Secuirty Orchestration, Automation & Response

Submitted By

Jenil Sadrani

18MCEI10

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2020

Secuirty Orchestration, Automation & Response

Major Project

Submitted in fulfillment of the requirements

for the degree of

Master of Technology in Computer Science & Engineering

(Information & Network Security)

Submitted By

Jenil Sadrani

(18MCEI10)

Guided By

Dr. Sanjay Garg

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2020

Certificate

This is to certify that the Major Project entitled **"Secuirty Orchestration, Automation & Response"** submitted by **Jenil Sadrani (Roll No: 18MCEI10)**, towards the fulfillment of the requirements for the award of degree of Master of Technology in Computer Science & Engineering (Information & Network Security) of Nirma University, Ahmedabad, is the record of work carried out by her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project part-I and part-II, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Dr. Sanjay Garg
Guide & Associate Professor,
CE Department,
Institute of Technology,
Nirma University, Ahmedabad

Dr. Sharada Valiveti
Coordinator M.Tech - INS,
CE Department
Institute of Technology,
Nirma University, Ahmedabad

Dr. Madhuri Bhavsar
Professor and Head,
CE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. R N Patel
I/C Director,
Institute of Technology,
Nirma University, Ahmedabad

Statement of Originality

I, **Jenil Sadrani, 18MCEI10**, give undertaking that the Major Project entitled "**Security Orchestration, Automation & Response**" submitted by me, towards the fulfillment of the requirements for the degree of Master of Technology in Computer Science & Engineering (Information & Network Security) of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made.It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

_____

Signature of Student
Date: 18 May, 2018
Place: Ahmedabad

Endorsed by
Dr. Sanjay Garg
(Signature of Guide)

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Dr. Sanjay Garg**, Associate Professor, Information Technology Department, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement throughout this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr. Madhuri Bhavsar**, Hon'ble Head of Computer Science and Engineering Department, Institute of Technology, Nirma University, Ahmedabad for her kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr Alka Mahajan**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation she has extended throughout course of this work.

I would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

- Jenil Sadrani

18MCEI10

Abstract

Application security has become an inseparable part of Information Security. Now a days organizations develop application for internal use as well as for selling it to the customers. These applications store a huge amount of data which would be of utter importance to the organization or an individual. Securing this information is important as most of the applications would be accessible by the means of internet. There are various Enterprise tools that help detect vulnerability in the application. One of the method used to detect vulnerability is Static Application Security Testing. Here the source code of the application is scanned and based on this, the vulnerability would be detected. Another method is Dynamic Application Security Testing. Here the tool behaves as an attacker and performs analysis to find out the vulnerability. These vulnerabilities should be resolved in order to make the application and in turn the organization's data secure. Every organization has a policy defined in order to make sure that the information that an application processes is secure. For this purpose SOAR tool comes handy. Whenever an application goes on the non-compliant side of the policy, certain steps are required to be taken in order to make them compliant with the organization's policy. If human efforts are used for this it would be very time consuming task and also less efficient. This is where SOAR comes into place. SOAR monitors application scan results and sends out automatic alerts to the concerned set of people if the application is not compliant with the policies defined in the tool. Thus in order to make sure the applications in an organization are compliant, SOAR is used.

Abbreviations

SAST	Static Application Security Testing
DAST	Dynamic Application Security Testing
XSS	Cross Site Scripting
OWASP	Open Web Application Security Project
SQL	Structured Query Language
NIST	National Institute of Standards and Technology
SOAR	Security Orchestration, Automation & Response
GRC	Governence, Risk and Compliance

Contents

List of Figures

Chapter 1

Introduction

1.1 Application Security

Application Security has become an important part of the bigger picture of Information Security. It is an essential part of any organization that deals with developing applications and use it internally or develop them to sell in the market. The amount of information that the application stores are increasing in a significant amount due to which securing this information is one of the major concerns of the organization. This is where Application Security comes into play a vital role. The Application Security helps you secure applications throughout the development and maintenance of the code. It helps to secure the code from known vulnerabilities like SQL Injections, Cross-Site Scripting, Cross-Site request forgery, etc which causes theft of information that might be of utter importance to the organization or an individual. Securing applications is becoming more and more important as they are being hosted on the internet and are accessible to the entire world.

Whenever a developer/development team is not aware about how to develop the application securely, it causes security holes in the system. The attackers take advantage of this hole and get into the system and steal the important data/information. To make sure these security holes do not exist in an application different types of security testing methods are performed.

Figure 1.1 shows various various stages of web application security testing methodology like information gathering, planning analysis, Vulnerability Detection, Penetration Testing And Reporting.

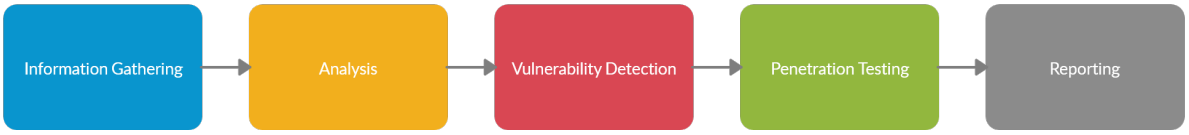

Figure 1.1: Application Security Testing Methodology

1.1.1 Identification Categories of Vulnerabilities

Whenever an application goes through the process of security testing, various vulnerabilities would be detected by the enterprise tool as well as would be detected manually. As and when the tools prompts alert identifying a vulnerability, there are chances where the prompt may not be an actual vulnerability. Below are some of the scenarios:

- **True Positive:** Which means tool has detected vulnerability when the vulnerability is present. Basically it will classify insecure system as a insecure.

	Vulnerability Detected	Vulnerability Not Detected
Vulnerability Verified	True Positive	False Positive
Vulnerability Absent	False Negative	True Negative

Figure 1.2: Test Statistics

- **True Negative:** Which means tool did not detect vulnerability when the vulnerability was absent. Basically it will classify secure system as secure.
- **False Positives:** Which means the tool has detected the vulnerability even if there is no vulnerability in the system. Basically it will classify secure system as insecure.

- **False Negatives:** Which means the tool did not detect the vulnerability when it as actually present. Basically it was identify the system as secure even if it is not secure.

1.2 Motivation

Nowadays every organization has a large number of applications running internally. It is a tough task to make sure each and every application is compliant to the Information Security Policies. Monitoring them manually and notifying people related to the development of that application is difficult and inefficient. SOAR tool will help make sure all the sources are monitored and if any condition leads to non-compliance, actions would be taken automatically so that the remediation can be made quicker.

1.3 Objective

The objective of SOAR is to monitor different sources like SAST Tool, DAST Tool, etc. and make sure the applications are compliant to the Information Security Policy of an organization. The polices are to be incorporated into the tool and made sure all the applications within the organization are compliant to the policies. This will reduce the human efforts and fasten the process of remediating the vulnerabilities in an application. Once the system finds the vulnerability, based on the severity of the application and business importance of the application, alerts are sent to the team who developed the application and the Application Security Analyst. Based on the confirmation further steps are taken. If the application stays non-compliant for more than a specified time period according to the policy, response plan gets activated and further actions are taken.

1.4 Scope of Work

There are 4 different stages in SOAR.

- **Monitor:** This phase monitors different sources such as SAST Tool, DAST Tool, Network Devices, etc. It monitors for vulnerabilities and the severity of the vulnerability.
- **Report:** Based on the severity of the application and business importance of the application, alerts with different priorities are sent to a specific set of people.
- **Analyze:** Once the alert is sent, the Application Security Specialist would analyze the vulnerabilities and come to a conclusion. If the vulnerability is True Positive, an alert with a higher priority is sent and further steps are to be taken.
- **Response:** After all the alerts are sent, and after a specified period, if the threat/vulnerability is not yet resolved, the incident response plan is activated.

SOAR helps the Application Security Team to work efficiently and find the applications that are non-compliant and also make sure the vulnerabilities are analyzed properly. Also SOAR helps the development team know the issue with their application so that they can fix this as soon as possible. It identifies the Risk it causes to the organization and based on that initiate the response plan in order to make sure there is no further threat to the information and the organization.

Chapter 2

Methods of Application Security Testing

2.1 Static Application Security Testing (SAST)

SAST is also known as White-Box testing. Here the source code is reviewed by different enterprise tools for finding out different kinds of vulnerabilities for different languages used to build the application. This kind of testing is performed when the application is in the static state.

SAST tools provide the team with real-time feedback and help them see the issues in the source code. Based on the results, the development team can modify the code do the necessary changes. The users can generate various reports based on the requirement and keep the track of the issues reported by the tool.

2.1.1 Key Steps to run SAST successfully:

- Finalize the tool based on the requirement
- Create the required environment for the tool to run
- Customize the tool based on the needs
- Analyze the scans
- Review the vulnerabilities and generate the report

2.2 Dynamic Application Security Testing(DAST)

Dynamic Application Security Testing (DAST) is a method that actively investigates the applications that are live and running with penetration tests to detect possible security vulnerabilities.

DAST is also known as Black-Box testing. Here the enterprise tool portrays itself as an attacker and performs the attack on the application to find out the vulnerabilities. This is performed when the application is in it's running state. This provides much more insights on how the application behaves while they are in production and helping the organization to address the critical vulnerabilities before a hacker would exploit the vulnerability and cause damage to the organization.

2.3 SAST v/s DAST

SAST uses the white-box security testing approach for the security testing of the application. The application is tested from inside out. DAST uses black-box testing approach where the user has no knowledge about the application and testing is to be done. The application is tested from outside in.

SAST does not require the deployed application where as the DAST needs the application to be in deployed state. SAST analyzes the source code, libraries etc without the application being deployed. DAST analyzes the application by executing it.

SAST finds vulnerabilities in the initial phase of SDLC whereas DAST finds it towards the final phase of SDLC.

SAST can-not discover the issues in run-time whereas DAST can find the issues in run-time.

SAST typically supports all kind of programming languages whereas DAST supports only web application and web services.

Chapter 3

Literature Survey

3.1 OWASP Top 10 Vulnerabilities

According to OWASP Top 10 (2017) below are the top 10 vulnerabilities for any web application:

1. Injection
2. Broken Authentication
3. Sensitive Data Exposure
4. XML External Entities (XXE)
5. Broken Access Control
6. Security Misconfiguration
7. Cross Site Scripting
8. Insecure Deserialization
9. Components with Known Vulnerabilities
10. Insufficient Logging & Monitoring

These vulnerabilities would be detected by the Enterprise SAST Tools as well as DAST Tools. Due to such vulnerabilities present in the application sensitive information of the users/organization is compromised. SOAR would help monitor the different

sources, and if those vulnerabilities are found, it helps in reporting the vulnerabilities to the required people. Based on the criteria of the vulnerabilities and on the business importance of the applications further reporting and response takes place[1].

3.2 What is Governance, Risk & Compliance?

GRC refers to a plan or a protocol to manage the organization's overall governance, the risk imposed to the organization and the compliance with the Information Security policies. It is an approach to align the IT with business objectives of an organization as well as governing the risk imposed and making sure the compliance requirements are met.

As per Peter Weill and Joanne W. Ross, IT Governance means: Specifying the right decisions and accountability framework to achieve a desirable behavior in an organization with the help of IT department[2].

Definition of IT Risk mentioned in Information Technology Risk Management ISO/IEC 27005:2008 is given as: Risk is some potential that a particular threat will exploit some of the vulnerabilities of assets/system/application in an organization, thus causing various kind of losses to the organization[3].

Overall Risk Severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

Figure 3.1: Risk Calculation Methodology

Compliance processes makes sure that the requirements of the organization and measures associated to it are identified and prioritized, so that the efficiency of the policies designed is monitored, flaws are addressed, and appropriate reporting on compliance status is available[4].

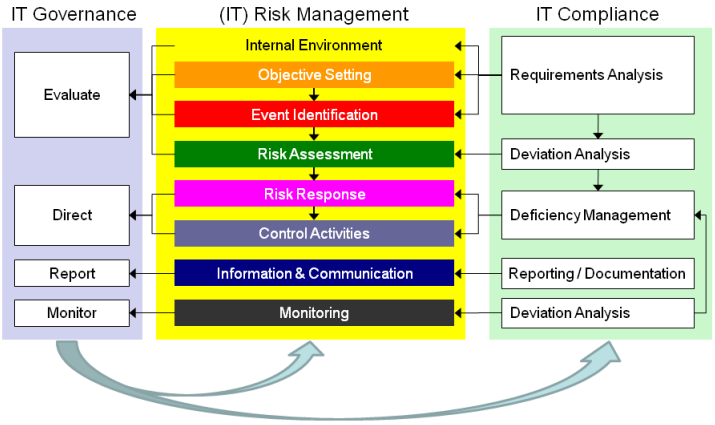

Figure 3.2: IT GRC Process Management
[5]

3.3 Why SOAR?

Using SOAR improves the efficiency and efficacy of the AppSec Operations team. It also helps in enhancing the way an incident is handled and also reporting and capturing the knowledge.

According to a survey by a company[6], here are the results that show why is SOAR effective in the organization whose main focus is securing the information that the application stores.

Not all the alerts generated by a system are as serious and as important as they prompt. Some of them don't require attentions where as some of them require high attention. Those alerts would be categirized in three categories: **Trivial Alerts**,**Minimal Isvestigation Alerts**,**Investigation & Remediation Alerts**.

In case of auditing and for compliance issues, reports are to be provided to the higher management as well as auditors. **Alert-Specific Reports** provide step-by-step description of what the response plan did in order to prevent the threat that was detected. **Managerial Reports** provide insight into performance of security operations. They help answer questions related to how many alerts were received, how many were closed, etc. There are various operational tasks such as **Creating Playbooks** for various tasks, **Shift Handover**, etc which require lots of time and money.

For an organization, SOAR enables:

- IT operations, Integrating Security & Threat Intelligence Tools
- Everything at a single place
- Responding the incidents in a short time
- Get smart and better intelligence
- Improve the communication and the reporting
- Speedup the decisions
- Flexibility, Extensibility, and Collaboration

Using SOAR improves the efficiency of the Application Security Operations team in terms of time spent, in terms of money and in terms of analysis of the vulnerability. Different policies are designed by the organization and based on those policies, different playbooks are designed and SOAR is customized for the use in organization.

Chapter 4

Security Orchestration, Automation & Response

4.1 What is SOAR?

SOAR stands for Security Orchestration, Automation & Response. It is a collection of solutions that help an organization to streamline the security operations such as **Threat & Vulnerability Management, Incident Response and Security Response Automation**. SOAR collects data from different sources regarding the security threats caused to the organization and automates the process to remediate some of the threats to an organization without any human intervention. It uses data from different sources, maps it with the security policy in order to find if an application is compliant to the policy or not. If the application is found to be compliant no actions would be taken and only monitoring would be done. If the application is found to be non-compliant[6], then a specific set of actions mentioned in the policy would be taken in order to make sure the application is now compliant.

4.1.1 Security Orchestration

It is a method of integrating different security tools to streamline the process and make it easier for security teams to work. In this phase data from various resources like SAST Tool, DAST Tool, etc., is collected and monitored to make sure the application stays compliant with the Information Security policies. Here vulnerabilities count provided the SAST tool & the DAST tool would be monitored and based on the policy defined, it

would be made sure that the count of vulnerability does not exceed an agreed number.

4.1.2 Automation

Some actions are to be taken in order to make the application compliant to the policies of the organization. These actions if performed manually, takes lot of time. Using Automation, it can be made simple to take certain actions without human intervention. Assume that an organization has a policy of making sure that the vulnerabilities count in an application at the end of month should be always zero. So if the scan prompts out vulnerability, the tool would automatically send out the alert based on the severity of the vulnerability and business importance of the application.

4.1.3 Response

In this phase the remediation plan for the incident is initiated to resolve the incident. Incident here is referred to a scenario where there are High Severity vulnerabilities in an application and they are exposed to the outer world. In such cases if the application is of high business importance, then the response plan for such would be blocking the source of the attack by modifying the network configurations. The response plan is built by keeping in mind various facts and the GRC model. Below mentioned are some such facts:

1. **Define, Analyze & Understand Risk caused to the organization**

    - Here is the time to create the risk assessment plan. Initially there should a level defined that should be known which would be treated as an incident. Then there is the need to deciding what kind of data is being processed by the application/system for which this actions would be taken. Next find out the stakeholders that have access to this data and what kind of benefit would the attacker gain by stealing that data. Understand the value of that data, where and how is it being stored and with this select the response plan appropriately.

2. **Document the Response Severity**

    - The organization in step comes up with a tree stating what kind of severity is to be addressed by whom. The severity is decided based on the threat the

vulnerability causes to an organization and the severity of the application. Maintaining this documentation for analysts when they need it in the event of a sudden incident would be proved efficient and effective.

3. **Run the Response Plan drill with the important key stake holders**

   - The most important step to note during this phase is to document all the steps of the process. Beginning from the response of teams that are not related to security to the reaction of leaders in event of a breach. This is the time to design the job and execute it.

4. **Develop a Disaster Recovery Plan**

   - If proper actions are taken by the analyst, the management and the stake holders there would be no need to get into this step at all. But if in some situation the plan fails, having a disaster recovery plan will prove to be beneficial. Enabling frequent backups, moving them into a dedicated server separate from the organization network is an example of simple disaster recovery plan.

4.2 Advantages of Using SOAR

1. **Cross team workflows and tools are connected and coordinated**

   - To make sure the organization is safe from all the risks and various attack vectors, they buy various firewalls, IDS, threat intelligence tools, anti-malware and anti-virus tools. Within no time, the number of tools pile up and the security team would be stuck managing various tools. These tools are not designed in such a way that they can communicate with each other, however the SOAR acts as an intermediate between them makes it easy for the security teams to manage and orchestrate the tools efficiently from a single point.

2. **Avoid alert overload**

   - Organizations have various tools for managing the security. With this comes a huge number of alerts that would be received from the tools by the analysts. The analysts would have to go through each of them and then forward it to the particular application administrator. Instead of this, SOAR helps design

polices that would meet the GRC requirements and if the policy is violated then direct alerts would be sent out to the application administrator. The analysts would come to know about the alerts and they would also start analyzing those instances. Thus this would reduce the load of analyzing even the ones that would be false positive as well as sending them manually.

3. **Standardize the incident response process**

   - Not all the analysts would handle the threat/vulnerability in a similar way. While this looks like a normal situation, this could lead to an increase in the number of false alarms associated with human error and thus lead to ineffective response handling strategy. SOAR surely does not replace the human intelligence as there are various aspects of cybersecurity that are best understood and put into action by human brain. SOAR can be used to resolve issues faster, with more efficiency and with more consistency.

Chapter 5

Conclusion

In an organization with large number of applications running inside, it is not possible that all the applications are compliant as per the GRC guidelines. In order to make sure that the applications stay compliant and there is no overhead of false positives on an analyst, SOAR tool proves to be helpful. So as to fulfil these conditions, SOAR can be used, which helps the analysts to automate the process of security operations regarding the non-compliance of their application. It reduces the human efforts and makes team effecient in the operations.

Bibliography

[1] T. AlSadhan and J. S. Park. Enhancing risk-based decisions by leveraging cyber security automation. In *2016 European Intelligence and Security Informatics Conference (EISIC)*, pages 164–167, Aug 2016.

[2] Peter Weill, Jeanne Ross, and IT Governance. How top performers manage it decision rights for superior results. *Harvard Business School Press, Boston, MA*, 2004.

[3] Iso/iec 27005:2008 information technology risk management.

[4] Christof Menzies, A Martin, M Koch, C Trebuth, S Esche, T Heinze, and P Stähle. Governance, risk management and compliance: Sustainability and integration supported by technology. *PricewaterhouseCoopers AG*, 2007.

[5] D. Puspasari, M. Kasfu Hammi, M. Sattar, and R. Nusa. Designing a tool for it governance risk compliance: A case study. In *2011 International Conference on Advanced Computer Science and Information Systems*, pages 311–316, 2011.

[6] A. Al-Omari, O. El-Gayar, and A. Deokar. Security policy compliance: User acceptance perspective. In *2012 45th Hawaii International Conference on System Sciences*, pages 3317–3326, Jan 2012.