because it's important that you ensure that the SDLC is performing the right activities and doing so with *qualified* personnel only.

Finally, enlist Quality Assurance personnel to test for security holes when working on this control. Your QA personnel should be trained to handle the basics of security testing. If your organization is inclined, also get a team of security assessors willing to do the heavy lifting.

Incident Response and Management (Control 19)
This control is just as important as Asset and Configuration Management. You need to locate a group of people within your organization who are interested and passionate about information security and get started with this implementation. These people do not all need to come from your IT department.

Some considerations: You might need legal guidance involving any NDAs (Non-Disclosure Agreements) that this team might need to sign since they may have temporary access to unlimited information. You'll also want to seek executive buy-in before you do anything with authority.

Penetration Tests and Red Team Exercises (Control 20)
It's likely that your organization does *not* need to hire a dedicated penetration testing team. You can likely find a reputable group of people at your organization willing and able to do this each day. However, depending on your organization and mission, you might get more for your money by outsourcing this task. Start small with a group of willing people at your organization. If you can't do it on your own, outsource.

6. CONTINUOUS IMPROVEMENT

Security is by no means a "one and done" approach. Even once you have integrated in various automated tools and protocols, you need to continuously be looking at what you have, looking at the industry, and making changes as needed. Over time, your organization will undoubtedly add new employees, new technology, and new systems you will need to account for.

To maintain a secure environment, you must adopt good security standards that will govern everything going forward. Without standards in place, new additions will lack security and interoperability.

Accepting Feedback
Earlier in the book, the importance of accepting feedback was highlighted. Due to its importance in your organization's continued improvement, it's critical that some final tips regarding this topic are covered.

- **How Can People Speak Their Minds?** To truly be willing and able to accept feedback from your staff on anything: systems, protocols, etc., you must have a channel for them to do so. This could mean investing in a tool that allows for easy feedback management, or simply opening your doors to a communicative environment. Communication is one of the cultural changes that must happen when you transition to a DevOps model.

- **Will Their Voices Be Heard?** If you only have a "communication" channel to appease staff, and it's really just a one-way blackhole where the feedback will never be heard or considered, you need to revamp that system. It's important that your employees know that their feedback is being heard and listened to. Again, if employees feel as though they are being ignored or that their opinion is not valued, they will be less receptive to change and have less loyalty to your organization.

- **What Will Be Done?** Measuring and evaluating feedback, especially that coming from people in higher positions and feedback that seems to be a shared consensus, can do a lot to improve your organization's culture, environment, and productivity. When you actually take the time to review the

feedback that staff are taking the time to provide, it can make a significant impact on your organization. Don't underestimate its value. Feedback can help you spot issues within teams, within systems, and even within processes that could have a greater impact on your organization.

- **Give Praise.** Finally, when you have used feedback from your employees to make a change for the better, let them know that their voices have been heard and that you will be doing something about it. This is a surefire way to raise morale, loyalty and improve the collaborative and communicative culture of your work environment.

Reaping the Benefits

The recent adoption of DevOps has been rapid and widespread. It is clear that the transformation has done a lot to help organizations improve their speed and effectiveness at meeting their goals. As cybersecurity risks continue to grow, security best practices must be included in every team's workflow. Regulatory requirements and solid security foundational controls must be a part of the software development lifecycle to ensure scalable success. By understanding and facilitating the cultural shift that DevOps requires, you can help your team work faster and more securely, with sustainable results.

•••

REFERENCES

1. https://aws.amazon.com/devops/what-is-devops/
2. https://insights.dice.com/employer-resource-center/sample-job-description-devops-engineer/
3. http://www.computerweekly.com/photostory/2240205822/How-to-become-a-system-administrator/1/System-administrator-Job-description
4. https://puppet.com/resources/whitepaper/state-of-devops-report
5. https://www.darkreading.com/cloud/98--of-companies-favor-integrating-security-with-devops/d/d-id/1329405?
6. https://techburst.io/transition-to-devops-5-sure-ways-to-fail-b059a83aa0ca
7. https://techburst.io/transition-to-devops-5-sure-ways-to-fail-b059a83aa0ca
8. https://www.infoworld.com/article/3254170/devops/understanding-tools-strategy-in-a-devops-world.html
9. http://www.systemicleadershipinstitute.org/systemic-leadership/theories/basic-principles-of-systems-thinking-as-applied-to-management-and-leadership-2/
10. http://stdaily.ghost.io/systems-thinking-is-not-the-same-as-holistic-thinking/
11. https://www.ansible.com
12. http://docker.com
13. http://chef.io
14. https://github.com
15. https://www.infoworld.com/article/3046038/application-development/why-jenkins-is-becoming-the-engine-of-devops.html
16. https://www.tripwire.com/products/tripwire-enterprise/
17. http://splunk.com
18. https://www.cisecurity.org/controls/
19. https://www.tripwire.com/state-of-security/security-data-protection/20-critical-security-controls-control-1-inventory-of-authorized-and-unauthorized-devices/
20. https://www.tripwire.com/state-of-security/security-data-protection/20-critical-security-controls-control-2-inventory-of-authorized-and-unauthorized-software/
21. https://www.tripwire.com/state-of-security/security-data-protection/20-critical-security-controls-control-3-vulnerability-management/
22. https://www.tripwire.com/state-of-security/security-data-protection/20-critical-security-controls-control-4-controlled-privileges/
23. https://www.tripwire.com/state-of-security/security-data-protection/20-critical-security-controls-control-5-secure-configurations/
24. https://www.tripwire.com/state-of-security/security-data-protection/20-critical-security-controls-control-6-audit-logs/
25. https://www.tripwire.com/state-of-security/security-data-protection/20-critical-security-controls-control-8-malware-defenses/
26. https://www.tripwire.com/state-of-security/security-data-protection/20-critical-security-controls-control-9-limitation-control-network-ports/
27. https://www.tripwire.com/state-of-security/security-data-protection/20-critical-security-controls-control-10-data-recovery/

28. https://www.tripwire.com/state-of-security/it-security-data-protection/security-controls/20-critical-security-controls-control-10-secure-configurations-for-network-devices/
29. https://www.tripwire.com/state-of-security/security-data-protection/20-critical-security-controls-control-11-secure-configuration-network-devices/
30. https://www.tripwire.com/state-of-security/it-security-data-protection/security-controls/20-critical-security-controls-control-13-boundary-defense/
31. https://www.tripwire.com/state-of-security/security-data-protection/20-critical-security-controls-control-12-boundary-defense/
32. https://www.tripwire.com/state-of-security/security-data-protection/20-critical-security-controls-control-13-data-protection/
33. https://www.tripwire.com/state-of-security/security-data-protection/20-critical-security-controls-control-14-controlled-access/
34. https://www.tripwire.com/state-of-security/security-data-protection/20-critical-security-controls-control-15-controlled-access/
35. https://www.tripwire.com/state-of-security/security-data-protection/20-critical-security-controls-control-16-account-monitoring/
36. https://www.tripwire.com/state-of-security/security-data-protection/20-critical-security-controls-control-17-awareness-training/
37. https://www.tripwire.com/state-of-security/security-data-protection/security-controls/20-critical-security-controls-control-18-application-software-security/
38. https://www.tripwire.com/state-of-security/security-data-protection/security-controls/20-critical-security-controls-control-19-incident-response/
39. https://www.tripwire.com/state-of-security/security-data-protection/security-controls/20-critical-security-controls-control-20-penetration-tests-red-team-exercises/

DRIVING DEVOPS SECURITY

Operations have become increasingly important as the software world shifts to a more service-oriented approach. Implementing a DevOps model is an essential move for most software companies to maintain success. The recent adoption of DevOps has been rapid and widespread while security best practices have been slow to keep pace. It is clear that the transformation has helped organizations improve their velocity and improve their products as they grow.

As cybersecurity risks continue to mount, security best practices must be included in every team's workflow. Regulatory requirements and solid security foundational controls must be a part of the software development lifecycle to ensure sustainable success. By understanding and facilitating the cultural shift that DevOps requires, you can help your team work faster and more securely, with scalable results.

Garrett Gillas is a creative technologist and developer from Portland, Oregon. He is currently a MarTech manager in Tripwire's Marketing Ops group. Before coming to Tripwire, Garrett previously managed software development teams at Nike and Razorfish.

Michael Waxman is a Cybersecurity Analyst and InfoSec Manager from Bend, Oregon. At Tripwire, Michael oversees managed services within the ExpertOps group. Before coming to Tripwire, Michael was a security consultant for the Bonneville Power Administration.

tripwire