ISSA JOURNAL

November 2019
Volume 17 Issue 11

Secure DevOps before DevSecOps

Security DevOps

Table of Contents

DEVELOPING AND CONNECTING CYBERSECURITY LEADERS GLOBALLY

Feature

Also in this Issue

The ISSA Journal (1949-0550) is published monthly by
Information Systems Security Association
1964 Gallows Road, Suite 310, Vienna, VA 22182
+1 (703) 382-8205 (local/international)

Hello, ISSA Members and Friends

Candy Alexander, International President

With the end of calendar year within sight, I've got to say that this really has been a year of change. There has been a huge amount of effort in setting the course for our path to a bigger and brighter future. I'll talk more about that next month but having the ISSA International Summit fresh in my memory, I'll share with you some of my thoughts and insight of the event.

First, let me say that you may have noticed that we scaled back on the size of the event, but certainly not the quality. In reflection of our event in Atlanta (2018), it was a conscious decision to not repeat some of the challenges such as overcommitments and co-locating the event with another conference. By scaling back our scope, we were able to focus on our program and content.

Our event began as it usually does with the Chapter Leaders Annual Summit. This year's meeting had 40+ chapters represented, sharing valuable experience and presentations from some of our strongest chapters. I am continuously impressed by the work being done by our chapter leaders and applaud not only their endless hours of hard work but also the wiliness to share their formulas of success with other leaders. Once again, the magic of ISSA comes through—one of the many reasons why I love this association and one of the priceless benefits of belonging.

Attendees of the Summit were treated to some of the best speakers of our profession with the folks such as Winn Schwartau (CVO and founder, The Security Awareness Company, LLC, a subsidiary of KnowBe4, Inc.), Sandra Joyce (Sr. VP, Global Intelligence, Fire Eye), and so many more top-notch ISSA members!

One of the most memorable session of the event was the Fireside Chat, hosted by our media partner ITSP Magazine, featuring Sean Martin chatting with ISSA Hall of Fame member Dr. Ron Ross (Fellow at the National Institute of Standards and Technology) on topics such as his start in cybersecurity and the progression of his career. If you haven't already seen the video recording, be sure to take a look!

As you may have sensed, the refocus of our efforts on content and programming were appreciated by so many members who attended the event, many of whom stopped me in the hallway to comment on their excitement for our new direction. This is something that the International Board of Directors and I are hard at work on: driving the value of membership and the strength of our association through content, programing, and knowledge-sharing opportunities. You will be seeing more emphasis on this in the coming months as we refine the ISSA International strategic goals and business planning efforts for an exciting 2020.

On a completely different note, I would like to share an experience I had the opportunity to have. I had been invited to offer a keynote address to a mid-market CIO event on the topic of information and cybersecurity. Ironically, the challenges that this group of folks face are the same challenges we face, only they don't have the luxury of focusing solely on security—or having a membership-based group like the ISSA to share knowledge and experience. I bring this up for your consideration to reach out to the IT community—invite them along with you to your next chapter meeting or local event.

My apologies for keeping this month's letter so brief. I will make up for it in next month's letter with a reflection of 2019! Until then, my virtual door is always hope!

Candy Alexander, CISSP CISM
ISSA International President
Candy.Alexander@ISSA.org

DEVELOPING AND CONNECTING CYBERSECURITY LEADERS GLOBALLY

ISSA
Information Systems Security Association

Information Systems Security Association
1964 Gallows Road, Suite 310, Vienna, VA 22182
+1 (703) 382-8205 (local/international)

The Information Systems Security Association, Inc. (ISSA)® is a not-for-profit, international organization of information security professionals and practitioners. It provides educational forums, publications and peer inte raction opportunities that enhance the knowledge, skill and professional growth of its members.

With active participation from individuals and chapters all over the world, the ISSA is the largest international, not-for-profit association specifically for security professionals. Members include practitioners at all levels of the security field in a broad range of industries, such as communications, education, healthcare, manufacturing, financial, and government.

The ISSA international board consists of some of the most influential people in the security industry. With an international communications network developed throughout the industry, the ISSA is focused on maintaining its position as the preeminent trusted global information security community.

The primary goal of the ISSA is to promote management practices that will ensure the confidentiality, integrity and availability of information resources. The ISSA facilitates interaction and education to create a more successful environment for global information systems security and for the professionals involved.

ISSA JOURNAL

Now Indexed with EBSCO

Editor: Thom Barrie
editor@issa.org

Advertising: vendor@issa.org

Services Directory

Website
webmaster@issa.org

Chapter Relations
chapter@issa.org

Member Relations
memberservices@issa.org

Executive Director
execdir@issa.org

Advertising and Sponsorships
vendor@issa.org

Moving Phorward into a New Decade

By Randy V. Sabett – ISSA Distinguished Fellow, Northern Virginia Chapter

Just when you thought the phishing epidemic couldn't get any worse, apparently it has (at least for "big game" or high-value targets). Consider this: on October 2, the FBI issued a Public Service Announcement to alert organizations to what are being described as *high-impact* ransomware events. While easy financial gain from a large number of random targets has been a typical pattern, a new wave of phishing attackers are focusing on high-value assets that are often very sensitive to downtime. To do so, attackers may still rely upon phishing as their attack vector.

Not surprisingly, there is no single victim profile. Healthcare providers, manufacturers, cities, and education institutions exemplify recent victims. Attackers often succeed by enticing an unsuspecting user to click on something in a phishing email, which can result in multi-faceted losses. This includes paying the ransom demand, along with lost business, downtime, corrupted files, overtime, third-party remediation, or higher insurance premiums. According to CyberEdge Group, 45 percent of ransomware victims pay the ransom, despite advice to the contrary from various stakeholders (including the FBI). Similarly, RecordedFuture reports that at least 17 percent of state or local government entities pay ransomware demands.

A victim organization often may be hindered without access to daily communication methods or data. An incident response plan that is followed and provides for out-of-band communications can help (also including when the attacker monitors in-band communications). Some steps to consider:

Deep breath

Victims can be intimidated by ransom demands that promise to escalate if the victim delays. Acting too quickly, though, could result in inadvertent loss of information necessary to regain access to and/or restore systems.

Determine status

Identify the extent to which the encryption impacted your specific systems and data, including whether backups exist, the criticality of your data, and the pervasiveness of the attack. Having worked with numerous clients in the situations, the importance of regular and tested backups cannot be overstated.

Coordinate with service providers

Consider retaining outside legal counsel, forensic investigators, crisis communications specialists, and IT support. Outside counsel can help establish attorney-client privilege while also preparing for any necessary notification. Forensic teams help determine how access was gained to the environment and attacker actions.

Notify law enforcement?

The FBI runs the www.ic3.gov website for reporting Internet crimes, with benefits including certain technical assistance or capture of the attacker. On the flip side, the FBI may become a potential obstacle or interfere with the investigation.

Determine whether to pay

Payment considerations include whether (a) it would violate US law prohibiting payments to certain entities or (b) cyber insurance will pay for terrorist acts. Further, the received encryption keys may not work completely, with purposeful design by the attackers for additional ransom demands.

Implement recovery steps

The primary goal is responding to a ransomware event. Doing so, however, could result in the inadvertent destruction of forensic artifacts that can assist in determining the incident's origin and scope, with any attendant potential legal obligations. Evidence must be collected in a forensically sound manner to enable later presentation, if necessary, to a court or regulatory body. Thus, it is critical to proceed simultaneously with respect to recovery and investigation.

Secure your environment

While the primary goal in any ransomware event is to become operational as soon as possible, be sure that your recovery efforts do not leave you susceptible to either re-encryption from the same malware or a new threat. An organization should strongly consider implementing threat-hunting software to determine whether residual threats exist within the environment and proactively complete cyber hygiene efforts.

Comply with legal obligations

A ransomware event that affects the security, confidentiality, availability, or integrity of data could trigger legal obligations to a variety of stakeholders. There are potential statutory or contractual notice obligations to customers, vendors, business partners, employees, or others depending upon the event's facts (that would be informed by a forensic investigation). Non-compliance with these obligations could lead to fines, penalties, or class-action lawsuits.

Ultimately, recovery may be slow, but things usually do improve. Most companies survive and many continue to thrive after a ransomware attack. If you get hit, remember you are the victim and you should be prepared with a compelling narrative as you remediate, recover, and move forward. On that note, I'm off

Continued on page 43

Women Leaders Impacting ISSA

By Curtis C. Campbell – ISSA Senior Member, Chattanooga Chapter

In the coming months, I plan to sit down with female ISSA chapter presidents to gain insight from their leadership experiences, successes, and challenges in leading chapters and developing and connecting a cybersecurity community. The interviews will explore their career journeys and accomplishments along the way and provide a glimpse into their skills and talents that have guided their contributions to the security community.

Preparing for this series highlighting women chapter leaders, I reflected back to ISSA's beginnings. What part have women leaders played within ISSA over the years, and how far have we come? What follows below is a short history lesson on ISSA, its impact globally, and the important part women in ISSA have played.

Early beginnings

In 1984, 35 years ago, ISSA was founded as a not-for-profit, international organization for information security professionals and practitioners. In 35 short years, we have seen technology transformation at lightning speed. In 1984, Dell computers launched and the Macintosh computer aired during the Super Bowl. PCs ran DOS. In the 1990s, Amazon, Yahoo!, and Mosaic Communications (later Netscape) were just beginning. Computers were common but featured floppy disks and dial-up.

By the mid '90s, most people used Windows, and the World Wide Web was up running. In 2014, the Web celebrated its 25th birthday.

In 2019, today's technology development ranges from IoT, AI, autonomous driving vehicles, and virtual reality to 5G, 3D printing, drones, biometrics, and quantum computing.

Throughout this time frame, we have witnessed the huge growth and impact of ISSA, with many women stepping up to a seat at the leadership table. During the formative years, ISSA focused on developing a successful global environment for security professionals with chapters forming all over North and Latin America, Europe, and Asia. Its international vision—to provide members a way to connect and collaborate, expand peer networks, enhance professional stature, and achieve career goals—worked.

Globally, ISSA continues to provide opportunities to develop and connect as respected and highly regarded voices of information security that influence public opinion, government legislation, education, and technology with objective expertise that supports sound decision making.

Women leaders

Early on, ISSA provided a welcoming diversity and inclusion path for women who wanted to get involved and lead the way. Without question, ISSA women have consistently paved roads and volunteered their talents and leadership abilities to promote and further the organization's mission. A good illustration of this is a recent snapshot of ISSA's International Board of Directors. The 2019 elected International Board includes five females in the roles of international board president, vice president, chief financial officer, and two female directors. Thirty-five years ago, ISSA elected a female, Sandra Lambert, as its first international board president to chart the course. In fact, out of 17 ISSA presidents to date, seven of those have been females.

Still work to do

Organizationally, ISSA is an inclusive environment and women in ISSA have narrowed the gender gap in cybersecurity. However, at the chapter level, more women should step into leadership. Currently, ISSA has 130 chapters with over 11,000 security professionals in 92 countries. Yet, ISSA female chapter presidents represent only 13 percent of 96 North American chapters (13 out of 96 chapters). In Canada, one female chapter president represents 20 percent of Canadian chapters (1 out of 5 chapters). In Europe, two female chapter presidents represent 16 percent of chapter leadership (2 out of 12 chapters). Currently, there are no female chapter leaders represented in Asia and Latin America.[1] Globally speaking, roughly 12 percent of ISSA chapters are represented by female presidents.

Regarding the career ladder, statistics show 12 percent of ISSA security professionals rank in top executive-level careers, with 39 percent representing senior-level career status, and 17 percent representing mid-level careers.[2] With 68 percent of ISSA membership ranking in mid-to-top level leadership positions, it would be informative to break down the percentage in terms of women vs. men in future articles.

As we make security a top priority in ISSA, we should continue the momentum for member involvement and growth. Bringing more women into the association and into leadership within ISSA should also be a focus. As sure as the demand for cybersecurity professionals continues with rapid technology development, leadership roles for wom-

Continued on page 43

1 ISSA.org

2 ISSA, https://www.issa.org/issa-media-kit/.

Homomorphic Encryption

By Robert Slade

Recently, a lot of my colleagues in security operations have become very excited about homomorphic encryption. It seems to be the latest "magic" security technology that will solve all our problems, but I don't think we've really provided a good outline of what it is, and, particularly, what it **can't** do.

The basic concept is to be able to encrypt some data, and then still perform some useful calculations, or process the data, without decrypting (and thus exposing) it. (Microsoft seems to want to tie homomorphic encryption to the "cloud," but that's just one possible use.)

IBM seems to be usually credited with the "invention" of homomorphic encryption. Its library, available on GitHub, is an implementation of the Brakerski-Gentry-Vaikuntanathan (BGV) form of homomorphic encryption. Microsoft also has a library on GitHub. Both of these libraries offer addition and multiplication. Google has a somewhat variant process offering comparison and limited addition (and, again, a GitHub library) (I'll come back to those functions.)

The thing is, as a concept, homomorphic encryption isn't new. We've been using it for decades. We've been hashing passwords, and storing them in hashed form. When we want to confirm one, we hash the submission, and then compare that against the stored hash. So we are encrypting data, and using it without decrypting it.

But that's only one, very specific, application. And that's the other thing about homomorphic encryption as a concept. It isn't a single implementation (shades of "blockchain" anyone?). If you want to perform different functions, you have to use different algorithms. With BGV you can add, multiply, and shift. With BFV you can do modular arithmetic. With CKKS you can do addition and multiplication, but you only get approximate results.

As a horrendously simplified example, if you want to sort, you can use the Caesar cipher. It hides the data, but keeps it in (mostly) ordinal position. (Except near wraparounds.) Most modular functions will have somewhat similar traits.

Another not-terribly-useful case would be an exact match search. You can use any block cipher in electronic code book mode, as long as the block size matches the record size. Any identical record is going to match the ciphertext, and therefore you can search without having to decrypt anything.

Microsoft specifically says, of its library, that "other operations, such as encrypted comparison, sorting, or regular expressions, are in most cases not feasible to evaluate on encrypted data using this technology." So, if you want those functions, you've got to invent other algorithms. And it is very unlikely that any algorithm is going to be invented that gives you a whole bunch of functions. When I say algorithm, it's not just the processing part that I'm talking about. It's related to how you encrypt the data in the first place.

In order to do homomorphic encryption, you have to encrypt the data in a certain way that is going to be susceptible to the processing you want to do. And that means you are accepting certain weaknesses in the way you encrypt the data. In my earlier examples, the Caesar cipher is very weak in terms of key address space. Electronic code book is the weakest mode for block ciphers. In the same way, real homomorphic encryption algorithms are going to have weaknesses related to the type of function you want to be able to perform on the encrypted data. (The Google project, for example, notes that their protocol has security against "honest-but-curious" adversaries, but has weaknesses, and provides no protection against malicious input.) The implementations may add protections to defend against those weaknesses. But the more functions and operations you add to the list of what you want to do with the encrypted data, the more weaknesses are going to be built into the algorithm.

Now, homomorphic is a technology to watch. It's developing, and it can be useful for certain applications. But it's not magic. If you have an application, you are going to have to find the right algorithm for it. If you have multiple applications, you may have to use multiple algorithms (which probably means multiple copies of your database, and there are risks associated with that). All the protections that have to be built in to the algorithms are going to mean that homomorphic encryption, like asymmetric encryption, is going to be heavily processing intensive (probably **very** heavily processing intensive).

Pay attention to it. But don't expect it to solve all your security problems.

About the Author

Rob Slade may be an AI experiment gone horribly wrong, and probably encrypted in some weird way. More information than anyone would want about him is available. It is next to impossible to get him to take bio writing seriously, but you can try at rmslade@outlook.com.

The Open Forum is a vehicle for individuals to provide opinions or commentaries on infosec ideas, technologies, strategies, legislation, standards, and other topics of interest to the ISSA community. The views expressed in this column are the author's and do not reflect the position of the ISSA, the ISSA Journal, or the Editorial Advisory Board.

What I Wish STEM Programs Would Get Right

By Jari Peters – ISSA member, Central Florida chapter

There is not a day that goes by that I do not read a post or an article talking about the shortage of cybersecurity talent and how we will suffer in the future due to the massive deficit. I am drawn to this topic because I am a veteran in this field, but also because I have two teenagers who will enter the workforce soon. Despite the fact that they both spend hours per day on video games and/or social media—an embarrassing parental admission—their interest in cybersecurity is non-existent.

My kids are smart and curious, but neither technology nor cybersecurity is on their list of interests. My take on this is that we are not enticing them in a meaningful way.

My tips to get teens interested in ~~STEM~~ *really cool security stuff*:

Make cybersecurity approachable and fun

Video gaming, competing, and socializing with friends is at the center of my son's world. Programs should weave these existing interests into the educational themes. I have seen it work firsthand. On a recent trip to Boston, we found a gaming place called SOGO Action that is taking this in the right direction! Teens walk in off the street and sit in a group with friends and play video games or learn about technology and coding. They were engaged for hours.

Make cybersecurity concrete

As another example, my daughter is an active user of social media. As a cybersecurity leader and as a mother, I worry about the darker side of social media. Why not create a program that will use this interest and encourage teens to consider how their information is handled? Imagine an information scavenger hunt that would test whether teens could guess a friend's location based on information on social media and after their interest is peeked explore how securing individuals' data is needed in all fields, including put into retail, fashion, law, communications, or marketing sites. What could be done if that information fell into the wrong hands? Without even using the words STEM or cyber we can engage teens.

My kids are smart and curious, but neither technology nor cybersecurity is on their list of interests.

Design activities around STEM and cyber

I do believe we are starting to see movement in the right direction. This year I learned about the GenCyber camps from a colleague while attending the Executive Women's Symposium in Washington, DC. The GenCyber Camps are spread across the country and are grant funded by the National Security Agency and the National Science Foundation. My son applied and was accepted through the lottery process!

The camp is approachable and concrete—and even a little cool. His favorite sessions:

- Hearing from a Chicago detective about how he catches bad guys with tech.
- Hearing from an ethical hacker about how he won competitions breaking into things and parlayed that into owning two companies.
- Getting to take the class on a college campus.
- He came home with a Raspberry Pi.

Continue to bring students to the workplace—or send the workplace leaders to them!

We have really smart and interesting people working in corporate America. Perhaps there are meaningful group cyber projects that include employee's teens? Some companies are even investing in public schools and programs to foster strong, innovative, and affordable technology education options. For example, Oracle Education Foundation provided a home for the new Design Tech High School, and Reliaquest became the first national technology partner of 3DE Schools. These are great examples of how we can engage and support the efforts to grow the next generation to join us!

These tips may, in the end, attract teens to cybersecurity by meeting them where they are and developing their interest in cybersecurity based on what already makes them tick. We will be enhancing the skills of an already tech-savvy generation and providing them a better chance of finding a well-suited and potentially lucrative livelihood.

About the Author

Jari Peters, CISM, CIPT, CIPM, ITIL Service Manager, is the Vice President of Security, Risk, and Regulatory Compliance for Oracle's Global Business Units. Jari has 20 years of experience managing technical teams in managed services and cloud, 15 of those years focusing on security, privacy, risk and compliance. She may be reached at jari.mallinder@oracle.com.

The Privacy Problem

By Karen Martin

People have worried that technology is eroding privacy since at least 1890. That's when Samuel Warren and Louis Brandeis wrote "The Right to Privacy" for the *Harvard Law Review*. When Warren and Brandeis warned that "modern devices" afforded abundant opportunities to violate privacy without any participation of the injured party, they were referring to handheld cameras and phonographs. The devices and the rate of technological innovation have changed, but the problem of privacy remains.

Privacy is difficult to define. Suggestions include the right to be let alone, the right of individuals to determine when, how, and to what extent information about them is communicated, or freedom from interference. We have not settled on a definition, even though we can generally agree on whether specific actions, such as hidden cameras in a bathroom, violate privacy. We know privacy when we see it, even if we can't easily define it.

This has not stopped privacy interest groups from pressuring governments at all levels to enact lots of privacy laws. The result is an abundance of similar, but not identical, laws with differing definitions of personal information that must be protected, differing levels of protection, and different criteria to determine whether or not a business is subject to the law. One helpful website has compiled lists of over 75 privacy-related laws sorted by country and region, including 28 state and federal laws in the US alone.

How do you protect something you cannot define? You can start with the obvious problems. Over the years, we identified and sought to protect particularly sensitive data, such as Social Security numbers, payment card numbers, and the Health Insurance Portability and Accountability Act's (HIPAA) 18 personal health information identifiers. Unfortunately, we keep finding new sources and types of data that threaten privacy—cellphone location data or genetic information, for example. In addition, data aggregation and analysis techniques are making it increasingly easy to develop profiles of individuals based on very little information. A quick web search yields numerous companies offering data-appending services that retrieve demographic information including employment, homeowner and marital status, as well as the social interests of customers using just the customers' names and postal codes.

Realistically, it is impossible to exhaustively list every type of information that might be used to violate privacy today, let alone predict what might be exploited in the future. Recent laws, such as the European Union's General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA) have offered broad definitions of personal data: "information relating to an identified or identifiable natural person" or "anything that identifies, relates to, describes, is capable of being associated with, or could be reasonably linked directly or indirectly, with a particular consumer or household." These definitions are not very helpful until regulators or courts provide more specific guidance. Clearly, however, the categories of personal information are increasing. The day may soon come when we will need to encrypt every single piece of information collected from or inferred about data subjects, to keep it secure from unauthorized access as clever data scientists learn to find new ways to correlate seemingly innocent data to recover sensitive information from it.

In addition, with the passage of the GDPR and other privacy laws we must accommodate other rights, which somewhat vary from country to country. Generally, individuals must be informed if their personal data is being collected or processed, must be allowed to opt-out or be required to opt-in, may object to automated processing, must be given access to their data on request, may amend or correct their personal information, and may require that their data be erased. Legislators may assume that those rights will be simple to provide, but if you are reading this, you have a better idea of how time consuming and expensive compliance can be.

There are obviously political, economic, and technical constraints on our ability to protect data subjects' privacy. Governments will always want to access personal data for reasons of public safety or national security. Data subjects will always want more protection of their information than the data collectors and controllers will want to provide. Information security is expensive and difficult; businesses will always weigh costs and benefits, and there will be limits to the amount that even wildly profitable companies are willing to pay to protect personal information. And we know all too well that technical measures to prevent data breaches are never perfect.

We don't really know what privacy is; we're still required to protect it; and any attempt to do this will always be imperfect. That's not the definition of "hard problem" that computer scientists use, but it's definitely hard enough for the rest of us.

About the Author

Karen Martin is a San Jose based Information Security Engineer. She may be reached at kjlmartin@gmail.com.

eLeviathan

By Luther Martin – ISSA member, Silicon Valley Chapter

Encryption is hard to understand, and has been for quite a while. It's not too hard to imagine it as the subject of the lost comedy *Keys* by Aristophanes from the fifth century BC, in which a security manager attends a trade show in search of encryption products for his company. He's taunted by a spectral chorus of erstwhile security vendors chanting "X.509, PKIX, PKIX" as he walks the exhibit hall where he has lots of trouble understanding the technologies that he sees demonstrated because of the incomprehensible vendor pitches. He leaves knowing that his company won't be able to comply with the Greek Data Protection Regulation (GDPR) unless they buy lots of encryption products, but really doesn't learn anything very useful.

But it shouldn't be too hard to understand one of the issues around encryption that has become relatively important today. In particular, many governments are proposing to regulate the use of encryption because its use has resulted in the Internet "going dark" for them, with encryption use by criminals thwarting their ability to perform the surveillance needed to eventually prosecute them.

If law enforcement is losing the ability to prosecute because criminals use encryption, then the political decision is a straightforward one: does the prosecution of criminals justify the potential loss of privacy that weakened encryption might cause? If criminals can freely communicate in ways that law enforcement cannot monitor, then it will be easier for them to carry out their criminal activities. If we regulate the use of encryption—perhaps banning its use outright, or mandating the use of backdoors—there is a chance that governments would abuse the ability to spy on their citizens, perhaps for somewhat shady reasons. The choice of whether to reduce crime at the cost of some privacy is fundamentally about trading freedom for additional protection.

Many people have thought about this very trade-off over the past centuries. It was central to Thomas Hobbes's book *Leviathan*, in which he assumed that the natural state of man is "solitary, poor, nasty, brutish, and short," and that we freely surrender freedoms to a powerful state (represented by his Leviathan, named after the powerful Biblical sea monster) to keep us safe from each other and avoid that undesirable outcome. It's not hard to imagine the need for a modern "eLeviathan" to keep us safe from modern threats, and one thing an eLeviathan might do is regulate the use of encryption. (It might also limit the use of social media, but that's something for another column.)

But making such a trade-off assumes the ability of law enforcement to monitor criminals is actually thwarted by the use of encryption, and it looks like that may not actually be the case. Since 2000, the US Courts annual *Wiretap Report* tracks how many wiretap warrants are issued each year and in how many cases surveillance authorized by a warrant was impeded by the use of encryption. Their data does not seem to support the idea that the Internet is "going dark" for law enforcement. In particular, there are typically a couple of thousand wiretap warrants issued each year, yet only a handful are ineffective because of encryption. And there does not seem to be a trend in which more use of encryption by criminals increasingly blocks the ability of law enforcement to carry out authorized wiretaps. (See "Crypto Wars II" in the January 2017 issue of the *ISSA Journal* for a more detailed discussion of this.)

On the other hand, it sounds like the use of encryption may actually be thwarting the ability of law enforcement to recover evidence from encrypted phones. I haven't seen any authoritative source of information on exactly how much encryption is affecting this, but from the few law enforcement people I've talked to I got the impression that this is indeed significantly affecting their ability to prosecute criminals. But if encrypted phones are the biggest issue law enforcement faces, it seems like a serious error in their communications strategy to call this the Internet "going dark" for them, since that phrasing implies that their problems are related to being unable to get useful information from wiretaps.

So before we create an eLeviathan that will regulate our use of encryption, it's probably best to ensure that the beast is really needed. Law enforcement may have a legitimate concern about being unable to get data from encrypted phones, but it certainly looks like the Internet is not "going dark" for them. Let's not create the eLeviathan quite yet.

About the Author

Luther Martin is a Distinguished Technologist at Micro Focus. You can reach him at luther.martin@microfocus.com.

The Open Forum is a vehicle for individuals to provide opinions or commentaries on infosec ideas, technologies, strategies, legislation, standards, and other topics of interest to the ISSA community. The views expressed in this column are the author's and do not reflect the position of the ISSA, the ISSA Journal, or the Editorial Advisory Board.

DevOps and Infosec

By Jason Remillard – ISSA member, Raleigh Chapter

Adopting the development methodologies in your own security programs, contributing to and fitting into existing release plans, and breaking up your larger security release requirements into the realities of real operating DevOps WILL greatly increase the success of your own security program!

Our shop is probably not unlike yours. We have a distributed workforce mixed with staff, contractors, BYOD, fixed assets—cloud and on prem—COTS, and homegrown services with a mixture of public and private cloud infrastructure.

We also have the added challenge of delivering hosted offerings for clients that include sensitive data and service components that need to be highly available and have slipstream feature enhancements to a global audience. We deliver a global privacy SaaS offering that has many iterative enhancements—security and otherwise—that have a tight release and update cycle. Coupled with the privacy offering are other attached services that are also not only sensitive but also must have a quick time to response to events—regular and unscheduled.

The infosec impacts on DevOps are pretty significant, more so in a highly integrated service with cascading or third party-based infrastructure or services. For many offerings, it could be as simple as our live chat or ticketing system not being available (during an outage or not) being a compounding factor affecting a service component.

Ask anyone in operations. You can't delegate or abdicate responsibility, and certainly you can't do this in the infosec realm. Has the infosec conversation changed in the DevOps realm in today's rapidly changing service delivery frameworks that are growing more popular today? We all know that "baking" security into the development process in the beginning is always a more efficient and effective process, especially in the "waterfall" days. But in today's world of mixed development, deployment, consumption, and delivery platform, it's a mixed bag.

More concerning is a variability of developer skill sets. With today's "click to push live" abilities and potential exposure to millions of end users in seconds, the risks are larger than ever.

So, what do we do?

Rather than spend time on a laundry list of tools and feature-functions reviews, perhaps a better use of our time is a classic "what happens when" exercise.

Our training tells us to expect something to happen, and plan for it. Simple.

We cannot portend to know every single threat or vulnerability—internal, external, known or created—today or years down the road. It also behooves us to understand the nomenclature of our colleagues in development/operations much more closely. For too many years infosec has been in its own regimented area spouting policy, frameworks, and methodologies without much consideration for the general realities of the overall business. The world is different now, and we all need to get along much more now. Certainly the requirements for security, policy, regulation, delivery, development, product, and indeed the business as a whole are much tighter and integrated.

Hardly newsworthy, it should be reinforced that infosec needs to leverage existing channels that the business and DevOps leverage for emergency and urgent upgrades and enhancements as needed. There are many examples today we can all leverage to ensure that when the "when it happens" scenario happens, there is a path to execute and develop that urgent security patch, remove that nefarious library, or disconnect that third-party vendor causing the issue.

These pathways should be tested just like any DR plan, and perhaps more importantly the appropriate delegated teams and processes should also be enabled to manage these same urgent functions.

Many of our systems are highly integrated with myriad other platforms distributed deeply across so many others it's usually hard to keep track of, in many facets, operationally, contractually, or technologically. Sometimes the "when it happens" won't be your organization – it will be someone else's.

Another often overlooked consideration is to ensure that your ability to receive notifications of issues from your service providers for security issues is open, monitored, and kept up to date. This communications channel is critical and should be tested regularly.

Quick tips:

Get on that backlog!

- Tech debt is your friend. The dev team will love you if you support their goals to help get their "technical deficits" resolved. Their tech debt is your opportunity to clean up your issues as well.
- Your team should have a regular slot in each epic (a major theme or story line for a release) at least, maybe every sprint (a small block of time/effort—usually many sprints are in a release or epic) or two – it's a great chance to get some items knocked in a regular time slot.

Continued on page 43

News That You Can Use…

Compiled by Kris Tanaka — ISSA member, Portland Chapter

ESG Survey Sees Long DevSecOps Road Ahead

https://devops.com/esg-survey-sees-long-devsecops-road-ahead/

Survey says…The adoption of DevSecOps is still uneven at best. Only 33 percent of the organizations polled reported that they are involving cybersecurity teams at the start of the application development process.

DevOps Now Most Sought-After Skill, and with Good Reason

https://www.zdnet.com/article/devops-now-most-sought-after-skill-survey-finds/

If you have DevOps skills, this is your time. Technology practitioners with these talents are more in demand than any other IT-related skill, even surpassing cloud and data science skills. As modern IT operations teams overhaul their traditional practices to generate greater innovation and improved agility, they are bringing on board those who can embrace DevOps practices such as agile planning, configuration management, continuous integration, and automated deployments.

Transitioning to DevOps: The Four Priorities

https://www.developer-tech.com/news/2019/nov/01/transitioning-devops-four-priorities/

It can be a little confusing. That's why when you begin your DevOps transition journey, you need to start with a strong foundation—setting goals. If you don't have a road map that clarifies what the business is trying to achieve, everything else that follows has the potential of going way off track.

DISA Looks to Expand DevSecOps to Its Infrastructure

https://www.fedscoop.com/devsecops-disa-infrastructure-fiscal-2020/

The US Defense Information Systems Agency will be focusing on 10 emergency technology areas in 2020. What's at the top of the list? DevSecOps.

Celebrate National Cybersecurity Awareness Month with CCPA FAQs!

https://www.natlawreview.com/article/celebrate-national-cybersecurity-awareness-month-ccpa-faqs

How did you celebrate National Cybersecurity Awareness Month last month? Did you download the toolkit? Did you play the 2019 Trivia Game? Or did you follow the lead of The National Law Review and brush up on your California Consumer Privacy Act (CCPA) information. January 1 will be here before we know it. Are you ready? Read more about NCSAM, the toolkit, the game and this year's theme, "Own IT. Secure IT. Protect IT." — https://www.dhs.gov/national-cyber-security-awareness-month

US and Taiwan Hold First Joint Cyber-War Exercise

https://www.bbc.com/news/technology-50289974

As war moves from the physical battlefield into cyberspace, new forms of cyber-cooperation are forged between nations. For the first time, the United States and Taiwan join forces in a week-long simulated cyber-war event, designed to shed new light on cyber attacks and threats.

Cyber Threats Set to Increase In 2020

https://www.techradar.com/news/cyber-threats-set-to-increase-in-2020

It's a little early. But just like the Christmas decorations that started to appear in stores in September, we are now starting to see the first signs of cybersecurity prediction reports for 2020. Not ready to take out your crystal ball yet? No worries — you still have time.

How to Safely Shop Online This Holiday Season

https://www.govtech.com/blogs/lohrmann-on-cybersecurity/how-to-safely-shop-online-this-holiday-season.html

Ready…set…shop! Did you know that Black Friday is now a global event? And Cyber Monday is expected to pull in $9.4 billion this year, a $1.5 billion increase from last year. As you begin to make plans to jump into the holiday sales frenzy, remember that cybercriminals are also making plans to gain access to your accounts by taking advantage of distractions during this annual epic increase in shopping activity.

DevSecOps Model Requires Security Get Out of Its Comfort Zone

https://searchsecurity.techtarget.com/feature/DevSecOps-model-requires-security-get-out-of-its-comfort-zone

For a long time, it's been a love-hate relationship between developers and the security community. But thanks to a rapidly shifting threat landscape, application security is now more of a priority than an afterthought. What is the DevSecOps culture like in your organization? Security in the News would like to know if you have already embraced the DevSecOps model. Was it an easy transition? What challenges did you face? Share your thoughts with Editor Thom Barrie.

ISSA Strategic Partner: ITSP MAGAZINE

At the Intersection of Technology, Cybersecurity, and Society.

An InfoSec Life | A Fireside Chat with NIST Fellow Ron Ross during ISSA International Summit 2019

By Sean Martin

In this fireside chat, Sean Martin discusses the profession of cybersecurity, past, present, and future with Dr. Ron Ross from the National Institute of Standards and Technology.

The interview covers a myriad of topics, including Dr. Ross' cybersecurity career with the Department of Defense, the Intelligence Community, and NIST; a retrospective of the key projects and cybersecurity initiatives he had led during his forty-five years of public service; mentoring the next generation of cybersecurity professionals; lessons learned; and key cybersecurity and privacy challenges and opportunities for the future.

Webcast & Podcast: Click Here.

An InfoSec Life | A Conversation with Vandana Verma

By Marco Ciappelli

All right, ladies and gentlemen, it finally happened: Vandana Verma is my distinguished guest on this An InfoSec Life podcast.

For those of you that haven't had the pleasure to meet her in person—Sean and I had this honor in Las Vegas this year—let me tell you, she is as nice as she sounds. In my opinion, there is nothing more valuable to add to someone's professionalism and skills than a big heart. It helps to make them a role model and an inspiration for any just entering—or that are about to enter—their career in the infosec community.

Podcast: Click Here.

On Disability, Technology, and Flourishing | A Conversation with Joel M. Reynolds

By Marco Ciappelli & Sean Martin

People with disabilities are experts at navigating a world that is not built for them—often by turning to technologies such as voice recognition devices and cochlear implants. But which technologies, and under what circumstances, truly enhance a person's ability to live the most meaningful, flourishing life? And which technologies, and in what cases, have the opposite effect?

The project we talk about on this podcast is funded by the National Endowment for the Humanities and will explore how technologies can be used to promote or thwart flourishing through conversations with people with disabilities.

Podcast: Click Here.

MITRE ATT&CK—This Is Not Just Another Framework | A Conversation at the Edge with Katie Nickels, Fred Wilmot, and Ryan Kovar

By Sean Martin

It took me a while to get the conversation with Katie Nickels and Fred Wilmot sorted so we could talk about all things MITRE ATT&CK. Fortunately, we found some time together in person in Las Vegas during Hacker Summer Camp. As a bonus, I also got the chance to meet Ryan Kovar who happened to be presenting on ATT&CK with Katie that same week. Ryan joined us for the conversation as well.

Have a listen as we explore what MITRE ATT&CK is, what it's for, who it's for, how to get started with it, how to be successful with it, and what scenarios could be leveraged to learn from others' successes and challenges.

Podcast: Click Here.

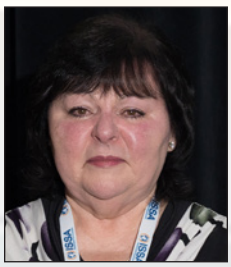
At the International Summit in Dallas

International Awards

Hall of Fame

Winn Schwartau

Honor Roll



Kelley Archer
Linda Archer Receiving

Honor Roll

Shawn Murray

Outstanding Organization

Girls Go CyberStart
Mandy Galantee

Security Professional

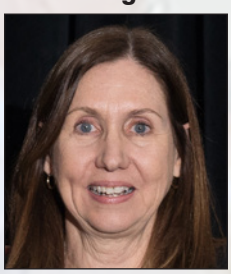
Arthur Cooper

Chapters of the Year

International

Paulus Cocu

Large



Ann Marie Colombo

Medium

Betty Burke

Small

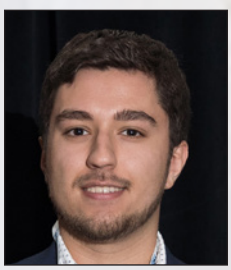
Curtis Campbell

Parting Gift Outgoing Board Member

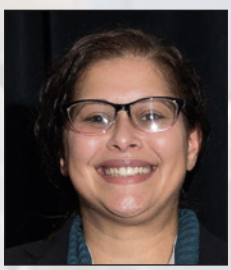
Roy Wilkinson

Volunteers of the Year

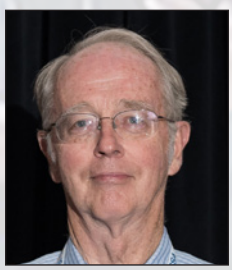
Don Creamer



Dylan Foos



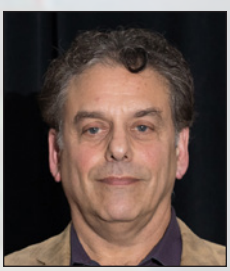
Nia Luckey



Warren Pearce

ISSA Fellows

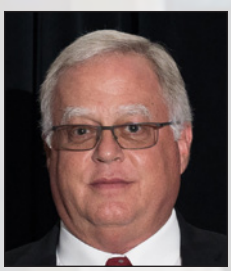
Distinguished Fellow



Richard Greenburg

Fellows

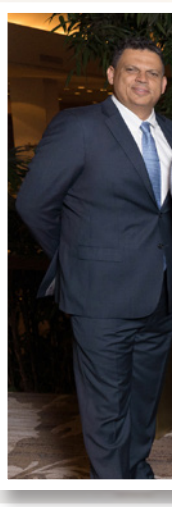
Edward Frye

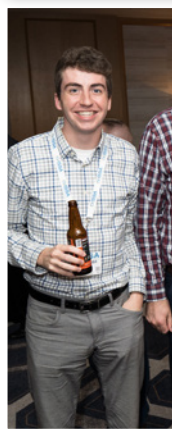


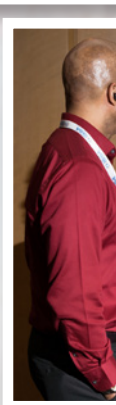
Howard Gordon

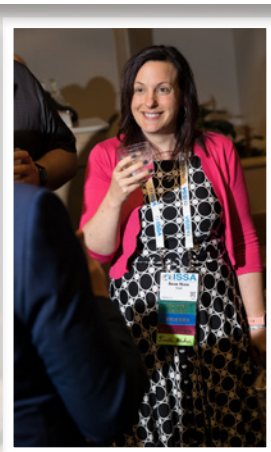
Jorge Orchilles

Michael Stanton









Reader Response

Re: Immaturity and Moral Hazard in the Cyber Insurance Market

By Kevin A. Sesock

ISSA Journal October 2019

Hi Kevin,

Thank you for the article on cyberinsurance in the latest *ISS Journal* – the best piece I've read so far on this topic. Very nicely written! Your comments on moral hazard were particularly thought-provoking.

You didn't mention ISO/IEC 27102, a new ISO27k standard on cyberinsurance. Personally, not being an insurance specialist, I find it a useful source of general information/guidance. I've reached similar conclusions to you about the need for both parties (insurers and insured) to tread cautiously in this evolving area of practice. I might even go so far as to say that the insurance industry's credibility is on the line here: if too many insurance claims end up in disputes, court cases, and reduced or nil payouts, the market will surely crumble.

Kind regards,

Gary Hinson
ISSA member, New Zealand Chapter

Hi Gary,

Thanks for the kind words! I actually wasn't aware of ISO27102—looking forward to the additional research!

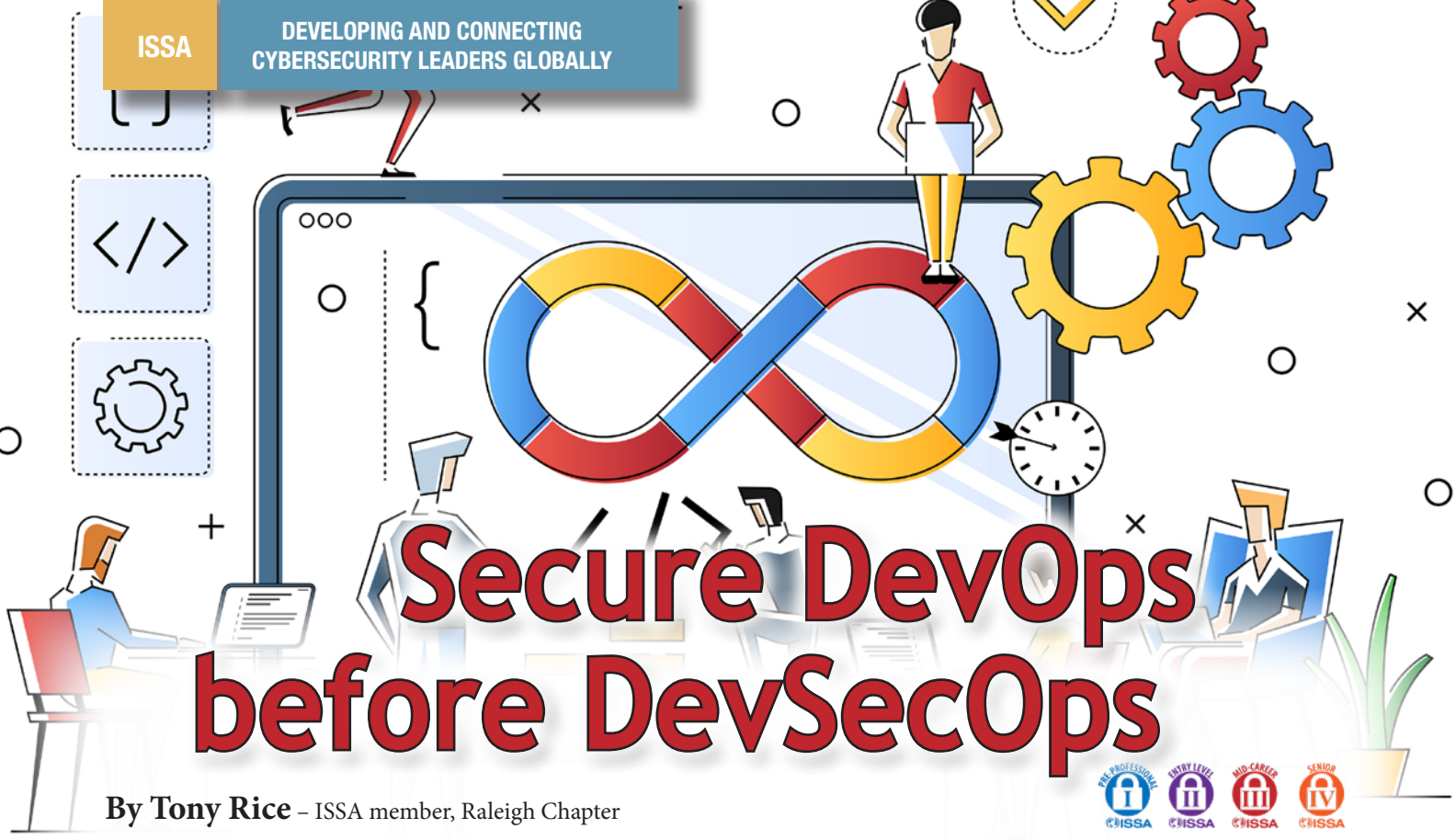
In the meantime, your concerns about credibility are insightful, and I wonder if the high-profile claims that have caused insurers to exit the market may be to blame. Some insurers would rather cut and run as opposed to be caught up in their own reputation's risk.

Thanks again for the feedback!

Kevin Sesock
ISSA member, Oklahoma City Chapter

The Open Forum

The Open Forum is a vehicle for individuals to provide opinions or commentaries on infosec ideas, technologies, strategies, legislation, standards, and other topics of interest to the ISSA community. Articles should be 700-800 words and include a short bio and photo. Please submit to editor@issa.org.



Secure DevOps before DevSecOps

By Tony Rice – ISSA member, Raleigh Chapter

This article discusses the opportunities DevSecOps offers to stand up infrastructure in a consistent secure way as well as move discovery of security flaws earlier and more often in the software development life cycle, with a back-to-basics view of securing access to these resources first.

Abstract

DevOps offers a world of possibilities to automate security checks to find vulnerabilities earlier, before they are deployed and when they are least expensive to fix. It also offers opportunities to more closely partner with developers and operations personnel. But before we turn DevOps into DevSecOps, the build pipelines enabling this automation must be locked down to prevent information about the vulnerabilities found from being leaked the way credentials are leaking today.

When security professionals hear the term "DevOps," a rainbow of automated security checks come to mind. Visions of vulnerabilities revealed early and often dance in their heads, quickly morphing DevOps into DevSecOps.

We dream of all that agile development methodologies offer to more closely partner with developers and operations:

- Helping developers stop the flow of OWASP Top 10[1] examples into the codebase.
- Guiding operations personnel to the consistency that infrastructure-as-code provides.
- Putting an end to the vulnerability whack-a-mole that artisanal handcrafted (and never again patched) instances bring.

Tool vendors stoke these fires with talk of how easily their tools with prepackaged test suites integrate into existing pipelines. It's even easier if you migrate everything over to their flavor of DevOps. And the dashboards, so many dashboards, each providing evidence of the progress being made to leadership eager to know how the investment is paying off.

Can DevSecOps finally fulfill the promise of moving vulnerability discovery to the left of development process where it is cheaper and easier? As time between deliveries shrinks from months to hours, DevSecOps must enable these things and more.

Security professional, heal thyself

Before rushing off to integrate the dynamic application security test suite and code scanner—the ones that cost so much yet get used so little—into a maturing DevOps pipeline, consider the threat landscape that DevOps itself introduces.

Functional and other tests in a DevOps pipeline describe where the product doesn't work. Security tests describe where the product is vulnerable. This is sensitive information that

1 OWASP Top 10, 2017 https://www.owasp.org/index.php/Top_10-2017_Top_10.

should be generated, stored, and transmitted in an appropriately secured environment.

Carelessly implemented Dev[Sec]Ops can broaden an already vulnerable attack surface. Your DevOps pipelines weren't built in a day and neither should your DevSecOps. So where to start?

Begin by securing the DevOps pipeline itself, using the same principals you've been preaching to the development and operation teams. A basic continuous integration/continuous deployment (CI/CD) pipeline is made of segments implemented by individual tools, each with its own set of credentials:

- **Source code repository**: to track changes as the development team moves the product forward such as Git, Bitbucket, Subversion, SourceForge, CodeCommit, Cloud Source, etc.[2]
- **Artifact repository**: to store the fruits of those builds such as Artifactory, Archiva, Nexus, etc.[3]
- **Deployment host**: somewhere to deploy the product such as cloud compute services like Amazon Web Services (AWS) EC2, Azure Virtual Machine, or Google Cloud Platform (GCP) Compute, or maybe serverless services like AWS Lambda, Azure Functions, or GCP FAAS; containers; or an even an on-premise host.
- **Automation platform**: to keep things moving through the pipeline when they should and stop their progress when they shouldn't such as Jenkins, Bamboo, CircleCI, GitLab, CodeBuild, etc.[4]

2 Neil Chue Hong, "Choosing a Repository for Your Software Project," Software Sustainability Institute, https://www.software.ac.uk/choosing-repository-your-software-project.

3 Carlos Sanchez, "Using Repository Managers," DZone Refcard #181, https://dzone.com/refcardz/binary-repository-management.

4 R. Vaasanthi, et al, "Comparative Study of DevOps Build Automation Tools," International Journal of Computer Applications, July 2017, https://www.ijcaonline.org/archives/volume170/number7/vaasanthi-2017-ijca-914908.pdf.

Many environments also include:

- **Communication tools** such as Slack, HipChat, WebEx Teams, etc. that enable developer collaboration and bots to ensure visibility of DevOps results.
- **Workflow and defect management tools** like Jira, Trello, Bugzilla, etc.

Begin by looking at the couplings along that pipeline of tools. Are they secure or is your hard work leaking into the hands of the bad guys? An automation platform like Jenkins is the grand central station of this system and a good place to start securing CI/CD efforts.

If your CI/CD pipeline grew out of a developer experiment that became a critical production resource over time, you might find developer personal userids and passwords throughout its configuration. This can lead to failing pipelines when that person changes the password. It is tough to remember all the places those credentials have been used. You might find that one developer account across each of those tools in the pipeline.

Those person-to-machine credentials should be replaced with machine-to-machine, rotated on a regular basis. Also use access tokens rather than userid and passwords whenever possible. If your tool of choice doesn't support access tokens, that's a good sign to keep looking at other tools.

However, as CI/CD pipelines proliferate, and they will, you'll quickly find that managing all those secrets will become cumbersome and risky.

Stop checking in credentials

You need look no further than files publicly available via GitHub.com for one of the most common problems in DevOps: storing configuration files and source code containing secrets in plain text, in a source-code repository.

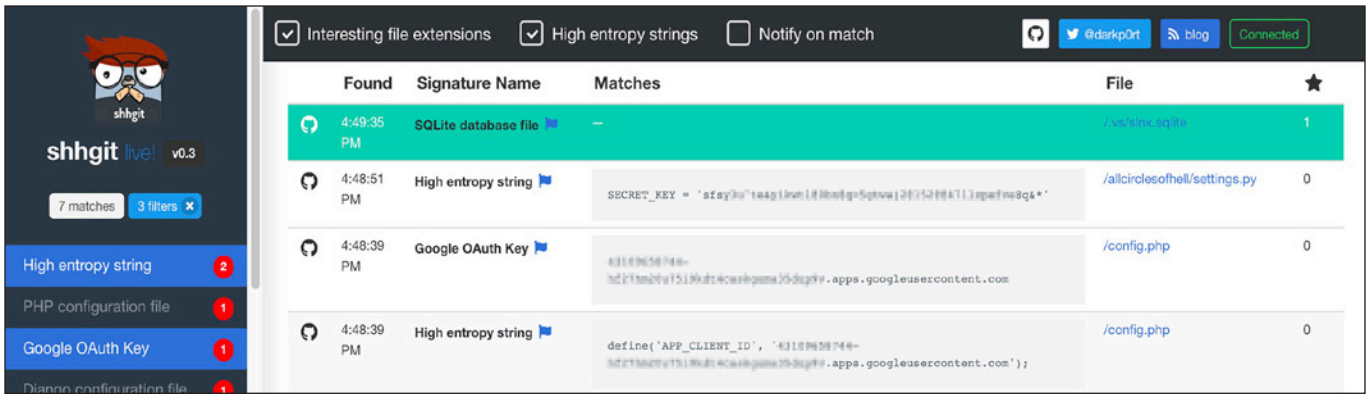


Figure 1 – Shhgit revelations

Attackers don't need to go to the trouble of a man-in-the-middle attack to steal credentials when someone has made it so easy by checking in everything they need to access each tool along the pipeline, even the destination hosts where the application is deployed to.

A code search of json files publicly visible on GitHub.com using the string *aws_access_key_id filename:*.json* produces several thousand results. You can watch people make this mistake every few seconds publicly, in near real time with shhgit.[5] The open source project by Paul Prince flags secrets and sensitive files as they are checked into GitHub in near real time (figure 1).

The Internet is forever and so is GitHub.

A similar search in commit comments containing the phrase *removing password* brings up page after page of examples of developers who caught and corrected their mistake.[6] But remember the Internet is forever. That leaked secret remains in the repository's history for all to see.

If a secret is exposed like this, change the password, revoke the token. Better yet, prevent this easy-to-make mistake in your DevOps pipeline by removing the possibility of using hard-coded credentials at all.

Now stop managing credentials

A secrets broker like HashiCorp's Vault, Confidant, Keywhiz, etc., or similar services offered by cloud service providers like AWS, Docker, or Google Cloud offer a different approach to help deal with the growing problem of secrets sprawl as DevOps becomes the norm. These brokers not only reduce the attack surface by implementing secrets programmatically via short-lived tokens, they also take care of lifecycle management, freeing you from the time-consuming and error-prone chore of rotating passwords or keys by hand.[7]

Secrets brokers like Vault provide an encrypted, single source of truth for secrets instead of spreading them across Jenkins Masters. It natively supports each of the authentication methods used by platforms mentioned here (AWS, Azure, and GCP) as well as some you might need in the future like Kubernetes, GitHub, etc.[8] You can control much of this right from your Jenkins Grand Central Station with plugins that support secrets brokers like Vault.

Role-based access control

You probably already have access control on source code, maybe even enforcing roles to limit interaction between branches and locking down sensitive areas to a trusted set of developers. This protects the source code, but don't forget to provide similar protections to binaries.[9] They are intellectual property as well, not that far removed from source code thanks to decompilers and other reverse engineering tools like the NSA's Ghidra.[10]

No user, neither flesh-and-blood nor DevOps machine-to-machine accounts, should be provided more access than necessary.[11]

Out of the box, most tools, especially artifact repositories like Artifactory, know nothing about your business. Default configurations are often designed to enable getting the tool up and running quickly rather than encouraging good security practices. Your first step with a new CI/CD tool should be configuring access to reflect the development workflow and users and systems that will interact with it.

Artifact repositories should have a minimum of three roles:

- Limited privileges intended for users and automation accounts for fetching artifacts for use in builds and orchestration tasks as read only.
- Limit privileges for destructive functions such as deleting artifacts to a separate administrative role.

5 Paul Prince, "Ah shhgit! Find GitHub Secrets in Realtime", GitHub – https://github.com/eth0izzle/shhgit/.

6 Search results, https://github.com/search?q=removing+password&type=Commits.

7 NIST Special Publication 800-53 (Rev. 4) "Alternative Security Mechanisms (CP-13), https://nvd.nist.gov/800-53/Rev4/control/CP-13.

8 HashiCorp Vault, "Auth Methods," https://www.vaultproject.io/docs/auth.

9 GitHub, "Access Permissions," https://help.github.com/en/articles/access-permissions-on-github.

10 Kelly Sheridan, "NSA Researchers Talk Development, Release of Ghidra SRE Tools," Dark Reading, https://www.darkreading.com/endpoint/nsa-researchers-talk-development-release-of-ghidra-sre-tool/d/d-id/1335536.

11 JFrog, "Configuring Security," https://www.jfrog.com/confluence/display/RTF/Configuring+Security.

- Administrators who are also users should be interacting with these systems with separate accounts to prevent costly mistakes.

Finally, log everything. DevSecOps combines a lot of distinct automation tools together. Things will go wrong and without detailed logs, finding out which tool(s) contributed and how is difficult to impossible.[12]

Now you may DevSecOps

Now that the DevOps pipeline is no longer leaking source code, artifacts, and the credentials needed to delete the whole thing, the pipeline is secure enough to get down to the business of improving the security of the products being built there.

A pattern that has been successfully applied to adding functional and other tests to continuous integration pipelines for years can be just as successfully applied using security checks as well: don't try to boil the ocean. Instead, organize all those tests that have been running through your head by a) the time it takes to run them, and b) their tendency to produce false positives.

Place those quick running, low false-positive ones in the pipeline to run on each code check. These are your security smoke tests, the most basic of sanity tests. They will catch simple problems that only waste developer, QA, and now security personnel time when they aren't stopped early. One way to determine if a test belongs in this frequently run set of tests: If you don't trust it enough to automatically notify developers or open defects, it doesn't belong in that set of very frequently run tests.

Tests that take longer to run like dynamic application security scans or those that produce false positives still provide value but should done periodically, in a separate pipeline, perhaps weekly. For those long-running, less than-reliable tests, carve out those that can be run quickly. The key to maximizing the value DevSecOps provides is continuing to evolve the tests that are run and when. For example, static security scanning is fast but notorious for producing lots of false positives. But some of those tests are quite reliable and possible only in that context, like those that find hard-coded secrets in source code. Carve those high-value, low-noise tests out and move them into the pipeline where they will be run more frequently.

Vendor tools part of, but not the complete solution

Tool vendors talk a lot about the volume of tests available across the tools they sell because it demonstrates the value of investing in their products. But that tool was built to help a broad spectrum of customers with only the most generic knowledge of your products, environment, or customers. Enabling on each option on each test will generate more noise than value, leading developers to ignore the tests.

Start small with free tools like OWASP Zed Attack Proxy (ZAP) for dynamically scanning web application projects.[13] Enable tests selectively. As you find tests that provide value, add them to the DevSecOps pipeline that make sense based on how much assistance the developer, operations engineer, or whoever will be acting on those results must perform. Some tests should be run on every code check-in, some once a week, some only before deployment.

Credentials accidentally leaked to a public source code repository should be revoked and changed. Better yet manage them in a secrets store like Vault.

You can't improve what you don't measure

Now that you've got security tests running, don't let pass/fail be the only metric you gather.

Use those measurements not just to identify problems as they are introduced into the codebase, but as an indicator of where the team needs help. When static code analysis routinely finds credentials and other secrets stored in source code (and checked into the repository), use this as an opportunity to educate those developers on more secure ways to use credentials, maybe extending a secrets broker like Vault into the application as well.

Additional value can be gained from these tests beyond individual results by looking for patterns in consecutive results. Also, when vulnerabilities found in scans are associated to defects, subsequent scans can be used to validate that the fixes submitted by developers actually resolved the vulnerability.[14]

Rushed DevSecOps Is Insecure DevSecOps

DevSecOps offers tremendous opportunity to accelerate the software development life cycle while enabling consistent, constantly improvable deployment of infrastructure. Hastily implemented, or prematurely moving half-baked, even experimental DevOps implementations could expose your source code, infrastructure and customer data.

About the Author

Tony Rice, CISSP, is a DevSecOps architect at Cisco. He regularly speaks on effectively incorporating application security into DevOps and leveraging the data that it generates to not only find vulnerabilities but generate evidence of compliance across a multitude of standards. He may be reached at trice@cisco.com.

12 NIST Special Publication 800-53 (Rev. 4), "Audit Review, Analysis, and reporting (AU-6), https://nvd.nist.gov/800-53/Rev4/control/AU-6.

13 OWASP Zed Attack Proxy Project, https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project.

14 Center for Internet Security, "CIS Control 3: Continuous Vulnerability Management," https://www.cisecurity.org.

DevSecOps

A Systemic Approach for Secure Software Development

By Seetharaman Jeganathan

The author reviews how security processes can be effectively embedded in the DevOps model to increase the success of IT projects in an organization.

Abstract

DevSecOps (security DevOps) is relatively new in the field of information security. The fundamental idea aligns with the concept of having security as an integral part of the software development principles, processes, and methodologies. The DevOps model is rapidly being adopted by the technology industry in order to support the need for developing and releasing core business systems and applications to customers in much faster and reliable ways than the traditionally followed software development life cycle (SDLC) models. The security industry has adapted to the demand for DevOps by introducing relevant processes in the form of DevSecOps principles and methods without affecting the original intent of DevOps. The author will review how security processes can be effectively embedded in the DevOps model to increase the success of IT projects in an organization. However, this article is not meant to review how to adopt DevOps for its benefits when compared to traditional approaches.

DevSecOps is relatively new in the field of information security. There are several definitions for DevSecOps; however, the more relevant ones in my view are:

> *DevSecOps enables organizations to deliver inherently secure software at DevOps speed* – Stefan Streichsbier—CEO & Founder, GuardRails
>
> *DevOps + Security = DevSecOps*

Fundamentally, information security functions have been providing confidentiality, integrity, availability, and accountability services to information systems and infrastructure. These services are often referred to as primary goals for information security functions. The primary objective is to secure the overall IT systems and business functions to support the growth of the underlying business. But in traditional software development models, security is often viewed as an afterthought wherein security testing is mostly conducted during specific testing phases of the software development life cycle, which are usually planned far ahead (right) in the schedule; hence security comes into play at a later stage in the cycle. This paradigm is changed in the DevOps + Security model where security is pushed to the forefront as much as possible and advised to begin from the early stages of the software development cycle.

DevSecOps – a practical approach

DevSecOps expects change in the organization's culture, behavior, and job functions of at least three different teams—development, security, and operations—wherein they are required to work together during all the phases of software development. Unfortunately, there is no one-size-fits-all model for all organizations. Hence, it is imperative for each organization to figure out how to work together to adopt this evolving new practice, tools, and technologies and secure the overall software delivery model. In order to achieve this, an organization's key elements such as people, process, and technology must change to adapt this shift in the culture. A conceptual model describes people, process, and technology as pillars building DevSecOps, supported by enterprise, IT, and security governance processes followed by the organization (figure 1).

People factor

Changes at the people/organization level is the first step in the journey towards accomplishing DevSecOps. In the traditional software model, different teams such as development, testing, and operations get involved during the specified cy-

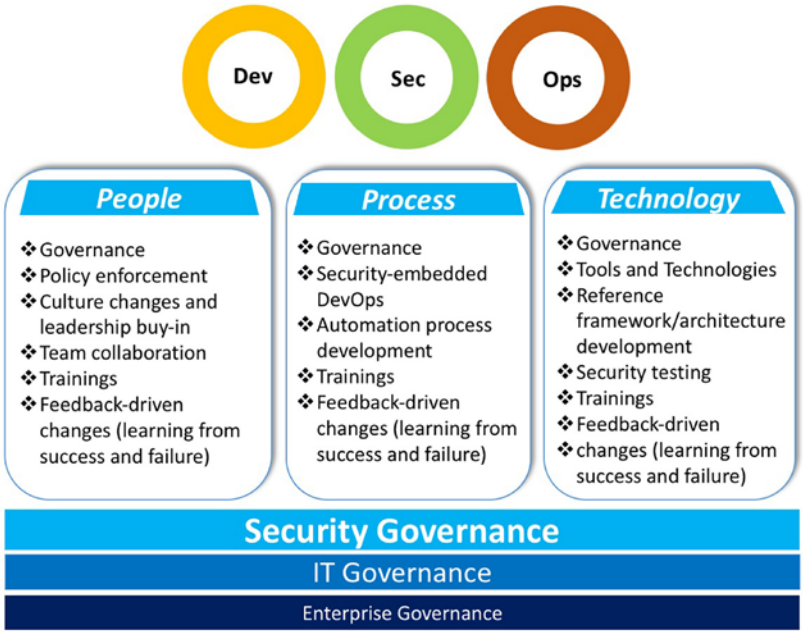


Figure 1 – DevSecOps conceptual model [3]

cles and perform their tasks in a sequential fashion. To adopt DevSecOps, we need to break these silos and make these teams collaborate from the very beginning of the development cycle.

Build a culture of security by educating the teams involved about the negative impacts of security breaches on the organization and develop the responsibility of adopting security as a foundational requirement for software development. Security cannot be compartmentalized as the responsibility of just a few security team members, but rather it is the collective responsibility of developers, testers, and operations *along with* the IT security team (figure 2).

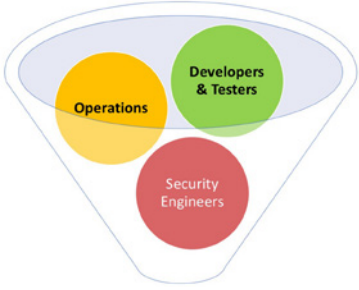


Figure 2 – DevSecOps team

This cultural shift is not going to be easy to make and succeed in its first attempt. It needs to begin with buy-in support from leadership by building an unambiguous business case with references to case studies, benefits, and ROI factors. Without leadership support, any cultural changes be in vain. Even after garnering the support, it is not advised to adopt the "go big or go home approach." For example, if an organization is running a mission-critical IT project to support its business functions, it is not advised to switch from traditional to this advanced model right in the middle of the project as it probably won't lead to success due to the various challenges involved. Instead, train the resources for this new model and run a smaller project to roll out the changes in the development processes. Learn from success and failures and reward the teams based on their performance and achieving desired outcomes. It is important to understand that DevSecOps adoption doesn't have to be at the entire organizational level from the beginning; instead it can begin with a focused group and slow rollout across several business units [4][5].

Process factor

Changes in IT & business processes come next and are vital for successful adoption of DevSecOps in an organization. Each organization is unique, depending on the IT environment complexity, architecture preferences, technology stack, risk tolerance levels, and operational maturity. Hence, changing and adopting any new processes, principles, and methods must go through careful planning, analysis, and iterative phases to successfully roll out the changes. Figure 3 shows the shift in the development process from Waterfall to Agile to DevSecOps and the reduction in the application development life time, which is a prime benefit of this change adoption.

In DevOps, everyone must focus on the customer, delivering the needs of the customers in a continuous integration and continuous delivery (CI/CD) fashion. But traditionally, security teams focus on security-centric goals to make the organization compliant with regulations and privacy laws. If security hinders the DevOps speed of CI/CD delivery model, then it will impact the success of the customer-centric delivery model. This is why security is expected to collaborate and become an integral part of the DevOps teams. Whether security adopts the concept of DevOps, or DevOps embraces security, the goal here is to deliver secure products/projects at DevOps speed. There is no hard and fast rule in defining the DevSecOps process for an organization. However, establishing a process model will lead to the next

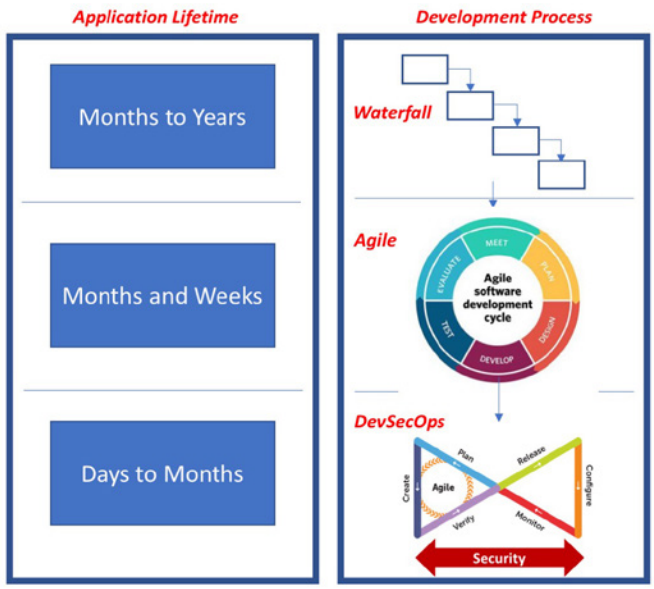


Figure 3 – Waterfall to DevSecOps [2]

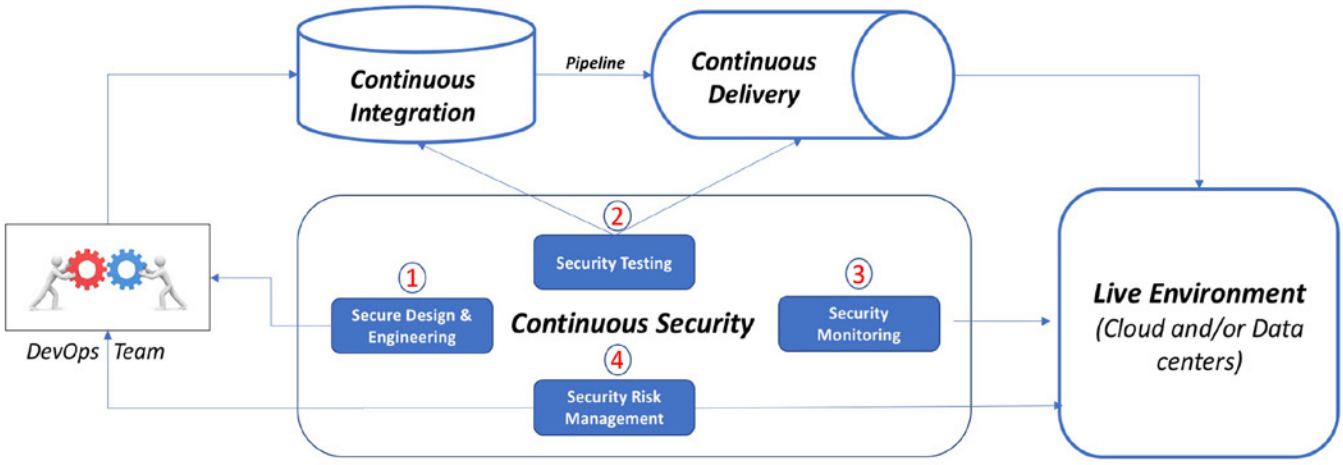


Figure 4 – DevSecOps continuous security model logical architecture [11]

step of selecting the tools and technologies to implement the process [11].

As security becomes a continuous part of the DevOps cycle, it can be referred to as a continuous security (CS) pipeline. Let's focus on this continuous security pipeline and build a process model (logical architecture) for implementing DevSecOps for any given organization. Figure 4 describes a conceptual DevSecOps process model. In effect, the process model has the following essential features:

- Continuous integration (CI)
- Continuous delivery (CD)
- Continuous security (CS)

In the proposed process model, the continuous security approach offers the following key services to the DevOps ecosystem.

(1) Security design and engineering

Security design and engineering services ensure that the products and services developed by the DevOps team comply with security best practices, regulations, standards, and laws and achieve data privacy and protection. Security requirements are carefully analyzed and properly designed during the requirements and design phase. In addition develop threat modeling (simple and complex) and implement controls during the coding stage, for example, implementing controls to prevent SANS 25 and OWASP Top 10 vulnerabilities for web-based applications.

Complex threat modeling is essential for business-critical systems to predict different possible attack vectors and to plan the system to be "fail safe" with no exposure of critical system data.

Secure coding is a development practice in which security-related weaknesses, defects, and integration errors are addressed by following the established simple and complex threat models, for example, applying controls for input validation, session management, user credential validation, user access control, data protection and privacy, logging, API security, detecting security misconfigurations, etc. while coding the software modules [3].

(2) Security testing

Security testing is the next critical facet of DevSecOps, where the software modules go through various testing cycles for quality assurance. Security testing should not only focus on the software modules but also the end-to-end pipeline, live production systems, infrastructure, databases, middleware components, and all integration points to protect from security attacks originating from any of them. When it comes to testing, it must differ from the traditional manual testing approach by adopting automation wherever possible.

The security team must collaborate with developers in testing the software modules for common vulnerability exposures such as SANS 25 and OWASP Top 10 to make sure that the security basics are followed and assured for quality. Over time these testing processes must be automated against CI/CD pipeline and all the defined test cases—functional and security—must pass before the code moves to the live production environment. Source code analysis or static application security testing (SAST) is a common process followed in analyzing the source code of software modules to detect commonly known security flaws and misconfigurations.

Developers and security teams must collaborate to integrate source code analysis into the integrated development environment (IDE) setup where coding is done to develop software modules. Likewise, dynamic application security testing (DAST) and interactive application security testing (IAST) are primarily focused on securing web applications against several know vulnerabilities before being released to the live environment. Developer, tester, and security teams must work together in implementing these mandatory testing cycles in an automated fashion in the DevOps pipeline [8][9].

The other two facets, security monitoring and security risk management are just not specific to DevSecOps but are commonly followed security principles that would add value in DevSecOps.

(3) Security monitoring

Security monitoring focuses on live and offline analysis of logs created in the live systems, infrastructure, and applications for known attacks and vulnerabilities and alerts the security team in order to respond to security incidents/loopholes. This also provides forensic capabilities when critical security incidents are being investigated. Intrusion detection, intrusion prevention, incident response, forensics analysis, etc. aren't any different for DevOps. But, they must be integrated into the process.

(4) Security risk management

Similarly, security risk management is a continuous process where security risks are analyzed and mitigated by applying cost-effective security controls. However, even risk management must work together and support the cycle at DevOps speed and not hinder the process. A lightweight approach or rapid risk-assessment (RRA) is preferred over the traditional approach for DevSecOps. Tailor the organization risk management process to DevOps and ensure the threat modeling process is handled effectively and that all applicable threats are addressed before the software module is released to the production environment. Where applicable, adopt some of the widely used threat modeling methods such as STRIDE (spoofing, tampering, repudiation, information disclosure, denial-of-service, elevation of privilege) and DREAD (damage potential, reproducibility, exploitability, affected users, discoverability) by Microsoft to support the threat modeling processes and quantify the risks to the organization for releasing vulnerable software to the live environment [6][10].

Technology factor

After the people and process pillars are defined, the focus should be on defining the technology stack to support the shift to DevSecOps adoption. We shall focus more on security considerations, tools, and technologies in this section. There are a good amount of options to choose from a wide variety of open source tools as well to support the DevSecOps technology stack. But it is important to compare and contrast the capabilities between open source and commercial products so as to make informed decisions on what is good and cost-effective for the organization. Figure 5 shows a typical

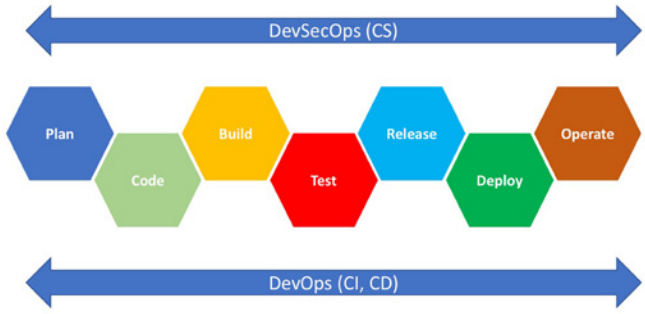


Figure 5 –DevSecOps pipeline stages (CI, CD, and CS) [9]

DevSecOps pipeline in multiple stages until it is released to the live environment.

Security considerations for the plan through operate stages of DevSecOps are outlined below. These are not exclusive or limited to what is called out but just a reference to develop further. We have the opportunity to adopt industry standards such as ISO 27002, NIST SP800-53, or others as a starting point and tailor to what will best fit for the organization.

Plan

This is the stage where security requirements and design are done for software development and securing the entire DevOps life cycle.

- Start with a high-level rapid risk assessment for the new release and quantify the risks by evaluating the threat models.
- Plan and secure the DevOps lifecycle tool, typically web-based tools such as GitLab, Azure DevOps, etc.
  - For example, secure access points based on role- or attributes-based access control models.
  - Protect user logon by integrating with company federation (identity provider) and web-access management tools if exist, otherwise with a compensating control to meet the requirements.
  - Apply 2FA/MFA based on the criticality of the environment and systems.
- Ensure user access keys, privileged service accounts, API keys, etc. are protected properly with privileged account security tools if exist, otherwise with a compensating control to meet the requirements.
- Define infrastructure protection controls and enforce segregation of duties. For example, developers don't need access to the live environment, only the operations team [1][11].

Code

- Apply secure coding practices, integrate SAST tools (OWASP SonarQube, Fortify, Veracode, Checkmarx, etc.) in the IDE tools (Eclipse, IntelliJ, Visual Studio, etc.).
- Enforce industry-followed secure-coding practices (e.g., OWASP and CERT) at this stage.

- Train developers to adopt security principles such as confidentiality, integrity, availability, and accountability while coding software modules.
- Conduct peer code reviews wherever applicable.
- Design and develop unit test scripts focusing on security along with functional unit testing of the modules, for example, input validation to prevent SQL injection attacks.
- Eliminate the use of vulnerable components from the beginning. For example, if adopting open source technologies, understand if it is being supported by an active working group to address any known vulnerabilities. Otherwise it is not advised to adopt them in the development tasks [1][11].

Build

This is the stage where software modules are checked into the source code repository and made available to package and bundle for deployments into next environments such as QA, user acceptance testing, pre-production, and production.

- Ensure best practices are followed in segregating the repository by branching it to production vs. non- production environments.
- Apply access and separation of duties controls to make sure developers aren't changing code directly in any higher branches but changing in development and passing all the required functional and security testing.
- Adopt industry-followed automated tools for building and packaging software modules (e.g., MSBuild, Maven, Gradle, etc.). Avoid manual intervention as much as possible.
- Adopt native access controls offered by the source code repository tools to prevent accidental misconfigurations, deletion of source code, and dependency errors.
- If leveraging public/private-hosted repositories such as GitHub, Git, GitLab, BitBucket, etc., additional controls are required for user access controls. Scan for privileged credentials such as password and keys to avoid security mishaps. There are known security breaches due to security misconfigurations at this stage.
- Apply 2FA/MFA to protect unauthorized access to code repositories from all the access points (web layer).

Test

Security testing is a major difference between traditional and DevSecOps development methods. Even though unit-level security testing is done by developers, extensive system and integration testing occurs at this stage to prevent various security flaws in the software modules. Security teams at this stage are referred to as blue team and red team, where red team focuses on offensive testing and blue team focuses on preventing attacks from the red team.

- Web application security scanning—leverage known tools, open source and/or commercial—to scan the built modules for known web application vulnerabilities. This is commonly referred as DAST and IAST testing techniques. There are wide range of tools to leverage for application security scanning and recommendations to fix common vulnerability exposures (e.g., OWASP ZAP, IBM AppScan, Veracode, Qualys, etc.) [8][11].
- Fuzzing tools (Radamsa, AFL, Burp Suite, ZAP etc.) that follow fuzzing techniques for negative testing and validating the behavior of software modules [11].
- Penetration testing (red team) – this is typically done by an external party with legal understanding with the orga-

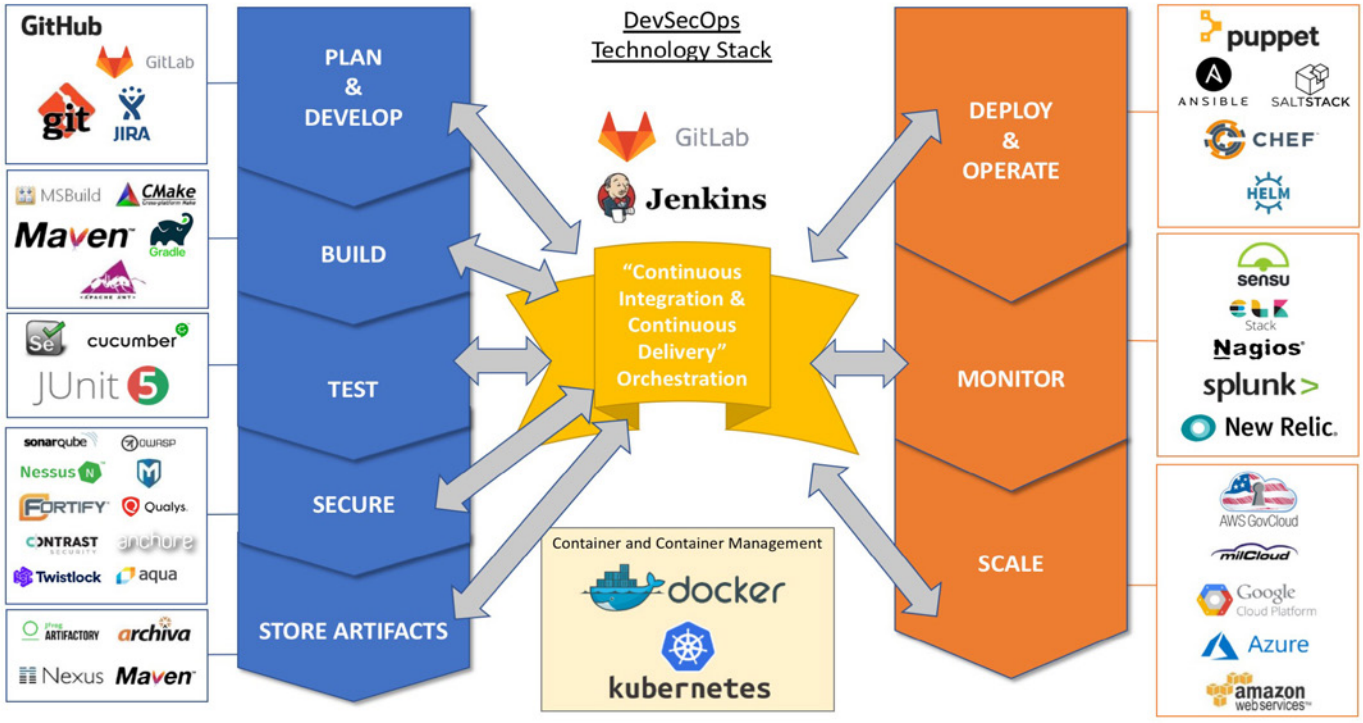


Figure 6 – DevSecOps technology stack sample reference
(Source - Nicolas-Chaillan-OSD-DoD-Enterprise-DevSecOps-Platform-DAU-Presentation-v1.3-7Mar2019)

nization to penetrate their systems and infrastructure to expose vulnerabilities and further help to fix the problems.

- Similarly, non web-based applications, APIs, data access layer, integration layer, and middleware components all must be scanned with appropriate vulnerability scanning tools and techniques for holistic security testing of the environment [3].

Release and Deploy

The release and deploy pipeline in DevSecOps is a set of processes, tools, and technologies where software modules are released to the lower and live environments on a defined schedule and/or polices that are applied for the specific release items. This is supported by various tools such as Puppet, Ansible, CHEF, etc. where scripts are written to automate the release items as per the configured policies.

If containers are used, extra caution is required in releasing the containerized modules to its storage such as Docker Hub.

Common security considerations that apply here:

- Ensure proper authentication controls are applied, prefer 2FA/MFA where applicable.
- User access controls. Grant access to users based on different DevOps roles, based on need-to-know principles.
- Protect privileged credentials by integrating with native or organization-level controls for privileged access security.
- Follow container security best practices for pushing and pulling container objects from storage. Manage the permissions in the pipeline so that accidental mishaps are avoided in the process. Adopt container signing practices for increasing the trust of containers before publishing to the storage hub [11].
- Ensure post release and deploy validation is automated to scan for common vulnerabilities such as credentials compromise, security misconfigurations, vulnerability scanning before and after release and deploy cycles. If a critical vulnerability occurs post new release, then roll back the changes immediately and fix the vulnerability before making it live.

Operate

This stage is last but not least. The operations team is heavily involved in this stage rather than the developers and testers. However, the security team is actively involved in continuous monitoring, analysis, and protection for the live-run environment, whether it is on the customer data centers or public/private cloud-based deployments.

All standard security considerations and operations controls apply here regardless of DevSecOps or any traditional method of software development cycle. Continuous security operations such as log analysis, incident response, forensic operations, intrusion detection and prevention, and post mortem analysis are some of the key considerations at this stage.

Technology factor – Selecting tools and technologies

DevSecOps is an emerging field with various tools and technologies available to support the process and principles. However, selecting the right tool sets could be a daunting task for any organization, and it heavily depends on the organi-

zation's technology adoption, training culture, and support from leadership. DevSecOps doesn't have to be an organization-wide initiative from the beginning but rather may be a small initiative from one or more of the business units and slowly expand to the entire organization based on the success and failure in the process adoption. Learn from mistakes and go through maturity cycles to figure out what would work better for your organization. As there is no one-size-fits-all, a sample is given in figure 6 of how various tools and technologies support the DevSecOps ecosystem.

Netflix's DevOps adoption case study is a personal inspiration and I am a fan of the "Simian Army" project where chaos techniques and tools are used to create havoc in the production environment and work backwards to prevent total shutdown. Adopt these principles in your journey as applicable [7].

Conclusion

As we have reviewed so far, DevSecOps is a cultural shift where people, process, and technology elements must embrace change and work together. Success or failure depends on how committed the teams are in the overall process. Starting from conceptualization through implementation, DevSecOps needs diligent effort from several teams to be successful. It is not a bad thing to fail but learn from mistakes and prevent them from re-occurring. By applying layered and continuous security for DevOps, it is mostly certain that any organization can successfully roll out this process and gain the envisioned benefits.

References

1. Allen, Ben, et al. "Continuous Security: Exploring the DevOps Toolchain," SANS, 11 Oct. 2018, https://blogs.sans.org/appsecstreetfighter/files/2018/10/DevSecOps_Exploring_Phase1-2.pdf.
2. Chaillan, Nicolas. "DoD Enterprise DevSecOps Platform," DAUAA – https://dauaa.org/wp-content/uploads/2019/03/Nicolas-Chaillan-OSD-DoD-Enterprise-DevSecOps-Platform-DAU-Presentation-v1.3-7Mar2019.pptx.
3. Elder, Michael, et al. "Security Considerations for DevOps Adoption." IBM – https://www.ibm.com/developerworks/library/d-security-considerations-devops-adoption/.
4. Janca, T. "DevSecOps: Securing Software in a DevOps World," DZone DevOps (2019, September 26) – https://dzone.com/articles/devsecops-securing-software-in-a-devops-world.
5. Lam, Thomas. "DoD Enterprise DevSecOps Reference Design" (2019, August 12) – https://dodcio.defense.gov/Portals/0/Documents/DoD_Enterprise_DevSecOps_Reference Design v1.0_Public Release.pdf?ver=2019-09-26-115824-583.
6. Mozilla. "Rapid Risk Assessment" – https://infosec.mozilla.org/guidelines/risk/rapid_risk_assessment.html.
7. Netflix Technology Blog. "The Netflix Simian Army," Medium, 20 Sept. 2018 – https://medium.com/netflix-techblog/the-netflix-simian-army-16e57fbab116.
8. OWASP. "Free for Open Source Application Security Tools." OWASP – https://www.owasp.org/index.php/Free_for_Open_Source_Application_Security_Tools.
9. Porter, Tom. "DevSecOps - A New Chance for Security," DZone DevOps, 19 July 2019 – https://dzone.com/articles/shifting-left-devsecops.
10. Shevchenko, Nataliya. "Threat Modeling: 12 Available Methods." Carnegie Mellon Univesity, 3 Dec. 2018 – https://insights.sei.cmu.edu/sei_blog/2018/12/threat-modeling-12-available-methods.html.
11. Vehent, Julien. *Securing DevOps: Security in the Cloud*. Manning Publications Co., 2018.

About the Author

Seetharaman Jeganathan, CISSP, has 17 years of experience in IT, security consulting, and project management. He focuses on information systems risk assessments, identity and access management, privileged account management, and cloud security consulting to his customers. He may be reached at seetharaman.jeganathan@gmail.com.

Securing Terminology: Lessons from Interdisciplinary Research

By Delmer Nagy, Herbert Gomez, and Christopher Copeland

IT security is an inherently interdisciplinary practice. This creates an amalgam of terms, acronyms, and concepts potentially causing confusion. Given the evolving nature of terminology, the authors demonstrate how traditional communication strategies need to be reinforced to ensure that the knowledge of organizational stakeholders does not hinder organizational security efforts.

As a relatively new discipline, information technology has encountered growing pains that are reflected in real-life situations. For example, at a recent organizational meeting (of a business that will remain unidentified), the information technology (IT) security department wanted to implement a "VM" program. A junior member of the IT department, believing he understood the IT security department's intention, showed initiative by starting a request for proposal (RFP) for VMWare and several other virtual machine (VM) vendors. But the security department wanted a vulnerabilities management (VM) program, not a virtual machine (VM) program. All the time and effort spent on the RFP was wasted, and worse still, morale and the relations between the junior member of the IT department within his own department and with the IT security staff were damaged.

IT security is an inherently interdisciplinary practice, as customers, business area experts, project managers, and IT developers all have different knowledge, skills, applications, and processes that integrate with IT security [10][11]. This creates an amalgam of terms, acronyms, and concepts potentially causing confusion. Given the evolving nature of terminology, traditional communication strategies need to be reinforced to ensure that the knowledge of organizational stakeholders does not hinder organizational security efforts. Customers, business area experts, project managers, and IT developers are all different types of stakeholders, or roles, that impact IT security. Subsequently, they need to all be talking the same language.

And because each of these stakeholders has specific knowledge and skills, and IT security needs to facilitate these stakeholders while maintaining security, IT security is an inherently interdisciplinary practice. A major challenge of any interdisciplinary effort is communication as the education, training, practice, and experience have created specific jargon, goals, and mental models for each group [6].

Compounding the differences in background are the goals of different security stakeholders. Customers, business managers, project managers, IT developers, and security analysts are all drawn together through products or services. However, customers, business managers, project managers, and IT developers can typically point at a single product or service and claim that project is "finished," "released," or "done." This perspective is in contrast with most of IT security that is recognized as a fluid and dynamic process [15]. While a specific security-related project may be completed or sunset, security is almost never "finished," "released," or "done."

This leads to the essential research question of this article: how can security practitioners start to limit communication errors among a diverse group of stakeholders surrounding organizational IT security? Entire books, university courses, and fields of study address communication problems. There is no "silver bullet" that a single article can provide for this monumental challenge. Through the examination of interdisciplinary research and research with a similar tangled group of stakeholders, this article presents three ideas to reduce communication errors: build an organizational glossary, integrate terminology into existing meetings to reinforce terms, and build a culture of communication to provide some starting points to bridge communication gaps.

Build a glossary

Doctoral students can relate to the section of their dissertation where they have a glossary of terms. This helps to put all readers of the work on common ground. Perhaps the confusion of a "VM program" could have been avoided by labeling the initiative a VMP, the more commonly accepted abbreviation of vulnerabilities management program. But the disruptive nature of information technology, to rapidly change, to introduce new functionality that can bridge multiple functional silos, and to use new standards, often introduces new terminology or re-uses existing acronyms.

The oldest and most common problem in interdisciplinary research is "talking past" stakeholders, or using a term or phrase that has a different meaning for each stakeholder [3]. As illustrated above, VM can easily take on multiple meanings when different roles need to work together [1].

To overcome similar confusion in terminology, academics who have worked on interdisciplinary teams suggest creating a common dictionary, or glossary, for terms [1]. While it may seem easier and quicker to use a third-party glossary, the question invariably becomes "Which one?"

Cybersecurity glossaries are linked to cybersecurity standards. To the uninitiated, the sheer volume of information about cybersecurity standards can be intimidating [2][26]. The amount of information is, ironically, due to how a cybersecurity standard is defined—with a common definition being "...techniques generally set forth in published materials that attempt to protect the cyber environment of a user or organization." This definition classifies any published material as a standard, from abstract-guiding documents to specific actions on individual pieces of hardware or software. To reduce the amount of information about cybersecurity standards, governmental or standard setting organizations' cybersecurity standards are often identified as cornerstone standards.

Even within this narrower group, there are many government standard-setting organizations that provide cybersecurity standards [19]. Globally, there are different perspectives between standard setting organizations [9]. But each organization represents specific stakeholder groups and provides standards for specific contexts [8][22]. For example, the International Standards Organization (ISO) and International Electrotechnical Commission (IEC) have developed cybersecurity standards to facilitate international trade. Another example is the North American Electric Reliability Corporation (NERC), which provides cybersecurity standards for power system owners and operators.

Additional standards that are far reaching and encompassing for commerce are standards that have been implemented by various private industries. The Payment Card Industry (PCI), a working body representing the major credit companies worldwide, sets standards for security compliance in the use and storage of credit cards for commercial credit transactions. The PCI-DSS is not a government standard, rather an industry-centric initiative for data protection complete with terms, conditions, and penalties for data breaches. PCI-DSS applies to any organization that uses credit card transactions globally. There are other governing bodies outside the United States. The European Union (EU) General Data Protection Regulation (GDPR) went beyond the suggesting of best practices or standardizations and regulated privacy and data protection at a legislative level. The EU body in charge of cybersecurity, ENISA, developed the frameworks for general data protections and information assurance.

One organization that has the potential to align these governmental standards setting organizations is the National Institute of Standards and Technology (NIST), which operating under the umbrella of the United States Department of Commerce is charged with promoting innovation and industrial competitiveness. But instead of setting a single, definitive cy-

bersecurity standard, or providing a number of specific standards, NIST has elected to present a general Cybersecurity Framework [19][20].

This framework is intentionally generic, perhaps so as to better meet the stated goals of promoting innovation and the development of technologies. Or, maybe this generic framework is provided to overcome the shortcomings of normative security standards [21]. While the framework itself draws from many other cybersecurity standards, the glossary reflects the challenges of flexibility. Within the NIST glossary, no singular definition is presented for each term. Rather NIST provides multiple definitions for terms and identifies where each definition is used [19][20]. While this enables the glossary to evolve and align with multiple standards, this flexibility has repercussions. Definitions for terms range from specific actions to the abstract, leaving room for interpretation of common concepts.

An extreme example could be the interpretation of access control, definitions of which range from including physical to digital limitations for systems.

An extreme example could be the interpretation of access control, definitions of which range from including physical to digital limitations for systems. Within this range a non-IT security professional could interpret the organization as having access control because the organization locks its doors at night. While the framework provides a good starting point for many terms, the framework is intended to be a series of guiding principles rather than specific definitions, or a checklist of specific actions needed for IT security [19][20]. For organizations that need to enact specific security measures, specific definitions are needed for that organization's environment.

Contrast the cybersecurity standards provided by governments and cybersecurity standards setting bodies with the cybersecurity standards needed for specific technologies. eXtensible Markup Language (XML), cloud computing, Internet of things (IOT) devices, radio-frequency identification (RFID), smart grids, and smart cars are just a small selection of cyber devices and technologies that need specific cybersecurity standards [5][12][16][17][23][24]. Many of these standards come from original equipment manufacturers, or OEMS.

With digital technologies ever expanding into new contexts, OEMs that provide cybersecurity standards come from a spectrum of industries and are not limited to those industries that manufacture hardware or code software [13]. OEMs include industrial manufacturers of all types. Unfortunately OEMS and other for-profit organizations have an incentive to use different terminology. This variance in terminology increases the switching costs associated with a provided standard and creates a barrier to switch technologies [4]. Consequently there is a history of research and growing calls by non-profit organizations that clamor for clarity [2][7][8].

So how does your organization make sense of the different standards, glossaries, and terminologies provided by government organizations, standard setting bodies, and industry organizations? Given the variation of individual terminolo-

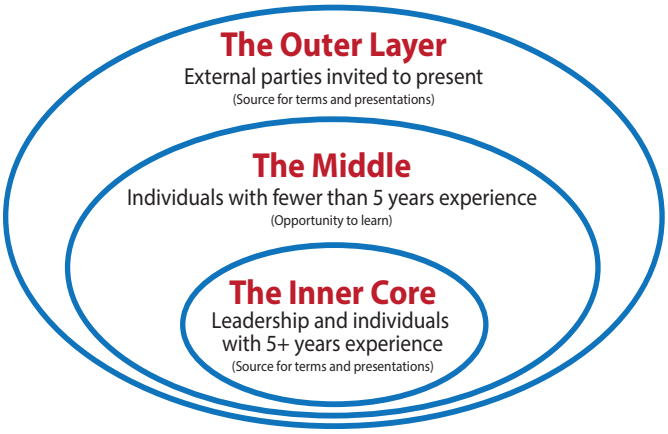


Figure 1 – "The Onion Approach" to sourcing glossary presentations

gy from one glossary to another, perhaps start by selecting a glossary from a governmental standard setting organization like NIST. This should provide an overview for terms and concepts that can then be supplemented with specific terminology used by your organization's hardware and software suppliers. These selections should be actively shared and promoted. It does little good to select glossaries other stakeholders are unaware of which definitions are being used.

While selecting glossaries, it is also important to involve non-technical stakeholders.

To improve ownership of terminology, it would be enhanced by engaging someone from each role that impacts security, allowing a communal sense of ownership and community buy-in [1][18]. Integrating customers, business managers, project managers, and IT developers into a common glossary would encourage a co-created, rather than a mandated dictionary. However, even if a glossary is co-created, the sheer amount of information available from existing sources will likely be daunting [2]. At best, stakeholders will scroll through this information. Realistically, many of these terms will be ignored [26]. Glossaries die on paper, without regular use and development. As NIST develops guidelines for the industry, so too should individual organizations keep pace.

Integrate terminology into existing meetings

The power of an organizational glossary, to standardize the meanings of terms, is magnified by processes that reinforce that content. It is logical to speak with leadership and see if they will allow for the first few minutes of a regularly scheduled meeting to cover terms and build an organizational glossary. For example, we suggest presenting a relevant term at most staff meetings. This quick overview should identify where the term is used within the organization, why the term is important, and who the term affects. By formally presenting the terms one or two at a time, the glossary of an organization will be expanded at a reasonable pace. If the presentation of a definition is held until the last part of a meeting, participants can get antsy—wanting the meeting to be over. A quick terminology overview seems most appropriate at the beginning of the meeting.

Finally, by starting a regular meeting with a short, rotating presentation, every group will have an opportunity to practice its communication skills. While presenting in front of peers can be perceived as a burden or chore, it is up to leadership to facilitate this process. This can help to remind individuals or small groups that they are a part of a larger team. How can communication improve without opportunities to communicate? Encourage out-of-the-box thinking and presentation methods for these definition presentations.

Building a culture of communication

Interdisciplinary research has a relatively short-term focus; these studies are typically limited to a single survey or a brief series of projects. The technology industry can be very similar; it is project oriented, fast paced, and part of this pace is employee turnover. Any effort to reduce communication errors between organizational stakeholders that doesn't account for employee turnover has a major oversight. While many factors influence employee turnover, drawing on a role theory perspective, an "Onion Approach" (see figure 1) is recommended to reinforce a culture of communication [14][25].

An expectation would be for "inner core" remembers—area leaders or individuals with five or more years of experience—to lead the glossary effort to identify and present new terms. While the beginning of mandatory meetings presents an ideal opportunity to introduce new terms, communication is a fluid and dynamic process. New terms can, and often are, introduced through emails and other communication. Inte-

Continued on page 42

Changing the DevOps Culture One Security Scan at a Time

By Jon-Michael Lacek

This article discusses the ideology of information security being a roadblock when it comes to DevOps project management and execution and demonstrates that available pipeline plugins do not introduce significant delays into the release process and are able to identify the vulnerabilities detected by traditional application scanning tools.

Abstract

This article discusses the ideology of information security being a roadblock when it comes to project management and execution. This mentality is even further solidified when discussing information security from a DevOps perspective. When an information technology (IT) security team has to manually obtain the application code and scan it for vulnerabilities each time a DevOps team wants to perform a release, the goals of DevOps can be significantly impacted. This frequently leads to IT security teams and their tools being left out of the release management life cycle. This article demonstrates that available pipeline plugins do not introduce significant delays into the release process and are able to identify the vulnerabilities detected by traditional application scanning tools. The art of DevOps is driving organizations to produce and release code at speeds faster than ever before, which means that IT Security teams need to figure out a way to insert themselves into this practice.

Introduction

The DevOps process is steadily gaining popularity throughout organizations; however, IT security continues to remain absent within the process. There have been several studies, such as the Ponemon Institute survey [6] that indicate the security tools available are too complex to integrate into a DevOps release pipeline or that they cannot perform an adequate security assessment compared to the stand-alone appliances.

In organizations where application security assessments are being conducted, they are traditionally assessed towards the end of the project plan as one of the final steps before the application is scheduled to be released. The IT security team will obtain the source code to perform static code analysis, followed by a dynamic assessment, which is typically managed in a certification environment where the application has been built and deployed. The results of these scans are then compiled and presented to the application developers. At this point, the development team and security team collaborate to understand the vulnerabilities presented in the document, determine how to correct them, and finally, estimate how quickly they can perform remediation. Project managers and business stakeholders have a decision to make: delay the application release in order to allow developers time to remediate, place compensating controls around the discovered vulnerabilities, or accept the risk that the known vulnerabilities present to the organization.

When organizations involve the IT security team early in the software development life cycle, the overall risk to the organization is significantly reduced. One solution for early integration is to configure scanning tools to be utilized in the DevOps pipeline build-and-release process. Not only will security teams be involved from the beginning, but this integration will now produce a continual feedback loop for developers each time they check their code back into their code repository. Since this option is available in the most common release pipeline tool sets such as Jenkins, Azure DevOps, or AWS, why has the practice of securing an application not become a standard exercise across all organizations with a DevOps culture?

The research presented in this paper will evaluate the work effort needed to integrate some of the existing application scanning extensions available in the most frequently used DevOps

pipeline release products. It will also evaluate the quality of scanning that they provide compared to the tools used in a traditional source code and dynamic analysis engine.

Figure 1 – Sample pipeline without security extensions

Impact of the DevOps Culture

The digital transformation is well underway across all business verticals, and the culture of DevOps is at the heart of the movement. High performing organizations, such as Google, Amazon, Facebook, Etsy, and Netflix, are routinely and reliably deploying code into production hundreds, or even thousands, of times per day using the continuous integration/continuous deployment (CI/CD) methodology [3]. How do organizations continue to create new applications or integrate feature enhancements to existing applications and deliver them at such a rapid pace all while ensuring the deployed code does not contain any vulnerabilities? A continual feedback loop is the cornerstone of the Agile development process. Therefore, when executed correctly, developers will continually receive feedback on the vulnerabilities present in their code. This new feedback loop will slowly change the security mind-set of the developers, consequently making secure coding a fundamental and sought-after skill within the organization, thus creating a culture commonly referred to as DevSecOps.

Figure 2 – Pipeline with security extensions added in two additional stages

This article demonstrates that the complexity of integrating the necessary tools in a CI/CD pipeline is no greater than the time or expertise needed to provide a security assessment of a web application when performed independently of the deployment of that same application. Moreover, "when security is integrated into the DevOps culture, high performing teams spend 50 percent less time remediating security issues than low performers. This is because they build security into the SDLC in contrast to retrofitting security at the end" [1].

By enabling the automation of static and dynamic analysis tools in a release pipeline, development teams automatically receive the vulnerability information present within their code each time the application is checked into its repository or sent through the build and release pipelines. The time savings realized when utilizing this process is two-fold. First, developers will be able to deploy applications or feature releases quicker as their secure coding skills increase. Security analysts can now spend their time on dynamic assessment, ethical hacking, or red teaming instead of continually configuring tools to scan applications. Perhaps the greatest benefit isn't the time savings itself, but rather that application scanning is taking place daily, at minimum, rather than only during the times IT security teams are made aware of application or feature releases, and is given the appropriate amount of time to assess those deployments.

A sample pipeline example has been constructed as shown in figure 1. This example is considered to be the traditional pipeline configuration without the integration of security scanning extensions. Figure 2 depicts an example configuration with two additional stages included that initiate the application scanning extensions discussed below.

Research method

Lab design

This research is being conducted using the Microsoft suite of tools available in the Azure cloud. An Ubuntu server has been created to host the application WebGoat, "a deliberately insecure web application" [4] that will be used as the application to evaluate the Azure DevOps pipeline extensions capabilities. The testing will include both static code analysis along with dynamic application scanning aspects that IT security analysts typically take when evaluating the overall security of a web application that is ready for consumption by the customer. Static code analysis will be conducted during the build process, while the dynamic application scan will take place during the release pipeline.

Testing methodology

Establishing a build and release baseline

Establishing a baseline requires measuring the time it takes to build and release the WebGoat application without tying in any security extensions. Security extensions such as SonarQube and ZAP will be integrated into the pipeline configurations to automatically evaluate the application from a static code analysis standpoint, as well as a dynamic analysis standpoint while it is in the process of deployment.

The first phase in the pipeline release process involves the build of the application. The WebGoat application files have been downloaded and stored in a Microsoft Azure Git repository. The build pipeline is configured to reference this code repository and compile the application. Capturing the time to build WebGoat without any static code analysis taking place establishes the baseline. The Microsoft Azure DevOps pipeline interface provides timings for each step in the build process, which takes approximately 3 minutes and 29 seconds to compile, as shown in figure 3 (next page).

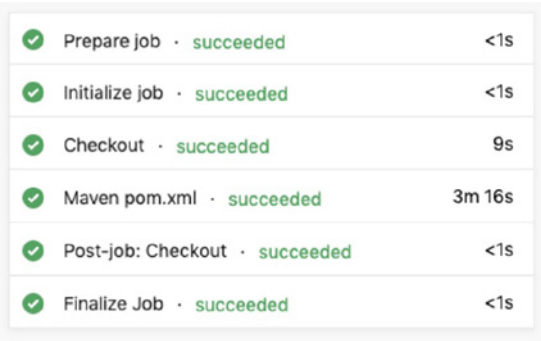


Figure 3 – Baseline build pipeline results

The application is then deployed using the Azure DevOps release pipeline configuration to a virtual server in the Microsoft Azure cloud. When initiated, this process takes approximately 28 seconds, as shown in the figure 4.

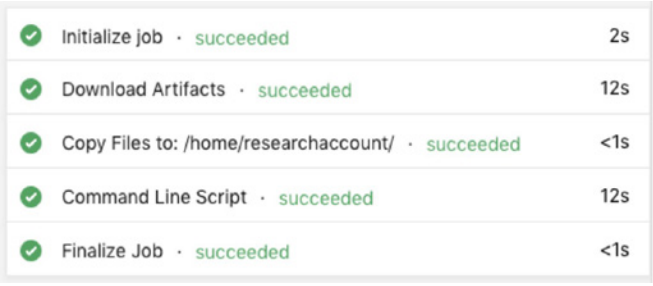


Figure 4 - Baseline release pipeline results

Establishing an application scan baseline

At this point, the application is running in a container on a VM. The installation of the OWASP Zed Attack Proxy (ZAP) takes place on a second virtual server running in the Microsoft Azure portal. The configuration of the proxy is then conducted to reference the WebGoat application previously installed and referenced above. This process is what is typically involved in a traditional means of dynamic application scanning where IT security analysts are provided with the web application URL for configuration and analysis within the application scanning tool.

The WebGoat application is composed of two websites: WebGoat and WebWolf, which interact with each other. When the ZAP tool is configured to point to the URLs and perform a spider of the sites, followed by an active scan. Table 1 shows the results obtained.

Time to spider the WebGoat	20 seconds
Time to scan the WebGoat	1 minute, 56 seconds
Time to spider the WebWolf	21 seconds
Time to scan the WebWolf	2 minutes, 5 seconds

Table 1 – Baseline application scan results

Table 2 documents the results of the active application security scan of both applications in the Docker container.

Vulnerabilities	Critical	High	Medium	Informational	Total # of URLs
WebGoat	2	1	3	0	11
WebWolf	2	0	0	0	5

Table 2 –Baseline vulnerability scan results

Automatic build integration

With the baseline timing to build the WebGoat application established, it is time to gather statistics indicating how much additional time static code analysis tests take when integrating with a build pipeline. The original build pipeline configuration referenced above has been modified to incorporate SonarQube extensions: "SonarQube is an open source product for continuous inspection of code quality" [2]. Since WebGoat is a Java-based application, SonarQube was chosen as the Azure pipeline extension to utilize. OWASP has additional static code analysis applications.[1] The list presented on the OWASP site provides the available coding languages that are supported by each tool.

When this build process starts, three additional steps are added to the pipeline. These steps add an additional 75 seconds in this build pipeline. The steps required to prepare SonarQube and reconfigure the build pipeline to publish the testing results to the SonarQube portal cannot be captured systematically. The steps needed for initial integration include deploying a SonarQube server or virtual instance of SonarQube, configuring a new project within SonarQube, and obtaining the necessary code snippet that is needed for configuration into the build pipeline for integration. Deploying and configuring a SonarQube server is a one-time setup. Configuration of each project pipeline that organizations are looking to integrate occurs once per project. After connecting the SonarQube project within each application pipeline, developers now have the benefit of continually receiving feedback about their code immediately after the completion of the build process.

Once the SonarQube extension has been configured to integrate with the build pipeline, the build is initiated and then completes, which subsequently publishes the results of the static code analysis to the SonarQube project page. Developers now have an extremely easy-to-use tool that describes what the problem is, why it is a problem, and the location of that problem within the code.

This summarized view of issues within the application are determined as the SonarQube extension evaluates the application and documents any piece of code that breaks a coding rule within its analysis. The evaluation of the application's code is broken down into one of three categories: bugs, vulnerabilities, and code smells [7]. While the scores will be interpreted differently across organizations, it is important to note that the developer now has a tool that provides immediate feedback each time the code is built. Tools like So-

1 OWASP Static Code Analysis, www.owasp.org/index.php/Static_Code_Analysis.

narQube provide resources to understand why that particular coding practice creates a vulnerability or bug and examples on how to correct the problems detected within the code. By repetitively reviewing problems within their code, over time, developers will change their poor coding habits.

In addition to the benefits realized from an application security perspective, most of the pipeline extensions provide valuable reports that contribute to enhancing the quality of the application in development. The SonarQube extension contains a section referred to as *code smells*, which highlights segments of the code that are confusing and difficult to maintain. Additional information about the code is classified into categories such as technical debt, code coverage, and duplications.

Automatic release integration

After the completion of the build process, a release pipeline can be constructed to deploy the artifacts produced as a result of the successful build. At this stage we can integrate dynamic application scanning tools, such as the Zed Attack Proxy, that will provide information regarding the most common, known vulnerabilities found in web applications. The testing conducted in this lab environment resulted in the addition of 3 minutes, 35 seconds to the release pipeline execution.

Tables 3 and 4 show the time taken to scan each application along with the number of vulnerabilities discovered for each URL.

Time to spider and scan WebGoat	1 minute, 49 seconds
Time to spider and scan WebWolf	1 minute, 28 seconds

Table 3 – Application spider and scan timings

App	# of Vulnerabilities Discovered	# of URLs
WebGoat	4	10
WebWolf	3	9

Table 4 - Application vulnerability count in automated pipeline integration

Findings

As shown in tables 5 and 6, the testing conducted in this research confirmed the hypothesis that the tools available in CI/CD pipelines have the same capabilities that stand-alone products have. Additionally, the time needed to integrate these tools is not significantly shorter or longer than the manual scanning methodology that traditional security teams utilize.

The success of integrating application scanning extensions within a CI/CD pipeline hinges on a well-planned process.

	Baseline timing	ZAP integration timing
Time to spider and scan WebGoat	2 minutes, 16 seconds	1 minute, 49 seconds
Time to spider and scan WebWolf	2 minutes, 26 seconds	1 minute, 28 seconds

Table 5 – Application spider and scan comparison timings

	# of Vulnerabilities		# of Vulnerable URLs	
	Stand-alone	Pipeline	Stand-alone	Pipeline
WebGoat	6	4	11	10
WebWolf	2	3	5	9

Table 6 – Application vulnerability scan comparison timings

The research conducted here demonstrates easy-to-accomplish steps to start integrating security tools into the DevOps application development pipelines. Once integration is concluded, security analysts have the ability to continue to enhance the scanning capabilities and the actions taken as a result of those scans. This configuration brings security into the CI/CD pipeline process by allowing them to continue to build upon the basic scanning tests conducted in this research.

Much like the developer of the applications utilizing a CI/CD pipeline, the security analysts responsible for integrating these extensions will continually be able to introduce enhancements in the form of advanced application checks. These checks would be configured similar to how developers configure integration tests in their release pipelines.

Although there is no significant difference in time between the traditional application scanning configuration versus the initial CI/CD configuration, a notable amount of time will be saved from that point on. Since the execution of the application scanning extension will be performed each time the application is deployed, there are no future time requirements involved. On the contrary, notification of each release must include IT security, giving them time to analyze the new code if organizations continue to use the traditional method of analysis. Historically, IT security teams are left out of future releases to existing applications, which can present a significant risk to the organization if vulnerabilities are introduced to the code.

An important point to note is that organizations would need to adopt a mentality that the integrated application scanners in the CI/CD pipeline would be evaluating the most common, well-known vulnerabilities and common coding mistakes. Understanding this would allow security analysts to dedicate more time to the in-depth functionality testing of the application while knowing that the baseline scan remains consistent during each scan.

Taking it a step further

As organizations continue to mature in their DevOps initiatives, the capabilities within their pipelines continue to grow. Maturity comes in the form of unit testing that is built out to ensure functionality within each aspect of the code or in the logic of the gates between each stage in a pipeline. Incorporating this mentality from a security perspective requires a strong partnership with the development team whose goal continues to be to deploy application and feature updates regularly.

A good starting point would be the integration of the analysis tools into the pipelines without impacting the build and release functionality. This step would be to gain visibility to the vulnerabilities and poor coding practices that exist in the application. When development teams have had a chance to review the output of the scans and consult with security analysts to understand and correct the vulnerabilities, a decision should be made to implement a gate that would fail the progression to the next stage in a pipeline, should a vulnerability be detected.

DevOps pipelines have a wide variety of control mechanisms within them that control the progression of an application through the pipeline. The concept of gates, which are configured to control whether or not the next stage of the pipeline can start, should be utilized as a maturity mechanism. As organizations begin introducing the security tools into their release pipelines, they could decide to configure the extensions passively. Using this methodology, organizations can automate the analysis of the source code and the scanning of the application, all while still allowing the release of the product to the customers. Reports are generated as a result of the security extension integration, allowing development teams time to review the vulnerability reports. Developers and security analysts can then prioritize them accordingly within their backlog of work.

An additional benefit of integration within a CI/CD pipeline is for applications that are not released on a regular or frequent cadence. Security analysts could work with the developers to set a scheduled release, which would trigger the application scan extensions. This re-occurring schedule would not impact the functionality of the application since the source code has not changed. Many of the dynamic code analysis tools are continually developing the tests that are executed within their scans to look for the up-to-date vulnerabilities discovered in the wild. For example, after the identification of the Heartbleed vulnerability, the ZAP community configured tests into the active scanner functionality to test each application it scans for its presence. If a pipeline is configured using security extensions and is set to release on a regular schedule, even if there have not been any changes to the code, developers would be notified if their application is vulnerable to the newly discovered vulnerability.

Another benefit of integrating security tools in the CI/CD pipeline is how the notification of vulnerabilities within an application can be configured. Many of the extensions have built-in functionality that integrates back into the tools that Agile teams use to manage their work. For example, in the Microsoft Azure DevOps tool sets, security analysts have the option of automatically creating work orders or bug items for each detected vulnerability and placing them in the developer's list of backlog tasks. Using this capability will help drive the adoption of integrating security into the DevOps culture. Developers will be far more likely to address the vulnerabilities within their code if they don't need to take manual steps to review and understand the discovered vulnerabilities. The extensions available provide the details about the vulnerability, which include resources on how to correct them.

Conclusion

Changing the culture in any organization involves the commitment of multiple teams and doesn't happen overnight. With the tools available today and the desire to continually

release applications or feature updates to existing applications at an increasingly rapid pace, the IT security community must help keep those applications, and the data they process, safe and secure. By leveraging the available CI/CD pipeline extensions, organizations can start integrating a security mind-set in the DevOps process and put secure coding at the forefront in the minds of the developers. Achieving this must and can be accomplished without forcing developers to abandon the CI/CD tools they are accustomed to today.

References

1. 2018 State of DevOps Report. (n.d.). Retrieved from https://puppet.com/resources/whitepaper/state-of-devops-report.
2. Docker Hub. (n.d.). Retrieved from https://hub.docker.com/.
3. Kim, G., Debois, P., Willis, J., Humble, J., & Allspaw, J. (2017). *The DevOps handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. Portland, OR: IT Revolution Press, LLC.
4. OWASP WebGoat Project. (2019). Retrieved from https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project.
5. OWASP Zap Attack Proxy Project. (2019). Retrieved from https://www.owasp.org/index.php/Category:OWASP_WebGoat_Project.
6. Ponemon Institute – "Bridging the Digital Transformation Divide: Leaders Must Balance Risk & Growth" (2018) Retrieved from https://www.ibm.com/downloads/cas/ON-8MVMXW.
7. SonarQube Resources. (2019). Retrieved from https://sonarqube.org/features/integration/.

About the Author

Jon-Michael Lacek is currently seeking a Master of Science in Information Security Management at SANS Technology Institute. With nearly 15 years of experience in the network and security fields, he is looking to leverage that experience as he transitions into management where he currently oversees the Operations, NOC, DevOps, and Automation teams at a privately held retail organization. In addition to the CISSP, Jon-Michael currently holds 5 GIAC certifications. He may be reached at jmlacek@gmail.com.

The Open Forum

The Open Forum is a vehicle for individuals to provide opinions or commentaries on infosec ideas, technologies, strategies, legislation, standards, and other topics of interest to the ISSA community. Articles should be around 850 words and include short bio and photo. Please submit to editor@issa.org.

ISSA CAREER CENTER

The ISSA Career Center offers a listing of current job openings. Among the current 916 job listings [11/1/19] are the following:

- **Manager, Emerging Standards**, PCI Security Standards Council – Remote, NA
- **Information Security Officer or Chief Information Security Officer**, Desert Research Institute – Reno, NV
- **Senior Security Engineer**, VHDA – Richmond, VA
- **BI Analyst / Senior BI Analyst**, Masco Support Services – Mooresville, NC
- **Senior Information Security Analyst**, VHDA – Richmond, VA
- **Cybersecurity Specialist,** Ada County Highway District (ACHD) – Boise, ID
- **County Privacy Officer/Ethics Officer**, San Bernardino County – San Bernardino, CA
- **Information Security Analyst**, Inspira Health – Vineland, NJ
- **Chief Information Security Officer and Senior Director of Information Security**, Northwestern University – Evanston, IL
- **IT Security Risk Assessment Analyst/IT Auditor,** Office of Information Security New York University – New York, NY
- **Chief Information Security Officer**, LCRA – Austin, TX
- **Information Security Researcher**, Consumer Reports – Yonkers, NY
- **Information Security Engineer-Perimeter Security Operations Manager**, PricewaterhouseCoopers – Detroit, MI
- **Information Security Engineer-Perimeter Security Operations Manager,** PricewaterhouseCoopers – Tampa, FL
- **Information Security Analyst (work from home),** American Red Cross – Washington, NA
- **Sr. Information Security Engineer**, Fifth Third Bank – Cincinnati, OH
- **Business Information Security Officer**, Fifth Third Bank – Cincinnati, OH
- **Information Security Eng. Sr,** Dollar General – Goodlettsville, TN
- **Senior Information Security Specialist**, CareFirst BlueCross BlueShield – Owings Mills, MD
- **Manager, Information Security Office**, Rice University – TX
- **Information Security Engineer**, Aetna – Hartford, CT
- Information Security Analyst-Encryption and Key Management, American Express – Phoenix, AZ

The Python Programming Language: Relational Databases

#8

By Constantinos Doskas – ISSA Senior Member, Northern Virginia Chapter

This article continues our discussion on database programming. In previous lessons we learned how to create SQL database tables, how to create INNER and LEFT JOIN, and how to ORDER the queries of tables by one or more columns. In this session we will learn how to combine data of multiple like tables and queries and create detailed or summary reports.

The business scenario

A company has multiple office/branch locations, and each location maintains a database of equipment that comprise the company's network. You are working for the VP of the network and security department. Every quarter you must create a report that lists all the network and security equipment that are owned by the company. You superiors also want a summary of the types of equipment in use.

Available data

The company operates in three states/locations: Washington metro, New York, and Los Angeles. Each location maintains a database with tables related to network equipment and configuration. To complete this task we must join data from these three locations into one table. Then, we will use this table to complete our task. Local administrators provided us with read access to three tables: security_appliances_DC, security_appliances_NY, security_appliances_LA.

In SQL methodology a UNION of two or more tables results in a union of unique records (rows) and a UNION ALL results in table of the all data from the tables involved. Therefore, a UNION ALL may result in duplicate records (rows).

The tables

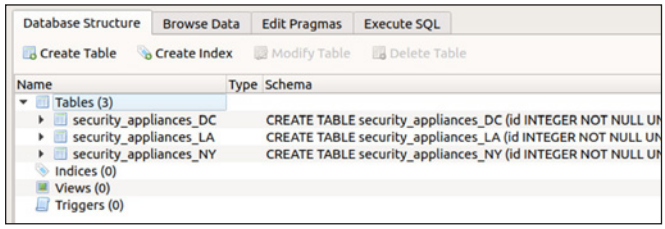


Figure 1 – The tables from the three company branch offices

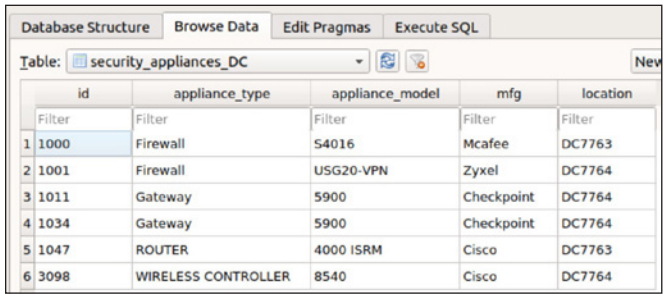


Figure 2 – The table from Washington DC location

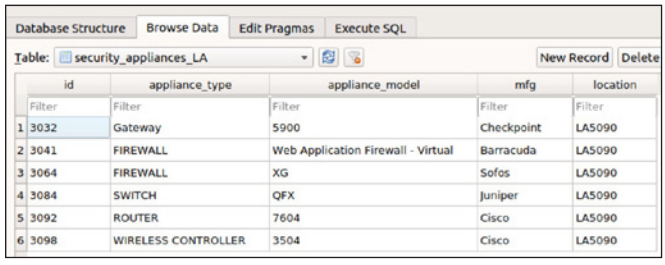


Figure 3 – The table from Los Angeles location

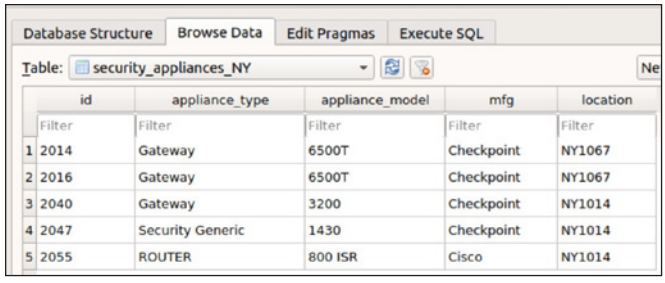


Figure 4 – The table from New York location

Based on the above information a UNION ALL will give us data for the detail report and UNION will provide us with the data needed for the summary report.

Solution

The SQL statements that we will use have the following structure:

```
CREATE TABLE New_Table_Name AS
    SELECT column1, column2, column3
    FROM Table1
    UNION ALL
    SELECT column1, column2, column3
    FROM Table2
    UNION ALL
    SELECT column1, column2, column3
    FROM Table3

CREATE TABLE New_Table_Name AS
    SELECT column1, column2, column3
    FROM Table1
    UNION
    SELECT column1, column2, column3
    FROM Table2
    UNION
    SELECT column1, column2, column3
    FROM Table3
```

Note that the tables that are in use in each location may have columns that are specific to the location. To avoid having issues merging the tables we have to select only the columns that are common and have the same type of data.

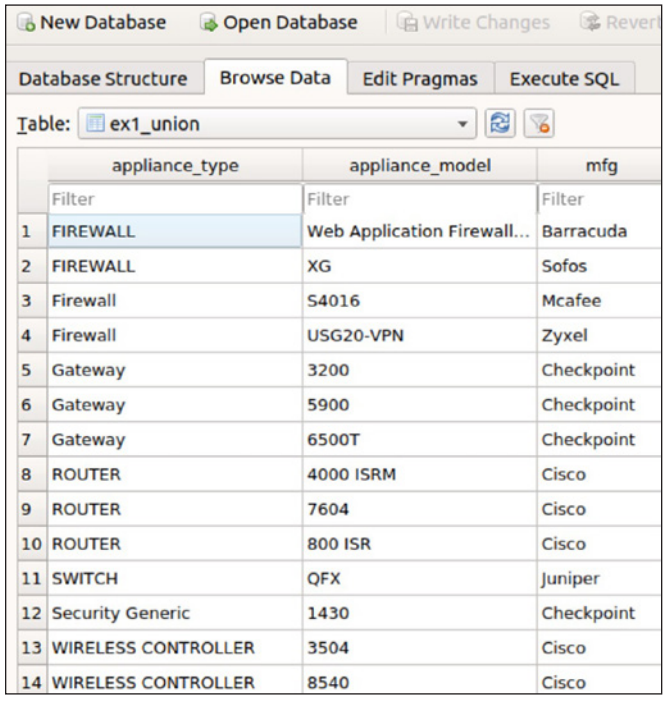
Here is the listing of the code:

```
# -- --------------------
# -- SQLite database and Python
# -- Constantinos Doskas
# -- Fall 2019
# ------------------- Topics --------
# - 1. Table Merging, uing UNION and UNION ALL
# -- --------------------------------
# -- Import the sqlite3 module
import sqlite3
# -- Create a reference variable and open the
     database
dbase=sqlite3.connect('UNION_EXAMPLES.db')
# -- A. Create a cursor
# -- It will be used to execute database
     statements
mydbCursor=dbase.cursor()
# -- B. Remove equipment table if existS
mydbCursor.execute("DROP TABLE IF EXISTS select_DC")
mydbCursor.execute("DROP TABLE IF EXISTS select_NY")
mydbCursor.execute("DROP TABLE IF EXISTS ex1_union")
mydbCursor.execute("DROP TABLE IF EXISTS ex2_union_
     all")
mydbCursor.execute("DROP TABLE IF EXISTS ex2_union_
     all_orderby")
mydbCursor.execute("DROP TABLE IF EXISTS ex2_union_
     all_orderby_desc")
mydbCursor.execute("DROP TABLE IF EXISTS ex2_union_
     all_orderby_asc_desc")
# --
list_of_tables=['security_appliances_DC',
     'security_appliances_NY','security_appliances_LA']
# -- The following 'for'
     loop will assign the value False if one of
     the tables (...DC NY or LA)
# -- does not exist. The program will not run
     unless all three tables are present.
may_continue=True
for aTable in list_of_tables:
 # --
 mydbCursor.execute(
  "SELECT * FROM sqlite_
     master WHERE type='table' AND name='{}'".
     format(aTable))
 ans=mydbCursor.fetchone()
 if ans==None:
  print('Table',aTable,' not found')
  may_continue=False
# -- C.
```

```
if may_continue:
 print('Merging and processing tables')
 mydbCursor.execute("""
 CREATE TABLE select_DC AS
  SELECT appliance_type, appliance_model, mfg
  FROM security_appliances_DC
 """)
 mydbCursor.execute("""
 CREATE TABLE select_NY AS
  SELECT appliance_type, appliance_model, mfg
  FROM security_appliances_NY
 """)
 # -- UNION combines two similar tables and
      produces unique Rows
 mydbCursor.execute("""
 CREATE TABLE ex1_union AS
  SELECT appliance_type, appliance_model, mfg
  FROM security_appliances_DC
  UNION
  SELECT appliance_type, appliance_model, mfg
  FROM security_appliances_NY
  UNION
  SELECT appliance_type, appliance_model, mfg
   FROM security_appliances_LA
 """)
 # -- UNION ALL combines two similar tables
      and produces all the Rows
 mydbCursor.execute("""
 CREATE TABLE ex2_union_all AS
  SELECT appliance_type, appliance_model, mfg
  FROM security_appliances_DC
  UNION ALL
  SELECT appliance_type, appliance_model, mfg
  FROM security_appliances_NY
  UNION ALL
  SELECT appliance_type, appliance_model, mfg
  FROM security_appliances_LA
 """)
# -- Sorting using the ORDER BY clause -
      Defaults to Ascending
 mydbCursor.execute("""
 CREATE TABLE ex2_union_all_orderby AS
  SELECT appliance_type, appliance_model, mfg
  FROM security_appliances_DC
  UNION ALL
  SELECT appliance_type, appliance_model, mfg
  FROM security_appliances_NY
  UNION ALL
  SELECT appliance_type, appliance_model, mfg
  FROM security_appliances_LA
  ORDER BY mfg, appliance_model
 """)
# -- Sorting using the ORDER BY clause -
      Using the DESC keyword
 mydbCursor.execute("""
 CREATE TABLE ex2_union_all_orderby_desc AS
  SELECT appliance_type, appliance_model, mfg
  FROM security_appliances_DC
  UNION ALL
  SELECT appliance_type, appliance_model, mfg
  FROM security_appliances_NY
  UNION ALL
  SELECT appliance_type, appliance_model, mfg
  FROM security_appliances_LA
  ORDER BY mfg DESC, appliance_model
 """)
# -- Sorting using the ORDER BY clause -
      Using the DESC and ASC keyword
 mydbCursor.execute("""
 CREATE TABLE ex2_union_all_orderby_asc_desc AS
  SELECT appliance_type, mfg, appliance_model
  FROM security_appliances_DC
  UNION ALL
  SELECT appliance_type, mfg, appliance_model
  FROM security_appliances_NY
  UNION ALL
  SELECT appliance_type, mfg, appliance_model
  FROM security_appliances_LA
  ORDER BY mfg ASC, appliance_model DESC
 """)
else:
 print("Create table, operation failed. Table(s)
      not found")

# -- D. Close the database
dbase.close()
```

The new tables



	appliance_type	appliance_model	mfg
	Filter	Filter	Filter
1	FIREWALL	Web Application Firewall...	Barracuda
2	FIREWALL	XG	Sofos
3	Firewall	S4016	Mcafee
4	Firewall	USG20-VPN	Zyxel
5	Gateway	3200	Checkpoint
6	Gateway	5900	Checkpoint
7	Gateway	6500T	Checkpoint
8	ROUTER	4000 ISRM	Cisco
9	ROUTER	7604	Cisco
10	ROUTER	800 ISR	Cisco
11	SWITCH	QFX	Juniper
12	Security Generic	1430	Checkpoint
13	WIRELESS CONTROLLER	3504	Cisco
14	WIRELESS CONTROLLER	8540	Cisco

Figure 5 – The table resulting from the UNION statement

Comments

Note that once the tables are merged we may use simple SELECT statements to process the data. However, the new tables

Database Structure | Browse Data | Edit Pragmas | Execute SQL

Table: ex2_union_all_orderby

	appliance_type	appliance_model	mfg
	Filter	Filter	Filter
1	FIREWALL	Web Application Firewall - Virtual	Barracuda
2	Security Ge...	1430	Checkpoint
3	Gateway	3200	Checkpoint
4	Gateway	5900	Checkpoint
5	Gateway	5900	Checkpoint
6	Gateway	5900	Checkpoint
7	Gateway	6500T	Checkpoint
8	Gateway	6500T	Checkpoint
9	WIRELESS C...	3504	Cisco
10	ROUTER	4000 ISRM	Cisco
11	ROUTER	7604	Cisco
12	ROUTER	800 ISR	Cisco
13	WIRELESS C...	8540	Cisco
14	SWITCH	QFX	Juniper
15	Firewall	S4016	Mcafee
16	FIREWALL	XG	Sofos
17	Firewall	USG20-VPN	Zyxel

Figure 6 – The table resulting from the UNION ALL statement.

Database Structure | Browse Data | Edit Pragmas | Execute SQL

Table: ex2_union_all_orderby_asc_desc

	appliance_type	mfg	appliance_model
	Filter	Filter	Filter
1	FIREWALL	Barracuda	Web Application Firewall - Virtual
2	Gateway	Checkpoint	6500T
3	Gateway	Checkpoint	6500T
4	Gateway	Checkpoint	5900
5	Gateway	Checkpoint	5900
6	Gateway	Checkpoint	5900
7	Gateway	Checkpoint	3200
8	Security Generic	Checkpoint	1430
9	WIRELESS CONTROLLER	Cisco	8540
10	ROUTER	Cisco	800 ISR
11	ROUTER	Cisco	7604
12	ROUTER	Cisco	4000 ISRM
13	WIRELESS CONTROLLER	Cisco	3504
14	SWITCH	Juniper	QFX
15	Firewall	Mcafee	S4016
16	FIREWALL	Sofos	XG
17	Firewall	Zyxel	USG20-VPN

Figure 7 – A table resulting from a UNION ALL statement ordered on mfg Ascending and appliance model Descending order.

may be sorted on one or multiple columns, which makes it easier to browse the data without further processing.

You may ask how can I apply this to make my everyday tasks easier. It depends on what type of tasks you are assigned and the nature of the data. Way back before network security became so very important, I worked in the telecommunications industry and we applied these principles to our everyday tasks as follows.

Telecommunications companies employ multiple heterogeneous devices on their networks. All these devices have one thing in common: they constantly produce various types of messages. These messages vary from "a door was opened in a facility" to malfunction warnings to system errors. There were so many different types of messages that there was a need to have specific departments monitoring subsets of these alarms. To make it easier for these groups to exchange information on related issues, we created a huge room where each group had it own workstations. One wall of this room was used to project data from computers via huge projectors. The data displayed was first merged onto one computer, and then it was displayed in real time but grouped by some form of relevancy.

In addition, data once merged was stored on IBM DB2 database tables for analysts to further search for causes of network failures or events that could help them predict a future failure of a device and order maintenance. Therefore, we had reactive and proactive capabilities. When you think about it, this is what is happening today with cybersecurity groups that monitor data on WireShark screens or use other data analysis tools. Data in this case is collected from workstations, servers, routers, firewalls, etc., merged, and presented. Glass-Lookers may be able to identify threats in real time. Events that can't be correlated in real time will be further processed and analyzed by scripts once the data is stored in database tables.

Review and conclusion

This article is part of a series of articles that aim to help cybersecurity professionals understand how information is acquired, organized, stored, searched, presented, and transferred within computer networks. SQL databases carry most of the weight of data processing in most data centers. Coupled with GUI development for client-server systems or web development, SQL databases play a central role in many cyber technology innovations.

So far we have examined most of the ways that data can be retrieved from database tables. Python is one of the top languages used in developing SQL-driven systems. I hope that you enjoyed the article and that you will find ways to apply the presented concepts. I am moving slowly on these concepts, giving you time to run the code and experiment. ISSA International makes available the code on the website.

You are always welcome to email me with any questions you have about the presented concepts. I wish you well and will be pleased to "see" you through the next article.

About the Author

Constantinos Doskas is head of the IT and Security Department of Olympus. He has been involved in information systems management and development for over 30 years. He is currently involved in mentoring graduate students and ISSA members in Northern Virginia. Topics include various programming languages and databases. He may be reached at cdoskas@ofdcorp.com.

Securing Terminology: Lessons from Interdisciplinary Research

Continued from page 31

grating these ideas into the organizational glossary not only highlights the importance of this effort, but serves to ensure that all stakeholders are using similar terminology.

Communication with the outer ring can be used as an opportunity to interact with the community. Not only can suppliers or customers be engaged in these terminology efforts, but invitations can be extended to university professors, open source project leaders, and other outside experts to present terms and ideas as community service or demonstrations of expertise.

Conclusion

IT security professionals must interact with a wide array of stakeholders. Customers, business managers, project managers, and IT developers all have areas of expertise and activities that can potentially compromise organizational security. Communication between these roles can be complex. Terms and acronyms like "VM programs" can take on different meanings, which can lead to confusion. Compounding the confusion stakeholders may face in individual organizations are the standards setting bodies, academia, industry groups, for-profit organizations, and not-for profit organizations that develop terms and ideas for use within the cyber domain. The array of contexts that need to be addressed is evolving and expanding with cyber technologies. To start bridging potential communication gaps between these stakeholders, interdisciplinary research recommends forming a glossary for a specific project, or in this instance your organization, reinforcing these terms through short presentations, and creating a culture of communication.

References

1. Arredondo, P., Shealy, C., Neale, M., & Winfrey, L. L. (2004). Consultation and interprofessional collaboration: Modeling for the future. *Journal of Clinical Psychology*, 60(7), 787-800.
2. Beckers, K., Côté, I., Fenz, S., Hatebur, D., & Heisel, M. (2014). A structured comparison of security standards. In *Engineering Secure Future Internet Services and Systems* (pp. 1-34). Springer, Cham.
3. Committee on Facilitating Interdisciplinary Research, National Academy of Sciences, & National Academy of Engineering. (1900). *Facilitating Interdisciplinary Research*. National Academies Press.
4. Farrell, J., & Klemperer, P. (2007). Coordination and lock-in: Competition with switching costs and network effects. *Handbook of Industrial Organization*, 3, 1967-2072.
5. Fernandes, B., Rufino, J., Alam, M., & Ferreira, J. (2018). Implementation and analysis of ieee and etsi security standards for vehicular communications. *Mobile Networks and Applications*, 23(3), 469-478.
6. Forman, J., & Markus, M. L. (2005). Research on collaboration, business communication, and technology: Reflections on an interdisciplinary academic collaboration. *The Journal of Business Communication* (1973), 42(1), 78-102.
7. Frühwirth, C. (2009, June). On business-driven it security management and mismatches between security requirements in firms, industry standards and research work. In *International Conference on Product-Focused Software Process Improvement* (pp. 375-385). Springer, Berlin, Heidelberg.
8. Furnell, S. M., Clarke, N., von Solms, R., Tsohou, A., Kokolakis, S., Lambrinoudakis, C., & Gritzalis, S. (2010). A security standards' framework to facilitate best practices' awareness and conformity. *Information Management & Computer Security*.
9. Hiller, J. S., & Russell, R. S. (2013). The challenge and imperative of private sector cybersecurity: An international comparison. *Computer Law & Security Review*, 29(3), 236-245.
10. Hu, Q., Hart, P., & Cooke, D. (2007). The role of external and internal influences on information systems security–a neo-institutional perspective. *The Journal of Strategic Information Systems*, 16(2), 153-172.
11. JASON: Science of Cyber-Security. Technical report, The MITRE Corporation (2010) JSR-10-102
12. Kaufman, L. M. (2009). Data security in the world of cloud computing. *IEEE Security & Privacy*, 7(4), 61-64.
13. Klonoff, D. C., & Kleidermacher, D. N. (2016). Now is the time for a cybersecurity standard for connected diabetes devices.
14. Korsakienė, R., Stankevičienė, A., Šimelytė, A., & Talačkienė, M. (2015). Factors driving turnover and retention of information technology professionals. *Journal of Business Economics and Management*, 16(1), 1-17.
15. Mishra, S., & Dhillon, G. (2006, June). Information systems security governance research: a behavioral perspective. In *1st Annual Symposium on Information Assurance, Academic Track of 9th Annual NYS Cyber Security Conference* (pp. 27-35).
16. Naedele, M. (2003). Standards for XML and Web services security. *Computer*, 36(4), 96-98.
17. Phillips, T., Karygiannis, T., & Kuhn, R. (2005). Security standards for the RFID market. *IEEE Security & Privacy*, 3(6), 85-89.
18. Ryser, L., Halseth, G., & Thien, D. (2009). Strategies and intervening factors influencing student social interaction and experiential learning in an interdisciplinary research team. *Research in Higher Education*, 50(3), 248-267.
19. Shackelford, S. J., Proia, A. A., Martell, B., & Craig, A. N. (2015). Toward a global cybersecurity standard of care: Ex-

ploring the implications of the 2014 NIST cybersecurity framework on shaping reasonable national and international cybersecurity practices. *Tex. Int'l LJ*, 50, 305.

20. Shen, L. (2014). The NIST cybersecurity framework: Overview and potential impacts. *Scitech Lawyer*, 10(4), 16.
21. Siponen, M., & Willison, R. (2009). Information security management standards: Problems and solutions. *Information & Management*, 46(5), 267-270.
22. Susanto, H., Almunawar, M. N., & Tuan, Y. C. (2011). Information security management system standards: A comparative study of the big five. *International Journal of Electrical Computer Sciences IJECSIJENS*, 11(5), 23-29.
23. Wang, Y., Ruan, D., Gu, D., Gao, J., Liu, D., Xu, J., ... & Yang, J. (2011, June). Analysis of smart grid security standards. In *2011 IEEE International Conference on Computer Science and Automation Engineering* (Vol. 4, pp. 697-701). IEEE.
24. Weber, R. H. (2010). Internet of Things–New security and privacy challenges. *Computer Law & Security Review*, 26(1), 23-30.
25. Wright, P. M., & McMahan, G. C. (1992). Theoretical perspectives for strategic human resource management. *Journal of Management*, 18(2), 295-320.
26. Zhou, X., Xu, Z., Wang, L., & Chen, K. (2017, April). What should we do? A structured review of SCADA system cyber security standards. In *2017 4th International Conference on Control, Decision and Information Technologies* (CoDIT) (pp. 0605-0614). IEEE.

About the authors

Dr. Delmer Nagy is an Assistant Professor at Tarleton State University, part of the Texas A&M System. He also used to be an entrepreneur and software developer. His main research interests are in the fields of organizational application of information systems, data security, network security, and project management. He may be reached at Nagy@Tarleton.edu.

Herbert Gomez is pursuing his Bachelors Degree in Computer Information Technology. His main research interests are in the fields of information security, network architecture, e-commerce, and technology project management. He may be reached at Herbert.Gomez@go.Tarleton.edu.

Dr. Christopher Copeland is an Assistant Professor at Tarleton State University, part of the Texas A&M System. His research interests include homeland security, information assurance, digital forensics, and law enforcement training. He may be reached at ccopeland@Tarleton.edu.

Moving Phorward into a New Decade

Continued from page 5

to click on the email I just received saying I've inherited a million dollars!

About the Author

Randy V. Sabett, J.D., CISSP, is an attorney with Cooley LLP (www.cooley.com/rsabett), a member of the advisory boards of the Georgetown Cybersecurity Law Institute and the RSA Selection Committee, and is the former Senior VP of ISSA NOVA. He has completed FBI Citizen Academy training in 2017, was a member of the Commission on Cybersecurity for the 44th Presidency, was named ISSA Professional of the Year for 2013 and an ISSA Distinguished Fellow in 2018, and can be reached at rsabett@cooley.com.

Women Leaders Impacting ISSA

Continued from page 6

en will continue to increase within ISSA to further support and advance the cybersecurity field.

About the Author

Dr. Curtis C. Campbell is VP of Atlantic Capital Bank in Atlanta, GA, and chapter president of ISSA Chattanooga. Curtis holds a PhD in Organizational Leadership in Information Systems Technology. She serves on the advisory board of University of TN-Chattanooga, a National Center for Academic Excellence for Cyber Defense (CAE-CD) studies. Connect with Curtis via curtis@mprotechnologies.com.

DevOps and Infosec

Continued from page 11

Make sure the distributed teams know what you are tracking on an epic and story/task level

- Get to the same language

Infrastructure is an iceberg of risk

- If you are part of the process, your upgrades and updates are not emergencies but part of the normal cycle. Getting this correct knocks off much of your risk metrics as well.

About the Author

Jason Remillard is CEO of Data443 Risk Mitigation, Inc. – A Data Privacy, Governance and Compliance SaaS & Services Provider. He may be reached at jason@data443.com.

ISSA EXECUTIVE CISO FORUM

The Executive CISO Forum is a peer-to-peer event – Members can feel free to share concerns, successes, and feedback in a peer-only environment.

ISSA Executive Membership Program

The role of information security executive continues to be defined and redefined as the integration of business and technology as it evolves. While these new positions gain more authority and responsibility, peers must form a collaborative environment to foster knowledge and influence that will shape the profession.

The Information Systems Security Association (ISSA) recognizes this need and created the exclusive Executive Membership program to give executives an environment to achieve mutual success. Connecting professionals to a large network of peers, valuable information, and top industry experts the program is a functional resource for members to advance personal and industry understanding of critical issues in information security.

Membership Benefits

- Free registration at four Executive Forums per year, including lodging for one nigh and all meals at each Forum
- Extensive networking opportunities with peers and experts on an on-going basis
- Privileged access to online community
- Direct access to top subject matter experts through educational seminars
- An effective forum for understanding and influencing relevant standards and legislation
- A unified voice to influence industry vendors
- Basic Wisegate membership, including exclusive access to the Wisegate community and Executive Forum private group

Visit ISSA.org => Learn => Executive Forum for more information or to register for the Forum.

ISSA Chapters around the Globe

Asia Pacific
Bangladesh
Chennai
Dehradun
India
Philippines

Canada
Alberta
Ottawa
Quebec City
Vancouver

Europe
Brussels European
France
Germany
Italy
Netherlands
Poland
Romania
Spain
Switzerland
Turkey
UK
Ukraine

Latin America
Argentina
Barbados
Bolivia
Brasil
British Virgin Islands
Chile
Colombia
Ecuador
Peru

Middle East
Bahrain
Egypt
Iran
Israel
Kazakhstan
Kuwait
Qatar
Saudi Arabia

USA
Alamo San Antonio
Blue Ridge
Boise
Buffalo Niagara
Capitol Of Texas
Central Alabama
Central Florida
Central Indiana
Central Maryland
Central New York
Central Ohio
Central Plains
Central Texas
Central Virginia
Charleston
Charlotte Metro
Chattanooga
Chicago
Colorado Springs
Columbus
Connecticut
Dayton
Delaware Valley
Denver
Des Moines
East Tennessee
Eastern Idaho
Fayetteville/Fort Bragg
Fort Worth
Grand Rapids
Grand Traverse
Greater Augusta
Greater Cincinnati
Greater Spokane
Hampton Roads
Hawaii
Inland Empire
Kansas City
Kentuckiana
Kern County
Lansing
Las Vegas
Los Angeles
Metro Atlanta
Middle Tennessee
Milwaukee
Minnesota
Montana
Motor City
National Capital
New England
New Hampshire
New Jersey
New York Metro
North Alabama
North Dakota
North Oakland
North Texas
Northeast Florida
Northeast Indiana
Northeast Ohio
Northern Colorado
Northern Virginia
Northwest Arkansas
Northwest Ohio
Oklahoma
Oklahoma City
Orange County
Phoenix
Pittsburgh
Portland
Puerto Rico
Puget Sound (Seattle)
Quantico
Rainier
Raleigh
Rochester, NY
Sacramento Valley
San Diego
San Francisco
SC Upstate SC
Silicon Valley
South Florida
South Texas
Southeast Arizona
Tampa Bay
Tech Valley Of New York
Texas Gulf Coast
Triad of NC
Utah
Ventura County
West Texas
Wyoming
Yorktown

ISSA.org => Chapters => Chapter Directory