Web Development in Advanced Threat Prevention

Rakesh Reddy Peddamallu

ELECTRONICS AND COMMUNICATIONRV COLLEGE OF ENGINEERING, Bangalore, India.

***Abstract*—In this paper, the event of the 2 merchandise SRE Dash- board and Firestorm that are a part of Advanced threat interference, the data integration by API's using React Reducer functions , and therefore the backend safety features like , preventing the IP addresses that are a part of protected networks and preventing XSS (Cross web site Scripting) attacks is conferred Cross-site scripting works by manipulating a vulnerable data processor so it returns malicious JavaScript to users. once the malicious code executes within a victim's browser, the aggressor will totally compromise their interaction with the applying.**

In this paper the UI for SRE Console web site and group action the web site with real time knowledge mistreatment API's and react technologies like React , Typescript and frameworks like Ant Design , Slipstream.

I. INTRODUCTION

SASE is everywhere. Recently, Secure Access Service Edge (SASE) has been launched seemingly overnight. Coined by Gartner, this term refers to the integration of a high degree of network-related security into a design that brings applications and services closer to the complete user. Like some shiny new toy, some vendors have jumped at the chance to become a SASE supplier. The result of connecting first can be a visible advantage.

However, these vendors ignore the basic rules of security: unified visibility and control. With SASE, some have broken that promise and returned to the era of complete security silos.

Security Manager allows organizations to manage security any- where in the cloud, on-premises, with unified policy management that tracks users, devices and applications. This policy is created once and applies to the whole. Use Cloud Administrator to pro- vide network-wide visibility and policy management for your on- premise, cloud, and service deployments.

Security Director is a portal from the Juniper portal to the Secure Access Services Edge (SASE) design. Easily manage and enforce security policies from a single user interface across all environ- ments. Use Security Manager's Integrated Security Orchestration Policy Helper to fine-tune threat mitigation policies and micro segmentation across your network.

This white paper introduces some back-end security features for developing protected cloud user interfaces and one back-end feature for cleaning the input fields of all online admin pages to prevent XSS attacks.

A cross-site scripting (XSS) attack unit is a type of injection in which a malicious script area is injected into a harmless, trusted website. XSS attacks occur when a sophisticated fraudster uses an online application to send malicious code, usually in the form of a browser script, to a specific user. The flaws that would allow these attacks to succeed on a regional unit are widespread, anywherean online application uses user input in output generated without support or encryption.

Applications of Advanced Threat Prevention

- Manage tens of thousands of sites simultaneously
- Create policies for validated threat prevention, user and application access control secure connectivity, and more — and apply them anywhere
- Protect private and public cloud workloads with metadata-based security controls
- Security Assurance guarantees that security rules are always placed correctly for intended effectiveness
- Correlate and analyze each stage of an attack in sequence, regardless of which product made the detection, and stop threats across your network with Policy Enforcer's one-click mitigation Problem statement of the project is to develop the UI for SRE Console website and integrating the website with real time data using API's and react technologies like redux , Typescript and frameworks like Ant Design , Slipstream.

II. LITERATURE SURVEY

O. C. Novac, D. E. Madar, C. M. Novac, G. Bujdoso , M. Oproescu, and T. Gal presents two simple applications, one in React.js and another in Vue.js and comparison between them, is presented The capabilities of these two frameworks were vary depending on the size of the applications that need to be im- plemented and comparison between them was made React.js and Vue.js frameworks in this case study were compared and the performance of these were checked during the experiment

D. T. H. Thu, D.-A. Nguyen, and P. N. Hungr proposes a Pattern- based Unit Testing method, namely PUT, to generate test data automatically for Typescript web applications The main idea of the paper is to analyse the internal structure of source code to detect patterns of attribute usages.

I. Vershinin and A. Mustafina discusses the TPC-H test and 22 of its queries, the execution of which allows you to analyse the per- formance of database management systems (DBMS). PostgreSQL, MySQL, and Microsoft SQL Server DBMS are considered and their performance is investigated based on TPC-H test queries with database (DB) of various sizes in the dB Visualizer

M. de Sousa and A. Gon calves presents a real case study, where the company humansoft, revised one of its main solutions, human portal, using one of the most recent and used web technology,React.js This project provides a first-class way to render children into a DOM node that exists outside the DOM hierarchy of the parent component.

B. Chen and Y. Shi proposed, new CSS features are proposed for this malicious hidden redirected attack web page, and theoretical analyssis and experimental verification are carried out. The method proposed in this paper is mainly for the identification of malicious web pages that use hidden redirection to download malware.

P. Rodeghero, C. McMillan, and A. Shirey , presents a study on exploring the use of API keywords within method summaries, explores the question of whether programmers use API's to de- scribe functionality implemented in source code that calls those API's,covers background information on API usage, method sum- maries

J. Harper and M. Agrawala presents a technique for converting a basic D3 chart into a reusable style template, Also focuses on Algorithms for constructing style templates from D3 charts. Deconstructing a D3 chart

A. Telea, A. Kerren, and A. Marcus [9] introduces VisGi, a tool to abstract and visualize the branch structure of Git repositories, as well as their folder trees Additionally, this paper provides Sunburst diagrams which are used to display the current content of these branches, the differences between each two branches, as well as the evolution along any singular selected path through the repository.

III. METHODOLOGY

Brief Methodology of the project To develop the UI and adding backend security features firstly the UI is developed as per the FIGMA design given by the UX team in Juniper and in that design several things like CSS and other design features should be followed

. To implement this in the next step the setup to work is made and all the necessary libraries are installed and also the frameworksafter this step we develop the UI .In the next step data integrationis done at web pages where real time data has to be populated .Once after this several backend features like preventing XSS (Cross Site Scripting) attacks [20] and other security related features are added

.In the next step code is reviewed by mentors and merged into master branch once upon the MR (Merge Request) approvals .In the next step QA test is done and verified In the final stage we generate the docker images of the developed code which will be done by the other team members ,docker images is a kubernetes concept which will rectify the issue of code breaking in some machines and code working fine in other machines the code which is developed is containerized which makes it portal to run code fine on any machine

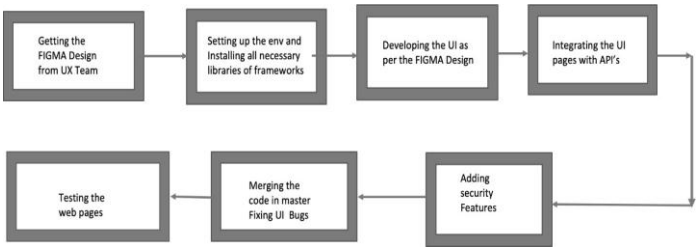

Fig. 1. Methodology of the project

Figma is a browser-based collaborative interface design tool with additional offline features that enable desktop

applications for macOS and Windows. The Figma mobile app for Android and iOS enables real-time viewing and interaction with Figma prototypeson mobile devices and tablets. Figma's feature set focuses on user interface and user experience design, with an emphasis on real-time collaboration, using various vector graphics editors and prototypingtools

IV. BACKGROUND

A. Backend Implementation for Firestorm of ATP

The main risks of using a public IP address have the advantage of being the same. Anyone, anywhere can connect directly to the device from the web. Cybercriminal As they claim, when you connect to the web, the web connects to it You, in this case - directly. By exploiting various vulnerabilities, cybercriminals They get your files and steal sensitive information to sell or blackmail. Attackers often change their network access settings. For example, by force A router that feeds phishing websites where you paste your login information. How do hackers know who to attack? First of all,there is something that is publicly available Internet services that often scan all IP addresses for vulnerabilities and create thousands of them For devices with exploitable bugs, just a click or two. When cybercriminals need a prod Not only keeping personal IP, specifically keeping your IP and using Skype, they are located. For example. Even if you visit the website once, your address will be displayed. However, the real IP address is not only used to hack your home network, In addition, it stops DDOS attacks by attacking with completely different packets. It uses the device and loads net channel and router at the same time. Your ISP You are protected - are you? Such attacks are usually against gamers and For example, a streamer knocks out an opponent by interfering with the enemy. Internet forum So, this is the reason for adding security features through the backend. Inspection.

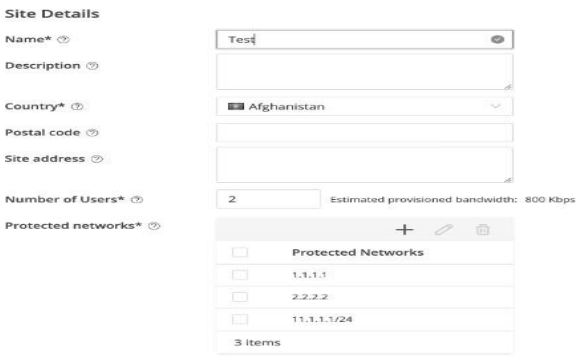

Fig. 2. Adding Security Functionality to the page

In the above figure security functionality is being added, the protected networks can take the input of either an IP address or a network and the validation is being added in the UI. In the next image there is a field to enter site IP address, that IP should not bein either the IP of protected network or in the entire network that is the required feature.

An IP address consists of a network number (routing prefix) and the remaining fields (host identifier) [21]. The remaining fields are identifiers that are unique to a particular host or network interface. Routing prefixes are often represented in both IPv4 and IPv6 using Classless Interdomain Routing (CIDR) notation. CIDR is a method used to create unique identifiers for networks and individual devices. For IPv4, networks can also be specified using a subnet mask, which may be represented as a dotted-decimal notation, as shown in the "subnet" field of the computer. Unlike host identifiers, which are locally unique identifiers, all hosts on a subnet have the same network prefix. In IPv4, these subnet masks are usedto distinguish between network numbers and host IDs. In IPv6,the network prefix performs a similar function to the IPv4 subnet mask, and the length of the prefix represents the number of bits in the address.

Fig. 3. Possible subnets of network - 11.1.1.0/24e

In the above figure possible subnets of network 11.1.1.0/24 are shown. Class A Blocks end with "/8" and contain 16 million IP addresses. Class B Blocks end with "/16" and contain 65,000 IP addresses. Class C Blocks end with "/24" and support a maximum of 254 IP addresses.

- Class A Example - 1.1.1.0/8 - 16 Million Hosts
- Class B Example - 1.1.1.0/16 - 65,000 Hosts
- Class C Example - 1.1.1.0/24 - 254 Hosts

So if we want to check if certain IP is part of 1.1.1.0/8 network then we need to iterate in an array of 16million items and it slows down the process as well as memory to store 16 million items is very much so a different logic has to be implemented which is mentioned below

B. Preventing XSS Attacks

Cross-site scripting (XSS) attacks measure a type of injection that contains malicious scripts. Otherwise, it is injected into a benign and trustworthy website. XSS attacks happen when you get an asso- ciate degree Attackers use Internet applications to deliver malicious code, usually in some type of browser Side script, to another user. The flaws that make these attacks successful are significant. It is widespread wherever Internet applications use internal user input Output is generated without validation or coding. An attacker uses XSS to send a malicious script to link an unsuspecting user. The end user's browser has no way of understanding that the scriptis ambiguous. Run the script. As a result, the script is believed to have come from a trusted, malicious source. Scripts access cookies, session tokens, or other sensitive data The code used to protect the form is shown in the figure. Here we get the values from the form, iterate over the form and then clear each value from the objectkey in the iterative process. GitHub's additional DOM cleaner open source library is used, and you can also pass a configuration variable if you want to clean it in a custom way. The code takes the raw values and uses the library to convert those raw values to cleared values for all text fields, for example if the user enters a descriptionlike: .

<script>hello </script>rakesh

It strips of the script and its content and the sanitized code returns only 'rakesh' as output in this way we are sanitizing all the fields and protecting the data by not allowing any java-script or database queries to get executed.

DOMPurify sanitizes HTML and prevents XSS attacks. You can feed DOMPurify with string full of dirty HTML and it will return a string (unless configured otherwise) with clean HTML. DOMPurify will strip out everything that contains dangerous HTML and therebyprevent XSS attacks and other nastiness. It's also damn bloody fast.We use the technologies the browser provides and turn them intoan XSS filter. The faster your browser, the faster DOMPurify will be in the end sanitization is done at all places where a user can input malicious code so in this way various cyber attacks like man in the middle ,DDOS and some other attacks can be prevented

Code for added XSS prevention feature

Fig. 4. Code to secure form from XSS attacks

CONCLUSION AND RESULTS

The pipeline results of the implemented code execution is shown below where all the jobs are satisfied and the build is passed with all the unit test cases passed and successful run of CI build.

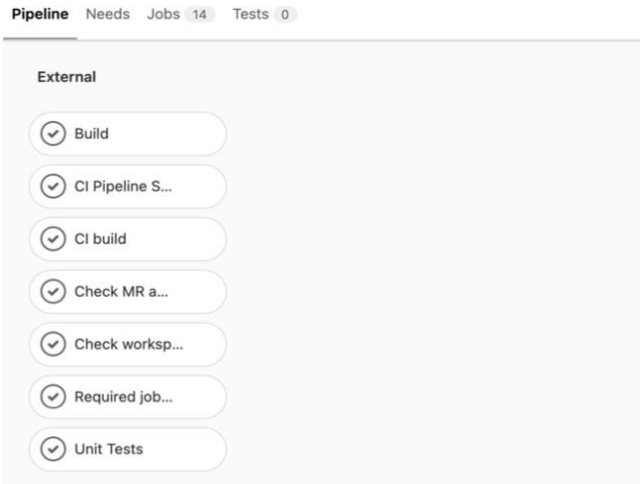

Fig. 5. Pipeline Results of the implemented Code

The Results of build execution and the code performance with time taken to get build is shown below. Confirming a build isa build You can test before checking the team and doing more tests. For these, see Core Unit Area This feature looks for things that ensure the app is stable and can be fully tested. Looking at build acceptance, it's usually a set of simple tests that are the most important Stream the device's computer code application. Make that failed build confirmation The test is rejected and the test continues with the previous build (at least one build that passed the acceptance test). This is a group of tests that are run on every new build of the product. It is testable before the build is free into the hands of the take a look at team. The build results of the developed code is shown in the Figures

Fig. 6. Result of Build execution and the time taken

The Results of the build execution are shown in the figure where the build time taken is 24 min's 26 sec and the execution of the shell script took around 16sec's

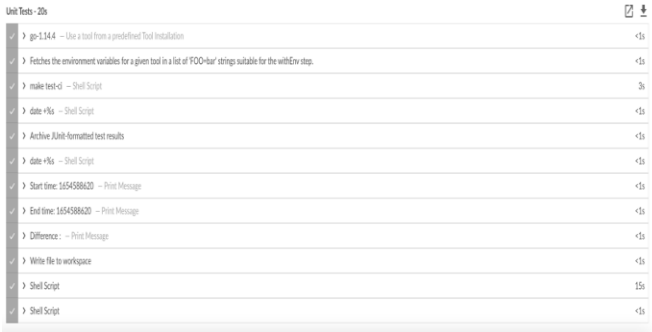

Fig. 7. Result of Unit Tests and the time taken

Unit tests include testing all units or private parts of your code program. This is a key level of useful testing. The purpose of unit testing is to validate the elements of the unit with their functionality.

The unit is a testable part of the package and may be tested throughout the event section of the device code.

The purpose of unit tests is to check the correctness of isolated code. Components are the unit of function or the private code of the device. A white box testing approach used for unit testing and usually performed by developers

The above image tells the results of Unit Testing passed which took around 20 seconds to get executed

Fig. 8. Result of Building Docker Images and the time taken

A Docker image is a file used to run code in a Docker container. A Docker image acts like a template as a set of instructions for building a Docker container. A Docker image also serves as a starting point when using Docker. An image is a snapshot of a virtual machine (VM) environment.

Docker is used to create, run and deploy applications in contain- ers. A Docker image contains the application code, libraries, tools, dependencies, and other files needed to run your application. When a user executes an image, the image can be one or more instances of the container. The following figure shows the result of building the Docker image.

REFERENCES

[1] Novac, Ovidiu Constantin, et al. "Comparative study of some applications made in the Angular and Vue. js frameworks." 2021 16th International Conference on Engineering of Modern Electric Systems (EMES). IEEE, 2021.

[2] D. T. H. Thu, D. -A. Nguyen and P. N. Hung, "Automated Test Data Generation for Typescript Web Applications," 2021 13th International Con- ference on Knowledge and Systems Engineering (KSE), 2021, pp. 1-6, doi: 10.1109/KSE53942.2021.9648782.

[3] N. Black, "Boris Cherny on TypeScript," in IEEE Software, vol. 37, no. 2, pp. 98-100, March-April 2020, doi: 10.1109/MS.2019.2958155.

[4] A. Javeed, "Performance Optimization Techniques for ReactJS," 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), 2019, pp. 1-5, doi: 10.1109/ICECCT.2019.8869134.

[5] Chen, Boshen Shi, Yijie. (2018). Malicious Hidden Redirect Attack Web Page Detection Based on CSS Features. 1155-1159. 10.1109/CompComm.2018.8780783.

[6] Harper, Jonathan and Agrawala, Maneesh. (2016). Converting Basic D3 Charts into Reusable Style Templates. IEEE Transactions on Visualization and Computer Graphics. PP. 10.1109/TVCG.2017.2659744.

[7] Nguyen, Trong and Nguyen, Anh and Phan, Hung and Tien, Nguyen. (2017). Exploring API Embedding for API Usages and Applications. 438-449. 10.1109/ICSE.2017.47.

[8] Acosta-Vargas, Patricia and Luján-Mora, Sergio and Salvador-Ullauri, Luis. (2017). Quality evaluation of government websites. ICEDEG. 10.1109/ICEDEG.2017.7962507.

[9] C. Boldyreff, "Determination and evaluation of Web accessibility," Proceed- ings. Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2002, pp. 35-40, doi: 10.1109/EN-ABL.2002.1029985.

[10] Vigo, Markel and Leporini, Barbara and Paternò, Fabio. (2009). Enriching web information scent for blind users. ASSETS'09 - Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility. 123-130. 10.1145/1639642.1639665.

[11] J. Carter and M. Markel, "Web accessibility for people with disabilities: an introduction for Web developers," in IEEE Transactions on Professional Com-munication, vol. 44, no. 4, pp. 225-233, Dec. 2001, doi: 10.1109/47.968105.

[12] Z. Jingyu, H. Hongchao, H. Shumin and L. Huanruo, "A XSS Attack Detection Method Based on Subsequence Matching Algorithm," 2021 IEEE International Conference on Artificial Intelligence and Industrial Design (AIID), 2021, pp. 83-86, doi: 10.1109/AIID51893.2021.9456515.

[13] A. Shrivastava, S. Choudhary and A. Kumar, "XSS vulnerability assessment and prevention in web application," 2016 2nd International Conference on Next Generation Computing Technologies (NGCT), 2016, pp. 850-853, doi: 10.1109/NGCT.2016.7877529.

[14] K. Ali, A. Abdel-Hamid and M. Kholief, "Prevention Of DOM Based XSS Attacks Using A White List Framework," 2014 24th International Confer-ence on Computer Theory and Applications (ICCTA), 2014, pp. 68-75, doi: 10.1109/ICCTA35431.2014.9521633.

[15] C. M. Frenz and J. P. Yoon, "XSSmon: A Perl based IDS for the detection of potential XSS attacks," 2012 IEEE Long Island Systems, Applications and Technology Conference (LISAT), 2012, pp. 1-4, doi: 10.1109/LISAT.2012.6223107.

[16] Liu, Miao, et al. "A survey of exploitation and detection methods of XSS vulnerabilities." IEEE access 7 (2019): 182004-182016.

[17] Gupta, Shashank, and Brij Bhooshan Gupta. "Cross-Site Scripting (XSS) at- tacks and defense mechanisms: classification and state-of-the-art." International Journal of System Assurance Engineering and Management 8.1 (2017): 512- 530.

[18] Sarmah, Upasana, D. K. Bhattacharyya, and Jugal K. Kalita. "A survey of detection methods for XSS attacks." Journal of Network and Computer Applications 118 (2018): 113-143.

[19] Grossman, Jeremiah, et al. XSS attacks: cross site scripting exploits and defense. Syngress, 2007.

[20] Wurzinger, Peter, et al. "SWAP: Mitigating XSS attacks using a reverse proxy." 2009 ICSE Workshop on Software Engineering for Secure Systems. IEEE, 2009.

[21] Shahriar, Hossain, and Mohammad Zulkernine. "S2XS2: a server side approach to automatically detect XSS attacks." 2011 IEEE Ninth International Confer- ence on Dependable, Autonomic and Secure Computing. IEEE, 2011.