DevSecOps: A Crucial Approach in Preventing and Responding to Worst-Case Cyberattacks like the SolarWinds Hack

Hasti Mohebali Zadeh
hastimz@kth.se

Omid Hazara
hazara@kth.se

April 10, 2023

Code of Conduct

We certify that generative AI, incl. ChatGPT, has not been used to write this essay. Using generative AI without permission is considered academic misconduct.

1 Introduction

Despite the fact that companies invest heavily to mitigate cyberattacks, security breaches occur from time to time with different levels of outreach. It is said that since 2000 more than 3.5 billion people have experienced personal data theft [1]. One company that was a victim of hackers was Solarwinds in the united states in 2020, where a group of hackers could penetrate not only the company's network but also the customers' personal computers. The hack's scope was unheard of and ranks as one of the biggest—if not the biggest—of its kind ever documented. The hackers' attack vector was the automated software update deployment with the "This release includes bug fixes, increased stability and performance improvements." as the description of the update [2].

DevOps utilization of security methods has apparently not shown a good track record by leaving the cross-examination of the code to the end of the development phase, therefore the need for a more robust security measure has been experienced. As a result, modern developers are required to incorporate security safeguards throughout the entire development process. In the context of DevOps security, this approach is known as DevSecOps.

In this paper, we try to shed light on the ways in which adopting the DevSecOps framework can enhance organizational resilience in the face of security breaches, using the SolarWinds incident as a case study.

2 DevSecOps Framework

DevSecOps framework involves incorporating security measures within a continuous integration and continuous delivery/deployment workflow. It integrates security into the software development lifecycle (SDLC) and the DevOps workflow by introducing security at each stage of the pipeline, rather than considering security as an afterthought. This methodology requires the entire DevOps team to share responsibility for following security best practices and enables automated security checks at every stage of software delivery. Hence, DevSecOps involves continuous integration, continuous delivery/deployment, continuous feedback, and continuous operations [3].

The framework has 6 phases as shown in Figure 1: Plan, Build, Test, Deploy, Operate and Monitor. **The planning** phase includes activities such as collaboration, discussion or dialog, review, and security examination and analysis. In **the Build** phase, the security analysis is applied through automated tools such as Static Application Software Testing (SAST) and unit tests. **The Testing** phase utilizes Dynamic Application Security Testing (DAST) tools to detect flows in the live application.

During **The Deployment** phase configuration differences should be reviewed and a key principle employed in this phase is the Principle of the Least privilege (PoLP). This principle is the key concern of the release phase which requires that any user, program, or process to have a minimum level of access to perform its function. This concludes that API keys and access token owners have limited access and that they are audited. Without this audit, an attacker may find a key that has access to unintended areas of the system which is very serious. **The Operate and Monitor phases** include monitoring the live application through Runtime Application Self-Protection (RASP), security monitoring, penetration testing, and bug bounty programs [1].

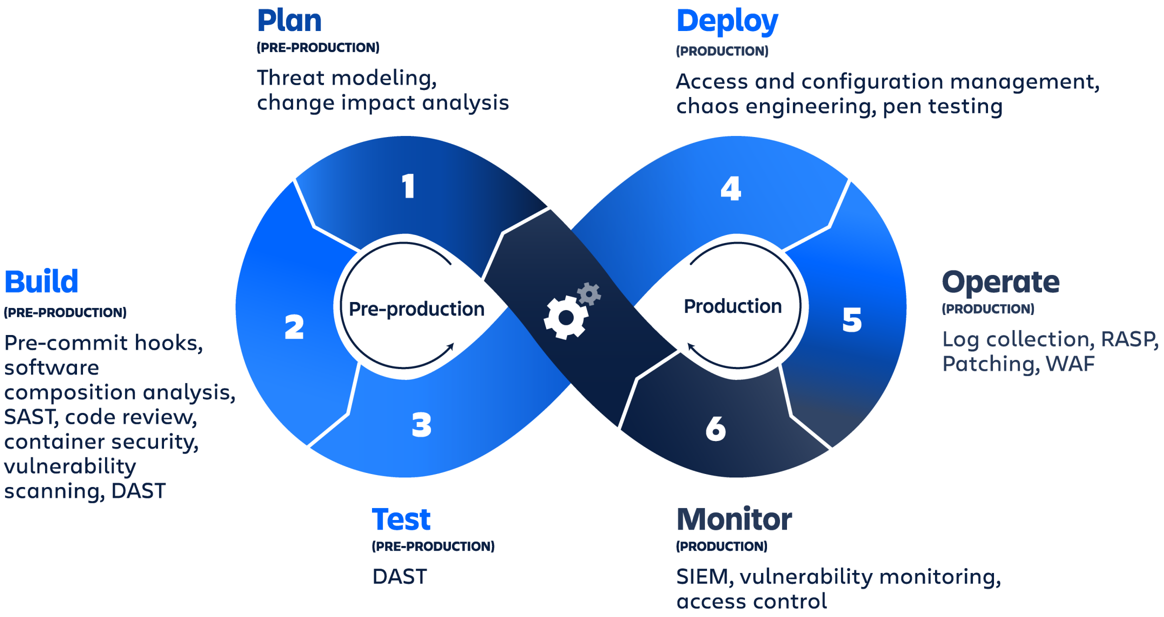

Plan
(PRE-PRODUCTION)
Threat modeling, change impact analysis
Deploy
(PRODUCTION)
Access and configuration management, chaos engineering, pen testing
Build
(PRE-PRODUCTION)
Pre-commit hooks, software composition analysis, SAST, code review, container security, vulnerability scanning, DAST
Operate
(PRODUCTION)
Log collection, RASP, Patching, WAF
Test
(PRE-PRODUCTION)
DAST
Monitor
(PRODUCTION)
SIEM, vulnerability monitoring, access control
1
2
3
4
5
6
Pre-production
Production

Figure 1: DevSecOps Framework [1]

3 DevSecOps Vs. Cyberattacks

How DevSecOps can prevent cyberattacks: As mentioned earlier, generally by implementing security practices throughout the software development lifecycle, DevSecOps tries to minimize the potential risk of being exposed to cybersecurity breaches and thus preventing cyberattacks as result. For example, Automated security testing and vulnerability scanning tools facilitate early awareness in case of vulnerability and safety issues. Another measure is threat modeling which involves designing and performing attacks on your own system early on during the planning phase or development cycle to identify how and where an attack is most likely to occur. STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege) are common threat modeling techniques that development teams can utilize to identify potential threats and assess the risks involved. This provides good insight for designing secure software architecture and proper implementation from the start [4, 5].

In order for the security measures to become reality, there needs to be a focus on collaboration between development, security, and operations teams in a DevSecOps environment. Teams need to meet and share knowledge, so that the gap between them is marginalized and a security-aware culture is fostered. This way, the developers know about the secure coding practices and security teams know about the new features and potential vulnerabilities [6].

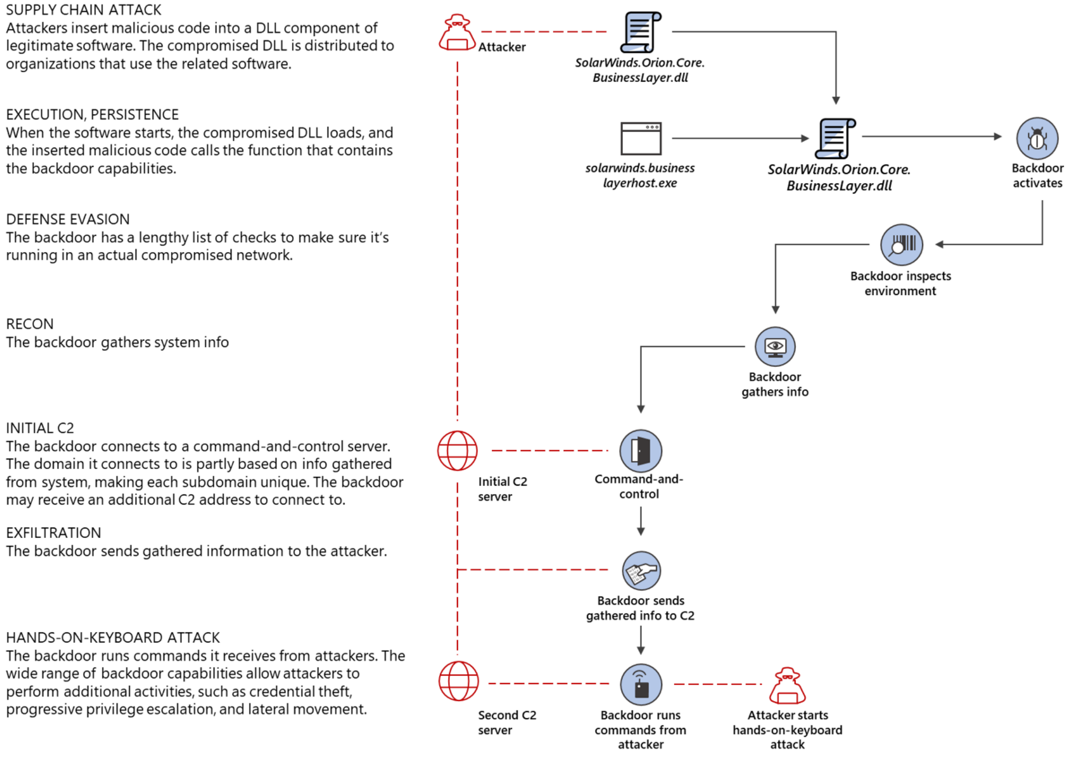
How DevSecOps can respond to cyberattacks: Real-time monitoring and logging in DevSecOps involves the implementation of tools and systems such as Security Information and Event Management

(SIEM) system to be able to detect security incidents in real-time. For example, if an authorized user attempts to login or gain access to a resource, SIEM system will detect this intrusion and trigger an automated and predefined response, such as blacklisting the IP address or notifying the security teams [4, 7]. Measures for mitigating the damages and prevention of future attacks can include ”Post-Mortem” analysis to identify the root causes, understanding of security breach(es) and implementation of patches to prevent future attacks.

4 Case Study: SolarWinds Hack

SolarWinds[1] is a software company that is based in Oklahoma, USA and provides system management tools for network and infrastructure monitoring. This means that SolarWinds Orion, as an IT monitoring system, can obtain log and system performance data. The term “SolarWinds hack” refers to the supply chain breach that involved the SolarWinds Orion system. In this incident, suspected hackers (that are believed to be from a foreign government) gained access to the networks, systems, and data of thousands of SolarWinds customers. Microsoft has identified this group as ”Nobelium”.

As explained in Figure 2, the hackers used the supply chain attack which works by targeting a third party with access to an organization’s systems (which in this case is the SolarWinds Orion Platform) instead of directly hacking the networks. Since this platform was used by many global companies and government agencies, the hackers only had to insert their code into a new section of software (a legitimate SolarWinds DLL file) which was distributed by the company as an update. They managed to stay undetected for 15 months by using techniques like obfuscation, time-based triggers, and mimicking legitimate network traffic. Finally, the hack was detected by FireEye which is one of Solarwinds’ customers who subsequently reported to SolarWinds and Microsoft [8, 9].



SUPPLY CHAIN ATTACK
Attackers insert malicious code into a DLL component of legitimate software. The compromised DLL is distributed to organizations that use the related software.

EXECUTION, PERSISTENCE
When the software starts, the compromised DLL loads, and the inserted malicious code calls the function that contains the backdoor capabilities.

DEFENSE EVASION
The backdoor has a lengthy list of checks to make sure it’s running in an actual compromised network.

RECON
The backdoor gathers system info

INITIAL C2
The backdoor connects to a command-and-control server. The domain it connects to is partly based on info gathered from system, making each subdomain unique. The backdoor may receive an additional C2 address to connect to.

EXFILTRATION
The backdoor sends gathered information to the attacker.

HANDS-ON-KEYBOARD ATTACK
The backdoor runs commands it receives from attackers. The wide range of backdoor capabilities allow attackers to perform additional activities, such as credential theft, progressive privilege escalation, and lateral movement.

Attacker
SolarWinds.Orion.Core.BusinessLayer.dll
solarwinds.businesslayerhost.exe
SolarWinds.Orion.Core.BusinessLayer.dll
Backdoor activates
Backdoor inspects environment
Backdoor gathers info
Initial C2 server
Command-and-control
Backdoor sends gathered info to C2
Second C2 server
Backdoor runs commands from attacker
Attacker starts hands-on-keyboard attack

Figure 2: SolarWinds Hack [9]

[1] https://www.solarwinds.com/

However, there are several ways that DevSecOps could have prevented or mitigated this attack. With platforms that have high popularity, there is often an intense focus on the speed of delivery which can affect security. While DevSecOps principles try to preserve this speed, there are some low-effort but high-impact security actions that need to happen in the development processes such as peer reviews of pull requests.

The attackers' code was likely inserted in a file that was not often viewed by developers, otherwise, they would have caught it while making a later change to the file. If other developers had reviewed the pull request, they would have noticed that the HTTP connections to a Command-and-Control server were malicious [10]. Another way that this incident could have been prevented is the effective use of SAST tools as they are designed to detect backdoor code which would have resulted in a warning on the vulnerability. While it is not likely that SolarWinds completely neglected using SAST tools, it is possible that developers did not prioritize SAST results. Hence, it is important to leverage SAST capabilities by following its proper processes [10].

5 Challenges in Adopting DevSecOps

While implementing the DevSecOps approach in an organization has many benefits, it can be challenging as well. Here are some common challenges that companies might face while adopting this approach [11, 12, 6]:

1. **Lack of knowledge and resources:** DevSecOps practices need a set of skills and adequate working knowledge that not all organizations have. Training the current developers or finding new employees with the required expertise can be expensive and time-consuming. Also, security and operations teams might not be very familiar with software development environments.

2. **Budget limitations:** Implementing this approach needs an investment in many things such as employees, training and perhaps tools. This problem can be even more significant for smaller companies with tight budgets. Also, DevOps stakeholders consider security as a barrier to delivering the product to the market fast.

3. **Integration and compatibility problems:** As the name suggests, DevSecOps requires close collaboration between development, security, and operations teams. It can be challenging to make sure that tools are integrated, continue to work effectively and are compatible with the techniques used in DevOps process.

However, there are also strategies to overcome these challenges and resistance in trying to adapt to DevSecOps such as [12]:

1. **Having education and training:** Companies and organizations can invest in training developers' technical and soft skills in order to successfully implement DevSecOps.

2. **Having a solid business case:** Organizations can tell the stakeholders about both benefits and risks associated with implementing this approach and provide a potent business case that illustrates the possible cost savings and improved results of DevSecOps in order to soften their view of this approach.

3. **Taking small steps:** In order to address the integration problem, companies do not need to implement DevSecOps all at once. Instead, they can begin with a smaller pilot project and highlight the benefits it brings.

Overall, these strategies can help to prevent or respond to worst-case cyberattacks such as the SolarWinds hack and strengthen the security of systems.

6 Reflection

Based on all the previous research done in this area, it becomes clear that the security knowledge of developers in software development and collaboration between the teams are far more important than the tools that DevSecOps provide. As an example, in the delivery process of a web application, everyone needs to communicate with each other effectively and be familiar with the basics of application security such as the top 10 Open Web Application Security Project (OWASP). Lack of knowledge, in our opinion, is one of the contributing factors as to why security breaches exist and hacks happen.

The SolarWinds hack mentioned in this essay serves as a sobering example that shows what can happen when security measures are not strong enough. This event became a wake-up call for many organizations and governments that were affected by the breach, leading them to reactively improve their security practices.

It is crucial to acknowledge that the deployment of DevSecOps is not a one-time solution, but rather a continuous process that requires continuous improvement and adjustment to emerging threats and vulnerabilities as they are discovered. In light of the continuous refinement of tactics and exploitation of newly discovered vulnerabilities by cybercriminals, organizations must sustain agility in order to respond and maintain the integrity of their systems. Regularly checking and updating security methods is very important for staying ahead of possible dangers. By keeping up with the newest information and best practices to handle threats, organizations can protect themselves better. Using a flexible and repeating process in DevSecOps helps businesses to guard their important things and stay strong against ever-changing online risks and threats, this becomes even more important with the emergence of stronger technologies where cyberattackers can gain technological superiority.

7 Conclusion

In this essay, we have described what the DevSecOps framework is and how its adoption can enhance organizational resilience in the face of security breaches. We also discussed the SolarWinds incident in detail and presented the challenges organizations may face when considering the adoption of DevSecOps, providing points for them to ponder.

In conclusion, while the adoption of DevSecOps may present certain challenges, the substantial advantages it offers far outweigh them. By including enhanced security, the likelihood of security breaches is minimized, and better resilience toward cyberattacks is achieved, making DevSecOps a crucial strategy for organizations in today's rapidly evolving cybersecurity landscape.

References

[1] Kev Zettler. *DevSecOps Tools: The DevSecOp tools that secure DevOps workflows.* Atlassian, 2023. URL: https://www.atlassian.com/devops/devops-tools/devsecops-tools.

[2] Dina Temple-Raston. *A 'Worst Nightmare' Cyberattack: The Untold Story Of The SolarWinds Hack.* Apr. 2021. URL: https://www.npr.org/2021/04/16/985439655/a-worst-nightmare-cyberattack-the-untold-story-of-the-solarwinds-hack.

[3] Akanksha Gupta. "An Integrated Framework for DevSecOps Adoption". In: *International Journal of Computer Trends and Technology* 70.6 (July 2022), pp. 19–23. DOI: 10.14445/22312803/ijctt-v70i6p102. URL: https://doi.org/10.14445%5C%2F22312803%5C%2Fijctt-v70i6p102.

[4] Håvard Myrbakken and Ricardo Colomo-Palacios. "DevSecOps: A Multivocal Literature Review". In: *Software Process Improvement and Capability Determination.* Ed. by Antonia Mas et al. Cham: Springer International Publishing, 2017, pp. 17–29. ISBN: 978-3-319-67383-7.

[5] Tony Hsiang-Chih Hsu. *Hands-On Security in DevOps: Ensure continuous security, deployment, and delivery with DevSecOps.* Birmingham, UK: Packt Publishing Ltd, 2018.

[6] Vaishnavi Mohan and Lotfi Ben Othmane. "SecDevOps: Is It a Marketing Buzzword? - Mapping Research on Security in DevOps". In: *2016 11th International Conference on Availability, Reliability and Security (ARES).* 2016, pp. 542–547. DOI: 10.1109/ARES.2016.92.

[7] SANS Institute. *A DevSecOps Playbook.* Tech. rep. 2018. URL: https://dsimg.ubm-us.net/envelope/384733/450933/sans-a-devsecops-playbook.pdf.

[8] Sean Michael Kerner Saheed Oladimeji. *SolarWinds hack explained: Everything you need to know.* June 2022. URL: https://www.techtarget.com/whatis/feature/SolarWinds-hack-explained-Everything-you-need-to-know.

[9] Tom Smith and team. *Analyzing Solorigate, the compromised DLL file that started a sophisticated cyberattack, and how Microsoft Defender helps protect.* Microsoft. Dec. 2020. URL: https://www.microsoft.com/en-us/security/blog/2020/12/18/analyzing-solorigate-the-compromised-dll-file-that-started-a-sophisticated-cyberattack-and-how-microsoft-defender-helps-protect/.

[10] Kyle McNulty. *Five DevSecOps Lessons from the SolarWinds Orion Attack.* blog.focal-point.com, Dec. 2020. URL: https://blog.focal-point.com/devsecops-lessons-from-the-solarwinds-orion-attack.

[11] Mahesh Nawale. *The Top Challenges of DevSecOps Implementation.* Zscaler, May 2022. URL: https://www.zscaler.com/blogs/product-insights/top-challenges-faced-organizations-implementing-devsecops.

[12] Roshan N. Rajapakse et al. "Challenges and solutions when adopting DevSecOps: A systematic review". In: *Information and Software Technology* 141 (2022), p. 106700. ISSN: 0950-5849. DOI: https://doi.org/10.1016/j.infsof.2021.106700. URL: https://www.sciencedirect.com/science/article/pii/S0950584921001543.