Improving Security of DevOps

Akshay Rane
arane1@stevens.edu
Stevens Institute of Technology
Hoboken, New Jersey

Revathy Ramasundaram
rramasun@stevens.edu
Stevens Institute of Technology
Hoboken, New Jersey

Rahil Patel
rahilpatel12@gmail.com
Stevens Institute of Technology
Hoboken, New Jersey

Yuanxin Liu
yliu250@stevens.edu
Stevens Institute of Technology
Hoboken, New Jersey

Yifan
luyifan@bu.edu
Stevens Institute of Technology
Hoboken, New Jersey

ABSTRACT

DevOps quickens the application development cycle by shifting everything to the left of the application life-cycle. This shift compromises the security aspects of the application as it requires a skilled team, and not all testers can code, and not all coders can administer the Operating system. DevSecOps merge security into the DevOps process, integrating into every step of the process. The challenge for the development team is to be a jack of all trades. The focus of DevSecOps is automating the entire Software Developement LifeCycle process with security built into each step.

KEYWORDS

DevSecOps

1 INTRODUCTION

"Dev" refers to software application development, and "Ops" refers to IT operations. Defining what DevOps engineers do is still in flux because DevOps is not a framework or a workflow. It's a culture that is overtaking the business world. DevOps is a concept with different interpretations and definitions, but when you get down to it, it's all about developers and operations teams breaking down silos and working together to innovate faster. For many companies, the ability to innovate at a rapid pace — responding to market conditions and customer feedback is a key factor for success. Adopting DevOps philosophy requires a new mindset, tools, and skills. DevOps integrates developers and operations teams in order to improve collaboration and productivity by automating infrastructure, workflow, and continuously measuring application performance.[1]

There are four fundamental core values in DevOps: Culture, Automation, Measurement, and Sharing. There are five levels of DevOps practice: values, principles, methods, practices, and tools. DevOps can be implemented in 3 phases:

1.1 Automated Testing

Agile method and automated testing are the foundation of DevOps competency. This involved writing tests within the code so that every change in the code can be evaluated by whether the test failed or not.[1]

1.2 Continuous Integration

Once we have effective code coverage for testing, the entire testing process is then automated. Jenkins is one of the most popular tools used to implement continuous integration. The concept is based on having a number of servers in the background testing the code and running it through every iteration possible in order to find out if it can create any bug in an automated way. Jenkins will then generate a report at the end of that process (such as what is the code coverage, whether testing failed or passed, etc.)[1]

1.3 Continuous Delivery

This phase is where we get real business value. The base consists of writing code in small chunks (new features, bug fixes, etc.) that are integrated, tested, monitored and deployed. The continuous delivery pipeline and tools for each organization is different. The idea of a pipeline is a series of phases, each backed by a specific tool. Let's look at the six key phases of continuous delivery and the tooling that's associated with it.[1]

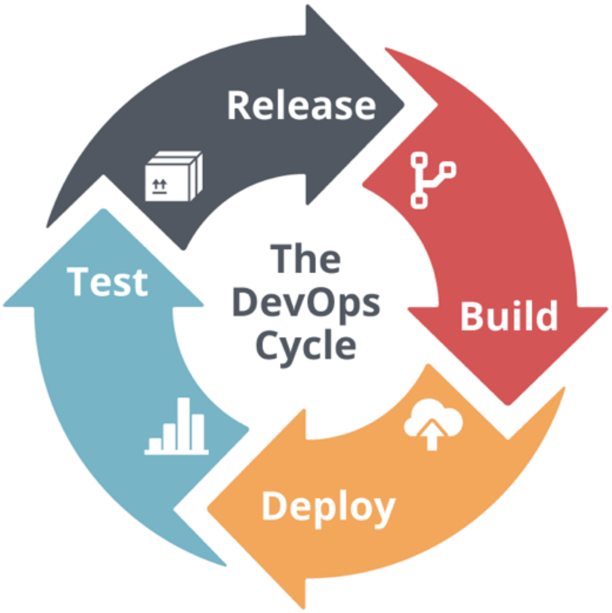


Figure 1: The DevOps cycle

1.4 How does a DevOps team accomplish its tasks

- Tools that allow the team to build and test code continuously such as Jenkins (monitor the execution of repeated jobs and integrate project changes more easily and access output for quickly identifying problems)[1]
- Tools for source control that allow the team to manage, track and document all the changes to the application code and configuration management code such as Git, GitHub, SVN, etc. (version control system for tracking changes in computer files and coordinating workflows among multiple teams and used for management in software development)[1]
- Tool for configuration management code (code deployment) that allows the team to deploy new code in an automated fashion maybe across several servers in different locations: Chef, Puppet, SaltStack, etc.[1]

2 SECURITY IN DEVOPS

DevOps security refers to the discipline and practice of safeguarding the entire DevOps environment through strategies, policies, processes, and technology. Security should be built into every part of the DevOps lifecycle, including inception, design, build, test, release, support, maintenance, and beyond. Today, this type of "baked-in" DevOps security is often called DevSecOps, which aims to improve security through improved collaboration and shared the responsibility that overlays the entire DevOps workflow.

A normal security testing of an application happens in the production environment. The DevOps team does it in the Development or the QA environment. The steps involved in the normal security scan are as follows:

- Develop the Code.
- Send code to security.
- Scan the code.

- Send PDF of the result(Done by the Security team).
- Find false-positive.
- Fix security defects.

This is a manual process and takes 1-2 weeks for completion.
Disadvantages:

- Too slow.
- Too many people involved.
- No empowerment of Dev teams.

2.1 Purpose

- Merging of security into DevOps process.
- Further the goal of giving Dev teams more ownership.
- Integrate automated security into every step of the process.

3 NEED FOR SECURITY

3.1 Code Should Be Secured

A total of 44% of developers cannot code securely. That's nearly half! Developers need to work with security teams to scan code for malicious content, constantly. Malicious code can be injected at any time in the building process. Why would a hacker wait for a finished product to hack, when they could add a small line of code during development that acts as a back door? But only developers who know what to look for during the development process will be able to combat this potential threat.[2]

Patterns for design to help developers write safe code should be set for all members of the DevOps team. This includes any and all code that will keep applications secure for the user and the network. Preventing abuse can only happen if DevOps and Security work together.[2]

3.2 Security Should Be in Every Stage of the Deployment Pipeline

Continuous deployment pipelines open doors to a larger area of attack to include your production system, the build, testing, and deployment environment. Because of this, security should be implemented at every stage of your deployment pipeline. This is the only way to secure your pipeline from outside attacks as well as insider attacks. Security measures can be used to ensure all changes are transparent and completely traceable. The only way to ensure code and applications remain untouched by unauthorized hands is to infuse security into the pipeline, from start to finish.[2]

DevOps and security together will bring about the best outcome for your enterprise. From protecting your code to ensure your deployment pipeline is safe, you will be able to move just as fast with security as you thought you could without it. It's time we embrace DevOps and security together, instead of keeping them apart.[2]

3.3 Benefits of a DevSecOps Approach

Security protocols that are baked into the development process rather than added as a "layer on top" allows DevOps and security professionals to harness the power of agile methodologies—together as a team—without short-circuiting the goal of creating secure code. A 2017 EMA report found the top two benefits of security operations (SecOps): better ROI in existing security infrastructure and improved operational efficiencies across security and the rest of IT.[2]

Another top benefit identified in the study was the ability to make full use of cloud services. For example, organizations running services in the Amazon Web Services (AWS) cloud reap the benefits of increased preventive and detective security controls within the continuous integration and deployment model of AWS. As more organizations rely on cloud applications to keep operations up and running, security efforts independent of those performed by AWS are crucial to prevent costly down-times.[2]

The safety measures inherent in DevSecOps has many other advantages. These include:

- Greater speed and agility for security teams
- An ability to respond to change and needs rapidly
- Better collaboration and communication among teams
- More opportunities for automated builds and quality assurance testing
- Early identification of vulnerabilities in code
- Team member assets are freed to work on high-value work

4 SECURITY ATTACKS

4.1 Security attacks on DevOps

A cybersecurity attack is a threat to the assets of an organization, exploiting vulnerabilities in applications. The assets could be credit card information, health records, employee salary information, etc depending on the organization. Open Web Application Security Project (OWASP) has identified SQL injection, sensitive data exposure, security misconfigurations, Cross-Site Scripting (XSS), insecure deserialization, and insufficient logging and monitoring as some of the most seen vulnerabilities. These vulnerabilities can be addressed, and the threats can be minimized by incorporating DevSecOps into the Software Development Life-cycle.

In this paper we will discuss SQL injection attacks, and Cross-Site Scripting (XSS) attacks, as they have been listed as the top 2 attacks vulnerabilities in web applications.

4.2 Exploiting Vulnerabilities

4.2.1 SQL Injection Attacks.

Many of the servers that store critical information for websites and services, use SQL to manage the data in their databases. A SQL injection attack targets this kind of a server to trick them into divulging sensitive information, which normally they wouldn't. The following query was the most used for this attack [12]:

Query	**Incidents**	**Goal of the attack**
1 and 1=2 union select password from qianbo_admin	634,566	Trying to query passwords

Executing this query results in returning all the rows from the table. A malicious user can exploit this to get all the information from the table that they are not authorized to read. The vulnerabilities exploited in this attack are lack of validation and sanitization of the input consumed by the web application.

4.2.2 How to prevent SQL injection attacks?

- Always, prepared statements and parameterized queries should be preferred, user-provided input should not be placed directly in the query.
- Confidential information stored in the database should be encrypted.
- A Web Application Firewall should be used for web-facing applications.

4.2.3 Real Incidents of SQL injection attacks in 2019:

- Medical Informatics Engineering - 3.5 million patient records exposed[13]
- Indian government bus booking site - complete database exposed[13]

4.2.4 Cross-Site Scripting Attack.

Cross-site scripting (XSS) is a code injection security attack targeting web applications which deliver malicious, client-side scripts to a user's web browser for execution. Targets are not attacked directly, but vulnerable websites and web applications are used to carry out cross-site scripting attacks when users interact with these sites/applications.

An unsuspecting user will visit a compromised website, at which point the attacker's malicious script is loaded and executed by the user's browser. JavaScript has been a popular choice for XSS attack authors, but an attack can be crafted with any language that is supported by browsers. While XSS attacks have been around for over 15 years, they've proven to be highly effective and are still frequently observed as a common and viable attack vector.

The code below is an example of a reflected XSS where an attacker could craft up a URL that passes a small, malicious script as a query parameter to a website that has a search page vulnerable to XSS:

http://vulnerable-website.com/search?search_term="<script>(bad things happen here)</script>"

When a target clicks the link, the vulnerable site accepts the query parameter "search_term", expecting that the value is something the target is interested in searching the vulnerable-website.com site for, when in reality the value is the malicious script. Since the vulnerable site didn't sanitize the search_term value, the malicious script is injected into the web-page that the target's browser is loading and is then executed by the target's browser.

This may lead to serious consequences such as theft of sensitive data, session hijacking, vandalism of website contents, redirecting websites to untrusted locations, and upload of malicious Trojan horse programs.

4.2.5 *How to prevent XSS attacks?*

- User-provided data should be sanitized before executed by the web server
- Enforce the Content Security Policy (CSP) - web servers should be configured to use the header Content-Security-Policy, and HTTPS instead of HTTP.
- Regularly using a web application vulnerability scanning tools such as Grabber, Vega, Zed Attack Proxy.

4.2.6 *Real Incidents of XSS attacks.*

- Magecart (2018) - a credit card skimming malware introduced by XSS, massive amounts of credit card information stolen
- The Samy attack (2005) - Within 20 hours, over a million users had fallen victim for the vulnerability

4.3 CloudBleed - A case study on security attack

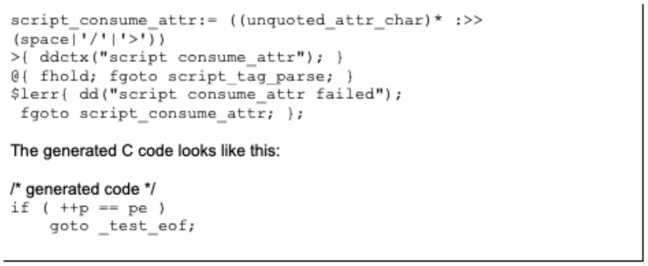
CloudFlare is a web infrastructure and website security company providing services like DDoS mitigation, Content Delivery Network services, Internet security, and Distributed Domain Name Server services. Over 20 million companies on the Internet use the services provided by CloudFlare. CloudBleed is a security attack that exposed sensitive information such as credit card details, cookies, chat messages, encryption keys, hotel booking and more on the CloudFlare network.

4.3.1 *Technical details of the attack.*

Many of Cloudflare's services rely on parsing and modifying the incoming HTML pages as they pass through their edge servers. For example, they insert the Google Analytics tag, safely rewrite http:// links to https://, exclude parts of a page from bad bots, obfuscate email addresses, enable AMP, and more by modifying the HTML of a page.

A software called Ragel was used to modify the page, that reads and parses the HTML to find elements that need changing. A single '. rl' file contains an HTML parser used for all the on-the-fly HTML modifications that Cloudflare performs. The issue was not on the

Ragel software itself, but on how CloudFlare developers used it. The following code is the root cause:



```
script_consume_attr:= ((unquoted_attr_char)* :>>
(space|'/'|'>'))
>{ ddctx("script consume_attr"); }
@{ fhold; fgoto script_tag_parse; }
$lerr{ dd("script consume_attr failed");
 fgoto script_consume_attr; };
```

The generated C code looks like this:

```
/* generated code */
if ( ++p == pe )
    goto _test_eof;
```

Figure 2: Root cause CloudFlare

From the code, web pages that contain broken HTML tags flow to the '$lerr' block where the pointer is incremented without checking for boundary conditions. The underlying issue was a buffer overrun that exposed all the confidential information unencrypted.

4.3.2 *Key Takeaways.*

There are 2 key takeaways from this incident:

– Removing broken HTML tags from websites - nearly 0.06% of websites have unenclosed HTML tags

– Proper unit testing - this buffer overflow error could have been easily avoided at the unit test level

It is essential for an organization to use a mix of security tools available to effectively safeguard their assets and save their reputation. Security has to be a part right from unit testing, integration testing, CI/CD pipelines, to vulnerability scanning in websites. In the next part of this paper, we will discuss the different tools that can be used as part of the DevSecOps pipeline.

5 CYBERSECURITY VENDORS

5.1 Who are cybersecurity vendors?

5.1.1 *The types of individuals and groups involved in cyber defense.*

The types of individuals and groups involved in cyber defense can be categorized into five groups with widely varying roles, motivations, and responsibilities. These groups each play an important role in reducing the risk of cyberattacks to valued assets, albeit with different levels of authority, ownership, skill and legal protections. One of the groups I mentioned above includes the cybersecurity technology vendors who produce products and services that stop cyber attacks.[14]

5.1.2 *Cybersecurity technology vendors.*

With the development of network technology, enterprises use network technology to conduct business, and profits have become more and more extensive. While enjoying the convenience brought by the network, the problem of network security risks has become increasingly prominent. In this circumstance, it is especially important to fortify the cyber defense. Today, the cost of developing a company's own cybersecurity system is quite high and the cost of maintaining a full-time cybersecurity department is still too high. Outsourcing your cybersecurity can provide a cost-effective solution that is managed by experts and customized to your unique needs and priorities. Therefore, cybersecurity vendors are favored by more and more companies. Serving essentially as defensive arms dealers, this industry has grown considerably in the past few years, and many small, medium, and large vendors exist around the world to help reduce risk.[14]

5.1.3 How cybersecurity vendors reduce risk? In simple terms, the vendor firstly defines what you need by an in-depth review of your company's IT infrastructure, the data you're protecting, and the risks that threaten it, then provide an appropriate solution for your company.[14]

5.2 How to choose the best vendor?

The cybersecurity industry is flooded with vendors—more than 1,200 to be specific. With so many vendors offering seemingly the same products and services, it's not always easy to choose the best fit to secure your assets.[14]

Below are some considerations and questions to ask yourself and your vendor when looking for a partner:

Define what you need: Your search should start close to home, with an in-depth review of your company's IT infrastructure, the data you're protecting, and the risks that threaten it. Only by thoroughly understanding your own cybersecurity needs can you identify a vendor who is equipped to meet them.[14]

Align your goals: Your cybersecurity strategy should support your business protecting the data that is most critical to your success against the threats that are most likely to compromise it. No single security solution will work for all organizations. To find vendors whose products and capabilities are a good fit for your particular needs, ask who they've worked within your industry, and how they've solved some of the problems that you face.[14]

Seek appropriate solutions: The cybersecurity industry is growing explosively, creating a dizzying array of new companies, products, and capabilities. A good security partner will work hard to understand your organization's needs and find the technologies that will work best for your environment.[14]

When choosing the right vendor, be sure to ask the following questions:[14]

- How much will it cost?
- What will it do for my organization?
- What benchmarks do you use to measure effectiveness?
- How will you protect the security of my organization's data?
- How will you assist in meeting compliance and reporting requirements?

5.3 Cloud security vendor

Storing data in the cloud is becoming increasingly popular as organizations recognize its benefits. As a result, cloud security is becoming equally popular. Sure, not every company needs to outsource cloud security, but a growing number find that it makes sense given some noticeable and powerful trends in the industry. Consider this:[15]

There's not enough talent to go around. The IT research firm (ISC)2 estimates that the worldwide shortage of cybersecurity professionals now approaches 3 million—half of that in North America. So, companies may not be able to access the talent they require without tapping outside vendors.[15]

Cyber vulnerabilities are always changing, and keeping up is important. As businesses adapt to a rapidly changing technological environment that enables e-commerce, mobile payments, cloud computing, Big Data and analytic, IoT, AI, machine learning, and social media, cybercriminals adapt too, finding new vulnerabilities in emerging technologies. Hiring a vendor gives your company a range of cybersecurity experts who are monitoring for and can quickly understand new and evolving threats. Having an in-house team of professionals might mean they are being pulled a million different directions and not able to keep up with threats as well as a dedicated vendor team could.[15]

Outsourced cybersecurity is cost-effective. In 2018, the average data breach cost nearly $3.9 million, according to the Poneman Institute. That's a large exposure, and yet many companies balk at the expense of maintaining 24/7 protection. A cybersecurity partner can provide the same level of security, if not more, at an affordable cost.[15]

5.4 Vendors doing DevOps security Security Compass

5.4.1 *Company profile.*

Security Compass is a leader in helping businesses proactively make their software secure and reduce the risk of cybersecurity breaches. Offering advisory services, training, and SD Elements, the leading Application Security Requirements and Threat Modeling (ASRTM) platform, Security Compass enables development teams to rapidly and efficiently deliver software that's secure by default. Security Compass serves some of the world's largest businesses including seven of the 15 largest financial institutions and four of the 10 largest technology companies in North America. The privately held company is headquartered in Toronto, Canada with global offices in the United States and India.[16]

5.4.2 *How Security Compass improve the security of DevOps?*

As the practice of DevOps unifies development and operations to release and maintain software with greater efficiency, AppSec and OpSec unite in SD Elements to inject security earlier into the software development lifecycle (SDLC). Where AppSec protects applications from attack and privacy breaches, OpSec protects operations using monitoring, ongoing management, security analysis, and risk assessment. Together, organizations can leverage the efficiencies of DevOps without sacrificing security.[16]
Key features and functionality of the SD Elements OpSec extension include:[16]

- SD Elements now secures the production environments of applications, also known as the "configure and deploy" stage of the DevOps cycle.
- SD Elements can be used to manage the security requirements of the deployment configuration settings alongside the requirements for the application itself to achieve DevSecOps.
- The SD Elements platform will immediately feature industry-standard benchmarks for securing application deployments on Amazon Web Services from the Center for Internet Security®.

"Agile development teams embrace DevOps to bring products to market faster – often skipping important security measures," said Rohit Sethi, COO of Security Compass. "SD Elements makes it easy for DevOps teams to manage the security considerations of the entire technology stack – both the software itself, as well as the operational security requirements of the Web server, application server, database server, and operating system that hosts the application. These production-environment capabilities, combined with our existing AppSec and just-in-time application security training, enable agile organizations to achieve a continuous and holistic software security program to better manage risk and protect sensitive data."[16]

6 KIUWAN

Kiuwan was released in 2012, by the firm Optimyth. Its software offers solutions for application security. Kiuwan has grown from 200 companies in 25 countries to 500 companies worldwide since it joined the idea family of DevOps tools in Oct 2018. According to its own website, Kiuwan provides service that "blocks attacks on vulnerabilities with a plethora of features that empower both executives and developers." So basically, on their website, there are two produces for choosing, which are Application Security Testing (SAST) and Software Composition Analysis (SCA). Those applications can scan users' code locally and then share results in the cloud. A tailored report will be generated according to the result of scanning code which helps users reduce technical debt and mitigate risk. But in order to get know-how Kiuwan works better, we need to understand one more key concept, which is pen-testing.

6.1 How Kiuwan works and What is pen-testing

As we learned in CS573, there is one important advantage of virtualization for cybersecurity is that it provides a “test box” in cloud service that makes every possible virus run in that test box first before entering the main process. A similar concept applies to pentesting as well. Pentesting is also known as penetration testing. A successful Pentesting will let cybersecurity protectors find out the weaknesses and backdoors that may threaten the security of the entire company. It’s one of the main tools that kiuwan use to protect the client’s benefit and let’s take a closer look at how it works.

6.2 How pentesting works

A penetration tester will “ethically hack” into the system by testing attacks on the system in order to find out weaknesses in security. Those testers will perform reconnaissance to find some vulnerabilities, exploit those vulnerabilities to gain access and then possibly extract some small piece of data of value to prove that the system is not secure. It’s important to notice that pentesting is not a one-time security test that will pass or fail forever. Pentester can always upgrade the program by putting up a patch after each pentest. It’s an ongoing process and that why pentesting is important and powerful in real-life use.

6.3 How to use Kiuwan

Using pen testing is just one of their tools to find out application weaknesses. The complete solution integrates with leading IDEs, build systems, bug tracking tools, and repositories to detect and eliminate vulnerabilities and provide full compliance with security standards at a significantly lower price than competitors. Although those process sounds complicated and seems usually take a long period of time to achieve, Kiuwan makes the process easy and quick in use. Users can simply download kiuwan in local pc, (for now only support in Mac, window, Cloud, Saas, and web. Not support mobile device in neither iOS and android) and scan the code. After successfully scanning code, the user can either analyze code locally by downing the Kiuwan local analyzer or analyze source code in the cloud by uploading source code on kiuwan’s could server. Click analyze and wait for results to be automatically published in the user’s kiuwan account.

6.4 Forecast of Kiuwan

In the future, Artificial intelligence will play a bigger role in cybersecurity, and as the leader in cybersecurity, Kiuwan will for sure work on that part. AI software can automatically detect and block threats before it can cause any damage. However, on the other hand, AI and machine learning can also improve hacker’s attack level. Therefore, Kiuwan will for sure works on improving its artificial intelligence technology against new threats in the future. Secondly, security spending will increase rapidly. According to Kiuwan’s own website, “In fact, it is estimated that businesses worldwide spent $114 billion on cybersecurity measures in 2018, and it is estimated that this number will likely increase to $124 billion in 2019.” Therefore, the stage for Kiuwan will for sure be bigger in the future. And in the end, without a doubt, a threat to cybersecurity will also increase. Since people are relying more on the internet, network crime has more profit space. Therefore more and more cybercriminals will appear with more advanced technologies in the future.

7 SONARQUBE

SonarQube is the leading tool for continuously inspecting the Code Quality and Security of the code-bases and guiding development teams during Code Reviews. SonarQube is a web-based open source platform used to measure and analyse the source code quality. SonarQube is written in java but it can analyze and manage code of more than 20 programming languages, including c/c++, PL/SQL, Cobol etc through plugins. SonarQube is maintained by SonarSource.[9]

SonarQube receives files as an input and analyzes them along with barriers. Then calculates a set of metrics, stores them in a database and shows them on a dashboard. This recursive implementation helps in analysis of code quality and how code improves over time.[9]

Products: Sonarlint - an IDE extension (free and open source), Sonar cloud - a software as a service with APIs, Sonarqube - an open-source platform

7.1 Security Tools

One of the most important goals of DevSecOps is to shift the security to the left. The leftmost section of the application life-cycle is the development phase. DevSecOps' primary goal is to empower the development team to be responsible for security by working with the Security team as the accountable partners as everything needs to happen quickly. The security team provides the Frameworks, Tools, and Expertise for security.

7.1.1 Static application security testing(SAST).

- Analyzes source code for security check.
- Supports one or more languages.
- Tends to have very high false positive rate.
- Products:
  - **Commercial:** Fortify, AppScan, Checkmarx
  - **Open Source:** FindSecBugs, Brakeman, PMD

7.1.2 Dynamic application security testing(DAST).

- Runs automated penetration scans.
- Tries to hack the website.
- Scans take a long time to complete.
- Products:
  - **Commercial:** WebInspect, Burp, AppSpider
  - **Open Source:** ZAP

7.1.3 Interactive application security testing(IAST).

- Security tests happen while app is used.
- Works using instrumentation.(See data going in and out from requests. Also able to see SQL queries.)
- Low false positive rate and immediate feedback.
- Products:
  - **Commercial:** Contrast, Seeker
  - **Open Source:** N/A

7.2 Static Code Analysis

Static code analysis is a collection of algorithms and techniques used to analyze source code in order to automatically find potential errors or poor coding practices. Static code analysis, also commonly called "white-box" testing, looks at applications in non-runtime environments. It is also considered as a way to automate code review process.[10]

The tasks solved by static code analysis software can be divided into 3 categories:[10]

- Detecting errors in programs.
- Recommendations on code formatting. Some static analyzers allow you to check if the source code corresponds to the code formatting standard accepted in your company.
- Metrics computation. Software metrics are a measure that lets you get a numerical value of some property of software or its specifications.

7.3 Features

- SonarQube doesn't just show you what's wrong. It also offers quality-management tools to actively help you put it right[10]
- SonarQube's commercial competitors seem to focus their definition of quality mainly on bugs and complexity, whereas SonarQube's offerings span what its creators call the Seven Axes of Quality[10]
- SonarQube addresses not just bugs but also coding rules, test coverage, duplications, API

documentation, complexity, and architecture, providing all these details in a dashboard[10]

- It gives you a moment-in-time snapshot of your code quality today, as well as trends of lagging (what's already gone wrong) and leading (what's likely to go wrong in the future) quality indicators[10]
- It provides you with metrics to help you take right decision. In nearly every industry, serious leaders track metrics. Whether it's manufacturing defects and waste, sales and revenue, or baseball hits and RBIs, there are metrics that tell you how you're doing: if you're doing well overall, or whether you're getting better or worse.[10]

7.4 Code Security Guidelines

The SonarQube Quality Model has three different types of rules: Reliability (bug), Vulnerability (security), and Maintainability (code smell) rules. But divided another way, there are only two types: security rules, and all the rest. The distinction between these two groups is not so much in what they catch but in where they come from and in the standards imposed on them.[8]

The standard for most rules implemented in SonarQube language plugins is very strict: no false positives. A lot of security guidelines talk about how "sensitive" data should be handled (e.g. not logged, not stored un-encrypted, etc.). It is not possible to tell in a rule which data is sensitive and which isn't, therefore, the preference is given to the no-false-positives standard. The idea is that the rule will flag anything suspicious and leave it to the human security auditor to cull the false positives and sent the real issues for remediation.[8]

7.4.1 *Vulnerabilities.*

- CWE(Common Weakness Enumeration)
  Common Weakness Enumeration is a formal list or dictionary of common software weaknesses that can occur in software's architecture, design, code or implementation that can lead to exploitable security vulnerabilities.
  The CWE is a hierarchy of weakness descriptions. The lowest level in the hierarchy is a "Weakness Base", which describes a granular weakness. Above Weakness Bases, are Weakness Classes and Categories. In general, rules are linked to Weakness Bases or Classes.[8]
- SANS Top 25
  The SANS Top 25 list is a collection of the 25-most dangerous errors listed in the CWE, as compiled by the SANS organization. The current SANS list is divided into three categories: Insecure Interaction Between Components, Risky Resource Management, and Porous Defenses.[8]
  The tags used for SANS correspond to its categories: sans-top25-insecure, sans-top25-risky, sans-top25-porous.[8]
- OWASP Top 10
  The OWASP Top 10 is a list of broad categories of weaknesses, each of which can map to many individual rules.[8]
  The tags used for OWASP correspond to the weakness categories: owasp-a1, owasp-a2, owasp-a3, owasp-a4, owasp-a5, owasp-a6, owasp-a7, owasp-a8, owasp-a9, owasp-a10.[8]

7.4.2 *Security Hotspots.*

Security Hotspots aren't necessarily issues that are open to attack. Instead, Security Hotspots highlight security-sensitive pieces of code that need to be manually reviewed. Upon review, you'll either find a Vulnerability that needs to be fixed or that there is no threat.[8]

Security Hotspots help focus the efforts of developers who are manually checking security-sensitive code. Reviewing Security Hotspots allows you to:[8]

- Fix security issues – Reviewing Security Hotspots gives you the opportunity to detect vulnerabilities and ensure issues are fixed before merging pull requests or releasing your branch.[8]

- Learn about security – SonarQube explains why your code was identified as a Security Hotspot and the link between your Security Hotspots and well-known attacks or weaknesses such as SQL Injection, Weak Cryptography, or Authentication. This helps you to know when you're working on security-sensitive code and to avoid creating Vulnerabilities.[8]

7.5 Security Mechanism

SonarQube comes with a number of global security features:

- on-board authentication and authorization mechanisms[8]
- the ability to force users to authenticate before they can see any part of a SonarQube instance[8]
- the ability to delegate to authentication[8]

Additionally, you can configure at a group or user level who can:

- see that a project even exists[8]
- access a project's source code[8]
- administer a project (set exclusion patterns, tune plugin configuration for that project, etc.)[8]
- administer Quality Profiles, Quality Gates, and the SonarQube instance itself.[8]

Another aspect of security is the encryption of settings such as passwords. SonarQube provides a built-in mechanism to encrypt settings.[8]

7.5.1 Authentication.

The first question that should be answered when setting the security strategy for SonarQube is: Can anonymous users browse the SonarQube instance or is authentication be required? To force user authentication, log in as a system administrator, set the Force user authentication property to true in the Security settings under Administration Configuration.[8]

- **Authentication Mechanisms**
  Authentication can be managed through a number of mechanisms:[8]
  * Via the SonarQube built-in users/groups database
  * Via external identity providers such as an LDAP server (including LDAP Service of Active Directory), GitHub etc.
  * Via HTTP headers

7.5.2 Authorization.

The way authorization is implemented in SonarQube is pretty standard. It is possible to create as many users and groups of users as needed. The users can then be attached (or not) to (multiple) groups. Groups and / or users are then given (multiple) permissions. The permissions grant access to projects, services and functionalities.[8]

Group: A group is a set of users. Two groups have a special meaning:[8]

- Anyone is a group that exists in the system, but that cannot be managed. Every user belongs to this group, including Anonymous user.
- sonar-users is the default group to which users are automatically added.

7.5.3 Permission Templates for Default Permissions.

SonarQube ships with a default permissions template, which automatically grants specific permissions to certain groups when a project, portfolio, or application is created. It is possible to edit this template, and to create additional templates. A separate template can be set for each type of resource.[8]

- **Project Permissions** Project visibility may be toggled between public or private. Making a project private hides its source code and measures from the Anyone group. For both public and private projects, four different permissions can be set:[8]
  - **Administer Issues:** Change the type and severity of issues, resolve issues as being

"Won't Fix" or "False Positive" (users also need "Browse" permission).[8]

- **Administer Security Hotspots:** With Security Hotspots, you can Open as a Vulnerability, Set as In Review, Resolve as Reviewed. With a Security Hotspot that's been opened as a Vulnerability, you can Reset as To Review or Resolve as Reviewed.[8]
- **Administer:** Access project settings and perform administration tasks (users also need "Browse" permission).[8]
- **Execute Analysis:** Execute analyses (project, view, report, developer), and to get all settings required to perform the analysis, even the secured ones like the Scm account password, the Jira account password, and so on.[8]

Private projects have two additional permissions:

- **Browse:** Access a project, browse its measures, issues and perform some issue edits (confirm/resolve/reopen, assignment, comment).[8]
- **See Source Code:** View the project's source code.[8]

7.5.4 Settings Encryption.

Encryption is mostly used to remove clear passwords from settings (database or SCM credentials for instance). The implemented solution is based on a symmetric key algorithm. The key point is that the secret key is stored in a secured file on disk. This file must owned by and readable only by the system account that runs the SonarQube server. The algorithm is AES 128 bits.[8]

8 CONCLUSION

DevSecOps focuses on integrating security with DevOps. This makes security agile and simple as the whole security setup is configured for all environments, and from the initial stages. The security team plays an important role. Security experts are advisors and coaches and turn the security checks into code. DevSecOps focuses on vulnerability scanning (related to Operating System), Network security, Automated patching compliance, and Encryption. It secures the application from potential attacks through the usage of automation tools.

REFERENCES

[1] Introduction to DevOps
https://blog.usejournal.com/what-is-devops-in-simple-english-6550fbb129bd

[2] DevSecOps vs Rugged DevOps
https://www.sumologic.com/insight/devsecops-rugged-devops/

[3] Best Practices for Security in DevOps
https://www.beyondtrust.com/blog/entry/devops-security-best-practices/

[4] Best Practices for Security in DevOps
https://www.beyondtrust.com/blog/entry/devops-security-best-practices

[5] Equifax Cyber Attack
https://www.calyptix.com/top-threats/biggest-cyber-attacks-2017-happened/

[6] Uber Cyber Attacks
https://www.wired.com/story/uber-paid-off-hackers-to-hide-a-57-million-user-data-breach/

[7] Code Analysis at Facebook and Google
https://techbeacon.com/devops/how-google-facebook-do-code-analysis

[8] SonarQube
https://www.sonarqube.org/

[9] Benefits of SonarQube
https://vizteck.com/blog/benefits-using-sonarqube/

[10] Why SonarQube?
https://dzone.com/articles/why-sonarqube-1

[11] Types of Attacks
https://www.rapid7.com/fundamentals/types-of-attacks/

[12] SQL Injection Attacks
https://securityboulevard.com/2019/06/sql-injection-attacks-so-old-but-still-so-relevant-heres-why-charts/

[13] SQL Injection
https://codecurmudgeon.com/wp/sql-injection-hall-of-shame/

[14] From CIA to APT An Introduction to Cyber Security, Edward Amoroso Matthew Amoroso Page14

[15] Cybersecurity vendor landscape
https://www.armor.com/blog/finding-your-fit-understanding-the-cybersecurity-vendor-landscape/

[16] Security Compass https://devops.com/security-compass-extends-devops-support-adding-software-operational-security-coverage-sd-elements-platform/

[17] Kiuwan Team. (2019) https://www.kiuwan.com

[18] Tom Mowatt, Managing Director, Tools4ever (2019). Kiuwan's application security testing platform helps teams realize DevSecOps goals