World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:18, No:5, 2024

Adding Security Blocks to the DevOps Lifecycle

Andrew John Zeller, Francis Pouatcha

***Abstract*—**Working according to the DevOps principle has gained in popularity over the past decade. While its extension DevSecOps started to include elements of cybersecurity, most real-life projects do not focus risk and security until the later phases of a project as teams are often more familiar with engineering and infrastructure services. To help bridge the gap between security and engineering, this paper will take six building blocks of cybersecurity and apply them to the DevOps approach. After giving a brief overview of the stages in the DevOps lifecycle, the main part discusses to what extent six cybersecurity blocks can be utilized in various stages of the lifecycle. The paper concludes with an outlook on how to stay up to date in the dynamic world of cybersecurity.

***Keywords*—**Information security, data security, cybersecurity, DevOps, IT management.

I. Introduction

IN the past couple of years, the software development approach DevOps (development and operations) has steadily gained popularity within the IT community. Teams are being staffed, and engineers demand to work according to its principles or philosophy. At the same time, cyber incidents all over the world have been rising [1]. Due to some very public hacks, awareness within the development community has increased significantly up to the notion that the DevOps approach be enriched with security elements to evolve into DevSecOps [2]. Although this broader approach is gaining some traction, more needs to be done to add actionable security knowledge to the DevOps approach, so teams – often not trained in security – can easily work with it. Also, from a regulatory point of view, possible security arrangements need to be considered from the beginning on. An example can be found with the South African POPI (Protection of Personal Information) Act, where Section 19 explicitly requires companies to adequately protect an individuals' information against loss, damage, or unlawful access or destruction [3].

The following paper will take six building blocks of cybersecurity and apply them to the DevOps approach. To do so, the paper will first briefly introduce the general lifecycle behind DevOps. Thereafter, the main body will discuss to what extent the cybersecurity blocks can be utilized in what stage of the lifecycle. In terms of security measures NOT applied in a typical DevOps setup, the paper draws on the authors' own experience working with DevOps projects and teams, where not otherwise indicated.

The paper concludes with an outlook on how to stay up to date in the dynamic world of cybersecurity and prevent organizational silos due to specific security expertise.

Andrew John Zeller, PhD is with adorsys Ireland, Ltd., Department Research in Dublin, Ireland (corresponding author, phone: +353 85 253 9575; e-mail: andrew.zeller@adorsys.com).

II. The DevOps Lifecycle

To introduce the reader to the concept of DevOps, this chapter will give a brief overview of the eight stages in the DevOps lifecycle [4].

1. PLAN: Business requirements are collected and the roadmap for the project is derived;
2. CODE: The architectural concept is written and application and supporting systems are built at this stage. Agile processes may be used to deliver the results;
3. BUILD: The various modules of the code stage are combined into a build;
4. TEST: The build is deployed into a test environment where tests can be performed on it;
5. RELEASE: Once all the tests are complete and the development team satisfied with the quality of the build, all elements will be bundled into a release. Environment variables will be reset and the adjustments for the production environment made;
6. DEPLOY: The release will be deployed to the production environment. In modern cloud system, the necessary infrastructure will also be deployed via scripts;
7. OPERATE: Once the application is live for customers, the operations team will manage the run-time and adjust operations parameters and resources according to performance and efficiency goals;
8. MONITOR: The entire environment will be monitored with the help of dashboards and automated alerting tools. Alerts can trigger pre-programmed reactions like automatic provisioning of resources.

For reasons of brevity, this chapter only provided a quick, high-level introduction. The DevOps lifecycle is discussed in more levels of detail by Morales et al. [5].

III. Application of Security Blocks to DevOps

The main chapter will apply security blocks to the above introduced DevOps lifecycle and suggest what teams should look at in particular over the various stages.

A. Risk Management

Assuming the lifecycle begins with its first phase PLAN (which of course is not always the case as the lifecycle oftentimes has been initiated years ago and is already running when you join or when it is reviewed and altered), a rigorous risk management exercise should be included from the first day on. If the team operates in a regulated environment, there should already be a comprehensive risk register in place. Given this, the team will now have to identify all the new risks specific to

Francis Pouatcha, MSc is Global Tech Lead and founder of adorsys Group and with adorsys Ireland, Ltd. in Dublin, Ireland.

their target application or target system, respectively. In addition, cross-references should be made to pre-existing risk determinations as the new application might change the risk landscape and not only add to it. To add expertise and rigor to the exercise, an outside expert could be brought in.

Once the risk exercises have been completed at the PLAN stage, they should be regularly drawn upon and updated throughout the entire lifecycle. This analysis will form the foundation upon which to decide for and prioritize security approaches and tools. In the BUILD stage, architects will use it to decide which security patterns to apply and how to engineer for the level of resilience required. Most importantly, at regular intervals the risk assessment needs to be compared to what risks have indeed crystallized over time.

At this point, industry security frameworks like PCI DSS, NIST CSF, or ISO 27001 also come into play. All of them have in common that they directly link into the existing risk management framework of the company (or at least they should). Based on the CIA principle (confidentiality, integrity, availability), information assets will be classified into various risk categories and prioritized according to their importance for the company. In this respect, typical information assets would be IT systems, processes, suppliers, and facilities. In a DevOps environment, an example for a critical process may be the agile development process and tools used with its integrated development environment (IDE), Github source code repository, CI/CD-pipeline as well as facilities the engineers might have access to while developing.

Also, the location of the company, project, and type of data involved need to be taken into account. Depending on this, services might have to be run in a specific geographical region. Data, for that matter, may have to reside within the boundaries of that area. And backups must not be taken outside of that area.

For all the risks identified, controls will have to be implemented and checked upon throughout the project lifecycle [6].

B. Passwords and Authentication

The ideas in the following chapter apply to the setup and management of the development environment but are especially beneficial in the later stages OPERATE and MONITOR.

Right at the beginning of the project, the team will have to subscribe to a common password policy that needs to be adhered to while in development. If compatible with the policies of the later run-time environment, this policy should also be used for the production environment. If external contractors are working in the environments, making these rules also apply to them should be an absolute priority.

A fundamental concept around Password and Authentication is Identity and Access Management. Both in the project and later in the application, not everyone is supposed to have the same level of access. In fact, according to the least privilege principle, team members and users should only have access to what they need, and nothing more. The necessary access permissions to act in the system are stored in roles. Roles will then be assigned to individuals. The team lead will regularly have to check whether the roles are still up-to-date or need adjusting.

Based on the possibility of hackers cracking a password or getting access to it by means of social engineering, it is a best practice to use multi-factor authentication (MFA) when logging into the any part of the system, especially when doing so as the root user. The development team lead must ensure this feature is enabled throughout the entire project.

In most automated CI/CD-pipelines, API keys are used instead of passwords. Being similar to passwords, they are designed for use by machines [7]. But unlike most human access methods which consist of a public username and a private password, API keys are long random strings. Therefore, they should never be stored and used in plain text and not be hardcoded into any program.

Another relevant approach that the team should consider is segregation of duties [8]. Along the lines of least privilege, different users and accounts will be assigned different rights. This way, the ramifications of a compromised account are limited to the rights assigned. Especially in combination with MFA the segregation of duties approach can be very powerful.

C. Cryptography

Cryptography will play a major role in the first three stages of the DevOps lifecycle: PLAN, CODE, BUILD. It is here where cryptography has to be taken into account the most, as the later stages will not be able to add adequate levels of encryption anymore once coding has passed an advanced point. At the time of writing, a common approach to secure the flow of data through networks is Transport Layer Security. But amongst other limitations, containerized applications, e.g., Docker containers controlled by Kubernetes, are not included in the security offered.

While the usage of data-at-rest encryption, i.e., the encryption of data stored in a data base or data lake, has become normal in production environments, data in transit is oftentimes not so much in focus. The inability to analyze these data whilst passing automatically and easily through network nodes is often the main reason for development teams not doing so. However, with hackers attacking web applications and their backend components inside the defense perimeter of a network, this would mean data in transit are not secure anymore. Therefore, data in transit have to be protected by encryption, too, and at all times.

To enable data in transit encryption, key management will have to be implemented for each of the systems involved and a public/private keypair as well as a signed certificate issued. The challenge here is always the secrecy of the private key. In a cloud-based environment like AWS, this can be done with tools like the AWS Certificate Manager in combination with the AWS Key Management Service. A more detailed discussion of cryptographic algorithms can be found at NIST [9].

D. Network and Application Security

This section applies primarily to the later stages of the DevOps lifecycle OPERATE and MONITOR. Most certainly, the architects will have to integrate these scenarios in the BUILD phase.

For ease of programming, development environments sometimes neglect security standards that apply to production environments. This leaves the final release prone to security gaps and should be avoided. API calls may serve as a point in case here. In the very early stages of development, calls are frequently mocked, before the real code is added. To avoid any potential vulnerability later on, the requesting system should apply a high security setting from the beginning on. Hence, the requester system can consider using bearer token-based access like OAUTH 2.0, the Open Authorization framework and standard [10].

Along the same lines, Network Access Control lists and firewalls need to be setup right from the beginning, instead of allowing all traffic to flow freely.

A typical attack that needs to be prepared for is the Distributed-Denial-of-Service attack. The objective of this attack is to overwhelm an application or a network, respectively, and bring the running system to a halt. Were it only for a single attacking machine, a firefall could easily block the IP address the attack is coming from. To avoid this, distributed attacks use a network of bots (hijacked computers) to confuse the defending system as to where the attack originates from. Therefore, architects will have to use various tools to mitigate this type of attack, e.g., load balancers, deployment to multiple regions, Content Delivery Networks that shield the IP addresses of the real applications, as well as network traffic scanners that can alert a monitoring system once unusual traffic patterns are detected.

According to the above introduced CIA principle, data read by applications should be checked upon concerning its integrity. Often, applications simply assume the data they rely on to be correct. Here, a 'Trust, but verify' culture could be applied [11]. The general idea is to trust the underlying data sources, but the system be architected to be able to undergo a regular self-audit and throw an error in case of a breach of integrity.

E. Business Continuity and Disaster Recovery

When disaster strikes, the application should be still able to run and serve the business. Rather than wait for a crisis, organizations have to expect them [12]. In the DevOps lifecycle, the following thoughts will need to be included when the initial architecture is being derived, but also once the application is up and running in the OPERATIONS stage.

One possible solution to securing business continuity here is to run the application on multiple instances in at least two geographically distant locations. Again, depending on the business needs, the architects will have to make the decision how to manage the failover scenario. In a hot link, the other instance can take over immediately and traffic will be routed to the backup system. In a warm or even cold scenario, the period of switching to the backup takes successively longer.

Every change made to the production and development environments should be stored in a trackable record. This is not only necessary for potential rollbacks, but also for remedying setups after attacks. Especially in a cloud-based environment, the CI/CD pipelines should be completely scripted, in order to automatically set up the system without the dangers of manual misconfigurations. AWS Config may be used to keep historical records of configurations and then have automated scripts use these snapshots to recreate the system including volume and database encryptions.

In a lot of cases, when the systems are being spun up, machine images are used to ensure infrastructure setup can be scripted. During this critical phase, a common attack vector would be to inject malware into the scripts used. As a consequence, images and scripts need to be protected or checked that they have not been tampered with. One way of doing this is to hash the file values and then check each time before the images or scripts are run. The hash will return a unique value for the input file and any changes to the source would immediately result in a different value. So, in case possible intruders have altered any images to make sure their malware remains in the systems even after relaunch of the system, a hash value will reveal the differences.

F. Penetration Testing

As the name already implies, penetration testing can be used primarily at the test stage of DevOps. Drawing on the risk management activities at the beginning of the life cycle, the development team should have a good understanding of possible vulnerabilities at this point. In addition, the top ENISA vulnerabilities should be included in the approach [13], where relevant.

Before commencing any testing, it is absolutely vital to get adequate authorization from management and where appropriate, from third parties. For example, the latter are relevant when working in a cloud environment or with a separate data center. Teams must be made aware that improper permissions to test may even result in running the danger of committing an offence under section 2 of the Criminal Justice Act 2017, Offences Relating to Information Systems [14].

Mostly due to cost and complexity, DevOps teams sometimes tend to security test the environment themselves or have an enterprise security team test it via a ticketing service [15]. This very much resembles a white-box test. This, of course, does not speak against conducting any internal security test. On the contrary, each sprint should include some standard tests to ensure a basic degree of security of the system by identifying any gaps as early as possible in the lifecycle.

However, for a full-scale security, test the point of view of a potential attacker should be included. This can be done best by bringing in specialized contractors to do this specific job. With internal testers, the teams might tend to go the easy way and attack the known vulnerabilities, which of course will have been protected by the time of the test. An outside team, or at least a separate team, will have to start from scratch and therefore has a higher chance to find potential gaps. When working with outside teams, a proper contact structure including a master service agreement and individual statements of work is recommended.

Depending on the industry the project is in, it might even be required to have outside parties conduct the penetration testing. In the banking, the PCI DSS standard requires participants to

have the tests done by an independent third party [16]. The process and results must be fully documented and implications discussed.

After testing has concluded, project management must request a detailed walk-through of the approach taken and the findings. This should be done by the provider of the penetration testing job. The results should also be communicated and explained to the sponsor of the DevOps project, as penetration test findings are usually followed by budget discussions.

IV. Conclusion

This paper presented selected security elements that should be taken into account when working on DevOps projects. Based on the nature of this paper, this can only count as a starter and there is much room for more research. Most certainly, this approach needs to be operationalized to help teams better work with it. A simple but straightforward idea would be to create an extended checklist every project manager may use.

At the same time, in the very volatile and fast-developing world of IT security tools and technologies, the company's knowledge must be kept up to date at all times. To do so, an organization could establish an information strategy based on three pillars:

1. Security news: For all the markets a company has operations in, it is best practice to subscribe to relevant security newsgroups and publications. Once subscribed, organizational ownership should be assigned while constantly monitoring these channels;
2. Courses and certifications: To maintain a broad knowledge-base, team members will have to regularly attend university courses and undergo relevant certifications. For the Republic of Ireland, a good overview has been provided by Carrol [17]. This can also be used to prove existing knowledge to partners and customers. At the same time, certifications can also help companies with attracting and retaining much sought-after IT security talent;
3. Industry groups: An interesting opportunity may arise through inter-company collaboration in industry groups. Latest IT security information and intelligence may be shared as well as the latest international trends in how to organize security structures, teams, and tools.

With all these skills added to the DevOps teams, another challenge may evolve. Knowhow and tooling expertise accumulated may become so specific and also hard to acquire in the market, that the team develops into a silo [18]. This organizational anti-pattern must be prevented. Therefore, the security skills mentioned will have to be bundled into repeatable internal learning blocks that new recruits can be walked through and, at best, as stated in the above information pillars, receive formal certification in.

References

[1] J. Boehm, D. Dias, C. Lewis, K. Li, and D. Wallance, *Cybersecurity trends: Looking over the horizon*. 2022, Available at: https://www.mckinsey.com/capabilities/risk-and-resilience/our-insights/cybersecurity/cybersecurity-trends-looking-over-the-horizon (Downloaded: 04 November 2022).

[2] S. Comella-Dorda, J. Kaplan, L. Lau, and N. McNamara, N., *Agile, reliable, secure, compliant IT: Fulfilling the promise of DevSecOps*, 2022. Available at: https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/agile-reliable-secure-compliant-it-fulfilling-the-promise-of-devsecops (Downloaded: 19 October 2022).

[3] South Africa Government, Protection of Personal Information Act. Available at: https://popia.co.za/ (Downloaded: 19 February 2023).

[4] H. Dhaduk, *DevOps Lifecycle: 7 Phases Explained in Detail with Examples.* The Simform blog, 13 January 2022. Available at: https://www.simform.com/blog/devops-lifecycle/ (Accessed: 19 October 2022).

[5] J. Morales, R. Turner, S. Miller, P. Capell, P. Place, and D.J. Shepard, *Guide to Implementing DevSecOps for a System of Systems in Highly Regulated Environments*. 2020, Carnegie Mellon University. Available at: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=638576 (Downloaded 19 October 2022).

[6] NIST National Institute of Standards and Technology, NIST Special Publication 800-115. 2021, Available at https://www.nist.gov/privacy-framework/nist-sp-800-115 (Downloaded 15 September 2022).

[7] C. Dotson, *Practical Cloud Security. A Guide for Secure Design and Deployment*. 2019, 1st edn. Sebastopol, CA: O'Reilly Media, pp. 60-65.

[8] L. Rice, *Container Security. Fundamental Technology Concepts that Protect Containerized Applications*. 2020 ,1st edn. Sebastopol, CA: O'Reilly Media, pp. 11-20.

[9] NIST National Institute of Standards and Technology, *Transitioning the Use of Cryptographic Algorithms and Key Lengths*. 2019, Available at https://csrc.nist.gov/publications/detail/sp/800-131a/rev-2/final (Downloaded 19 September 2022).

[10] J. Richer, A. Sanso, *OAuth 2 in Action*. 2017, 1st edn. Shelter Island, NY: Manning.

[11] M. Kleppmann, *Designing Data-Intensive Applications. The Big Ideas Behind Reliable, Scalable, and Maintainable Systems*. 2017, 1st edn. Sebastopol, CA: O'Reilly Media, pp. 530-545.

[12] D. Telem, K. Sadek, H. Nijjar, and D. Knott, *Crisis Management & Business Continuity Guide*. KPMG. 2020, Available at: https://assets.kpmg/content/dam/kpmg/ca/pdf/2020/03/cyber-resilience-crisis-business-continuity-planning-en.pdf (Downloaded: 19 October 2022).

[13] European Union Agency for Cybersecurity, *ENISA Threat Landscape.* 2021. Available at: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2021 (Downloaded 16 August 2022).

[14] International Comparative Legal Guides, *Cybersecurity Laws and Regulations Report 2022 Ireland*. 2022, Available at: https://iclg.com/practice-areas/cybersecurity-laws-and-regulations/ireland (Accessed 04 November 2022).

[15] McKinsey and Company, *Cybersecurity in the Digital Era*. 2022, Available at: https://www.mckinsey.com/~/media/McKinsey/Business%20Functions/Risk/Our%20Insights/Cybersecurity%20in%20a%20digital%20era/Cybersecurity%20in%20a%20Digital%20Era.pdf (Downloaded 02 September 2022).

[16] PCI Penetration Test Guidance Special Interest Group Security Standards Council, *Penetration Testing Guidance*. 2017, Available at: https://listings.pcisecuritystandards.org/documents/Penetration-Testing-Guidance-v1_1.pdf (Downloaded 19 October 2022).

[17] J. Carroll, *Cybersecurity Training and Education in Ireland – Where do I start?*, Fortify Institute Blog, 12 June 2022. Available at: https://www.fortifyinstitute.com/blog/cybersecurity-training (Accessed 05 November 2022).

[18] M. Skelton, M. Pais, *Team Topologies. Organizing Business and Technology Teams for Fast Growth*. 2019, 1st edn. Portland, OR: IT Revolution, p. 76.