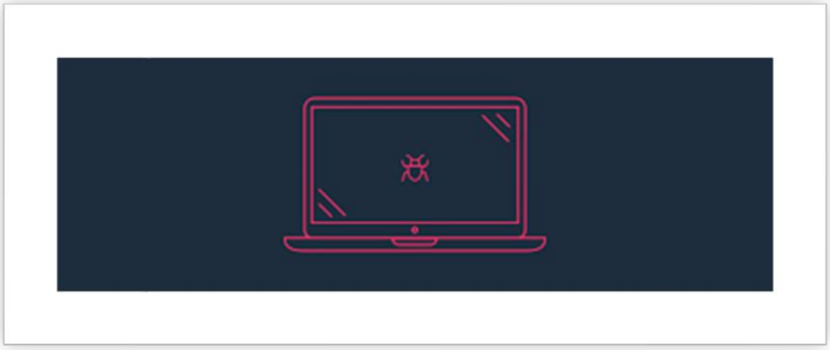


Malware Detection & Classification

Team-2.12

Abstract:

The Malware Detection and Classification Project is an important effort in the fast-evolving field of cybersecurity. It aims to create a smart and robust system capable of accurately identifying and categorizing malware, which poses a threat to our digital world. The project employs advanced machine learning techniques and large malware datasets to support cybersecurity endeavors. Conventional malware detection methods based on signatures are ineffective against new and unknown malware variants, making malware identification and classification a crucial aspect of cybersecurity. Machine learning systems can be trained on massive datasets of malware samples to detect patterns that distinguish malware from legitimate software, and to categorize previously unidentified malware types.

Vision:

The Malware Detection and Classification Project aims to develop a sophisticated cybersecurity system that detects and categorizes malware to boost businesses and individuals' resilience against online attacks. The project team is creating ML models, including anomaly detection and static and dynamic malware analysis, to implement defenses against malware assaults for enterprises. Additionally, a centralized malware detection and categorization system is being developed to safeguard businesses of all sizes and protect digital ecosystems across various sectors.

- **Accurate:** Even if a malware variant has never been seen before, the system needs to be able to identify and classify it with high accuracy.
- **Fast:** The system must be able to identify and categorize malware samples instantly.
- **Scalable:** To manage high amounts of malware samples, the system must be scalable.
- **Easy to use:** Security experts of various skill levels should have no trouble using the system.

This project's vision-based malware detection and categorization system has the potential to be applied in several different scenarios, such as:

- **Endpoint security:** The system may be used to defend against malware assaults on endpoints, such as PCs and mobile devices.

- **Network security:** The system may be used to watch for malicious behaviour in network traffic.
- **Cloud Security:** The system may be used to thwart malware assaults on apps and data stored in the cloud.

- **Security analysis:** Security analysts may utilize the system to find and examine fresh malware versions.

Benefits:

- **Security enhancement:** The system can contribute to security enhancement by identifying and categorizing malware variants that are not identified by conventional techniques.

- **Cost savings:** The system can assist in lowering the expenses of malware assaults, including those related to data breaches and downtime.

- **Productivity gains:** The system can aid in productivity gains by decreasing the time users must spend addressing malware outbreaks.

Conclusion:

The Malware Detection and Classification Project aims to strengthen cybersecurity defenses by developing a dynamic system that identifies and classifies malware using machine learning and large datasets. The project is fine-tuning and improving the system to ensure its correctness, dependability, and flexibility. The project hopes to create a more secure online environment and promote innovation, growth, and security in the digital world. The Cyber Detection and Classification Project is also creating ML-based defenses against cyber assaults on enterprises. This project has the potential to significantly advance cybersecurity and offers several advantages to users.

Vulnerabilities Report for Practice Website

1. **Vulnerability Name:**

Protection Mechanism Failure (Web Application Potentially Vulnerable to Clickjacking)

CWE: CWE-693

OWASP Category: A04:2021 – Broken Access Control

Description:

This weakness covers three distinct situations. A "missing" protection mechanism occurs when the application does not define any mechanism against a certain class of attack. An "insufficient" protection mechanism might provide some defences - for example, against the most common attacks - but it does not protect against everything that is intended. Finally, an "ignored" mechanism occurs when a mechanism is available and in active use within the product, but the developer has not applied it in some code path.

Business Impact:

Protection Mechanism Failure in web applications potentially vulnerable to clickjacking can result in severe business impacts. Clickjacking is a malicious technique where attackers deceive users into performing unintended actions. When protection mechanisms fail, it can lead to unauthorized data exposure, financial losses, and reputational damage. This, in turn, can erode trust, harm customer relationships, and lead to legal consequences, making robust security measures crucial for preserving a business's integrity and success.

Vulnerability Path: http://testfire.net/

Vulnerability Parameter: https://demo.testfire.net/bank/main.jsp

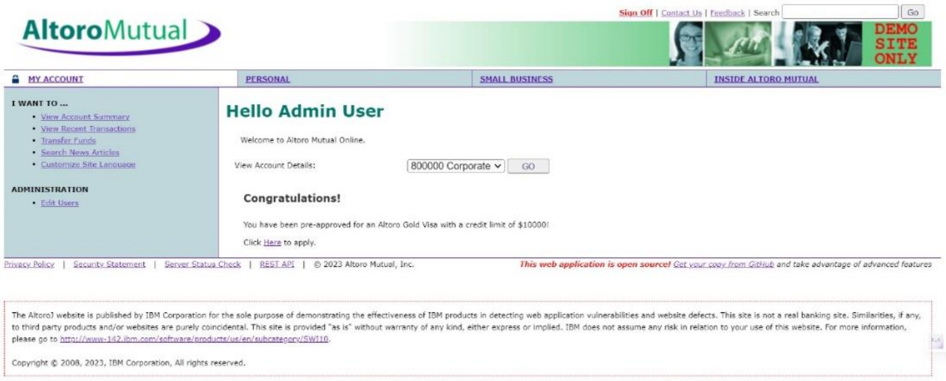
Steps to Reproduce:

Identify a security feature or protection mechanism in the application (e.g., authentication, access control).

Attempt to bypass or defeat the security mechanism.

Observe whether the application allows unauthorized access or circumvention.

If you successfully bypass the protection mechanism, you have reproduced a protection mechanism failure issue.



Hello Admin User

Welcome to Altoro Mutual Online.

View Account Details: 800000 Corporate

Congratulations!

You have been pre-approved for an Altoro Gold Visa with a credit limit of $10000!

Click Here to apply.

This web application is open source! Get your copy from GitHub and take advantage of advanced features

The AltoroJ website is published by IBM Corporation for the sole purpose of demonstrating the effectiveness of IBM products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. IBM does not assume any risk in relation to your use of this website. For more information, please go to http://www-142.ibm.com/software/products/us/en/subcategory/SWI10.

Copyright © 2008, 2023, IBM Corporation, All rights reserved.

Recommendation:

Return the X-Frame-Options or Content-Security-Policy (with the 'frame-ancestors' directive) HTTP header with the page's response.
This prevents the page's content from being rendered by another site when using the frame or iframe HTML tags.

2. Vulnerability Name:

Improper Input Validation (CGI Generic XSS)

CWE: CWE-20
OWASP Category: A04:2021 – Injection

Description:

The product receives input or data, but it does not validate or incorrectly validates that the input has the properties that are required to process the data safely and correctly.

Business Impact:

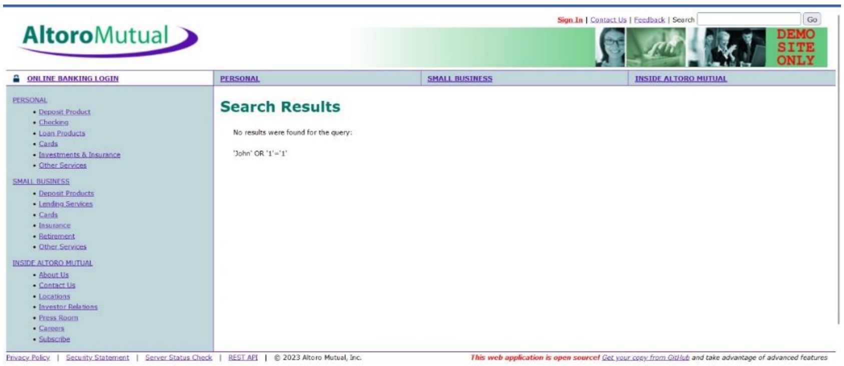
Improper input validation, specifically in the context of CGI scripts, can lead to Cross-Site Scripting (XSS) vulnerabilities. The business impact of this vulnerability includes risks of data theft, unauthorized access, and damage to a company's reputation, potentially resulting in financial losses, legal liabilities, and loss of customer trust.

Vulnerability Path: http://testfire.net/
Vulnerability Parameter:
http://testfire.net/search.jsp?query=%27John%27+OR+%271%27%3D%271%27

Steps to Reproduce:

Identify a user input field in the target application, such as a search box or login form.
Attempt to enter malicious input, like SQL code, HTML tags, or special characters, in the input field.
Submit the input and observe if the application processes it without validation or sanitization.

If the application accepts and processes the input without any validation or sanitization, you have successfully reproduced the issue.



Search Results

No results were found for the query:

'John' OR '1'='1'

Recommendation:

The recommendation for addressing Improper Input Validation (CGI Generic XSS) is to implement robust input validation and filtering mechanisms in web applications to prevent the execution of malicious cross-site scripting (XSS) attacks, typically by sanitizing and escaping user input data before rendering it on web pages.

3. Vulnerability Name:

Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') (CGI Generic XSS)

CWE: CWE-79
OWASP Category: A04:2021 – Injection

Description:

The product does not neutralize or incorrectly neutralizes user-controllable input before it is placed in output that is used as a web page that is served to other users.

Business Impact:

The business impact of improper neutralization of input during web page generation, commonly known as 'Cross-site Scripting' (XSS), can be significant. It can result in compromised user data, loss of customer trust, reputational damage, and potential legal and regulatory consequences. XSS attacks can enable malicious actors to inject malicious code into web pages, leading to data theft, account hijacking, and other security breaches, which can harm a company's bottom line and credibility.

Vulnerability Path: http://testfire.net/
Vulnerability Parameter:
http://testfire.net/search.jsp?query=%27+%3Cscript%3Ealert%28%22XSS+Attack%21%22%29%3B%3C%2Fscript%3E%27

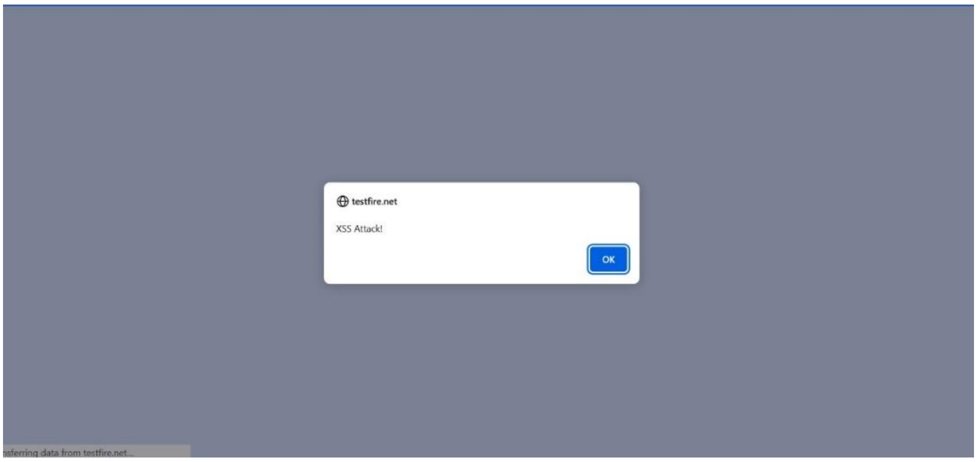
Steps to Reproduce:

Locate a web page that includes user-generated content, such as comments or input fields.

Insert malicious script code, such as JavaScript, into the input field.

Submit the input, and view the page to see if the script executes in the context of another user's session.

If the script runs when viewed by another user, you have successfully reproduced a Cross-site Scripting (XSS) issue.



testfire.net

XSS Attack!

Recommendation:

The recommended solution for Improper Neutralization of Input During Web Page Generation, commonly known as Cross-site Scripting (XSS), is to implement proper input validation and output encoding to prevent malicious scripts from executing on a website. This involves validating and sanitizing user inputs and encoding output data to ensure that potentially harmful code cannot be injected or executed on the web page.

4. Vulnerability Name:

Improper Neutralization of Script-Related HTML Tags in a Web Page (Basic XSS) (CGI Generic XSS)

CWE: CWE-80

OWASP Category: A04:2021 – Injection

Description:

The product receives input from an upstream component, but it does not neutralize or incorrectly neutralizes special characters such as "<", ">", and "&" that could be interpreted as web-scripting elements when they are sent to a downstream component that processes web pages.

Business Impact:

The business impact of Improper Neutralization of Script-Related HTML Tags (Basic XSS) in a web page can be severe, leading to compromised user data, damaged reputation, legal consequences, and financial losses. This vulnerability allows attackers to inject malicious scripts into web pages, potentially stealing sensitive information, disrupting services, and undermining customer trust.

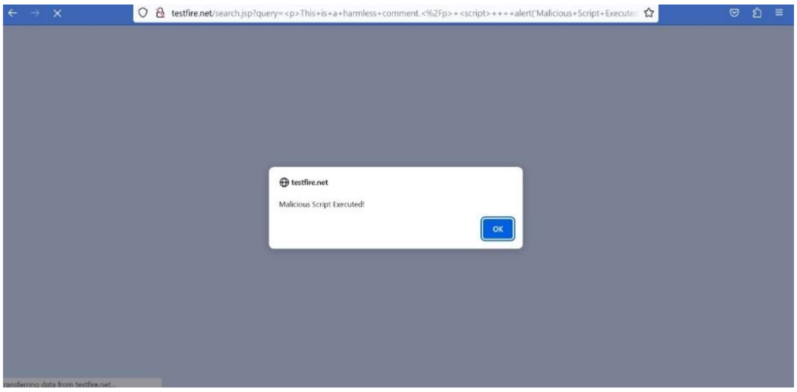
Vulnerability Path: http://testfire.net/

Vulnerability Parameter:

http://testfire.net/search.jsp?query=%3Cp%3EThis+is+a+harmless+comment.%3C%2Fp%3E+%3Cscript%3E++++alert%28%27Malicious+Script+Executed%21%27%29%3B+%3C%2Fscript%3E

Steps to Reproduce:

Find a web page with user-generated content, like comments or profile descriptions.
Inject a malicious script by adding HTML tags containing JavaScript into the content.
Save or post the content and view the page to check if the script is executed.
If the script runs in the context of the page, you have successfully reproduced
a Basic XSS issue



testfire.net/search.jsp?query=<p>This+is+a+harmless+comment.<%2Fp>+<script>++++alert('Malicious+Script+Execute

testfire.net

Malicious Script Executed!

Recommendation:

To mitigate Improper Neutralization of Script-Related HTML Tags in a Web Page (Basic XSS) (CGI Generic XSS) vulnerabilities, ensure proper input validation and output encoding to prevent malicious script injection. Utilize web application firewalls (WAFs) and security best practices, like using Content Security Policy (CSP) headers, to protect against cross-site scripting attacks.

5. Vulnerability Name:

Improper Neutralization of Special Elements in Output Used by a Downstream Component ('Injection') (CGI Generic XSS)

CWE: CWE-74

OWASP Category: A04:2021 – Injection

Description:

The product constructs all or part of a command, data structure, or record using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify how it is parsed or interpreted when it is sent to a downstream component.

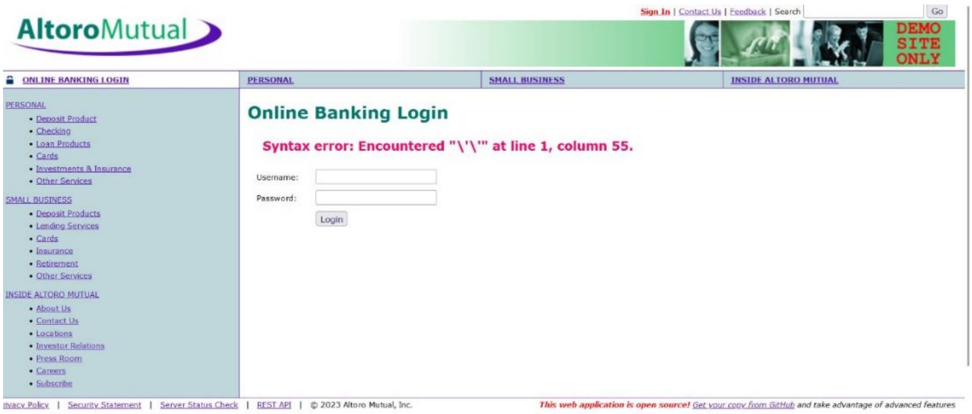
Business Impact:

The business impact of "Improper Neutralization of Special Elements in Output Used by a Downstream Component" (CGI Generic XSS) includes potential security breaches, data theft, and reputation damage, leading to financial losses, legal liabilities, and customer trust erosion.

Vulnerability Path: http://testfire.net/
Vulnerability Parameter: http://testfire.net/login.jsp

Steps to Reproduce:

Identify a user input field that influences a downstream component, such as a search bar or a form.
Attempt to inject malicious code, like SQL, command injection, or other malicious payloads, into the input field.
Observe if the malicious code is executed by the downstream component.
If the malicious code is executed without proper neutralization, you have successfully reproduced the issue.



Online Banking Login

Syntax error: Encountered "\'\'" at line 1, column 55.

Username:
Password:

Recommendation:

The recommendation for addressing Improper Neutralization of Special Elements in Output Used by a Downstream Component ('Injection'), often related to CGI Generic Cross-Site Scripting (XSS), is to implement proper input validation, output encoding, and security controls in your web application to prevent untrusted data from being executed as code. This helps safeguard against XSS attacks, ensuring the integrity and security of the application.

6. Vulnerability Name:

Insufficiently Protected Credentials (Web Server Transmits Cleartext Credentials)

CWE: CWE-522
OWASP Category: A04:2021 – Identification and Authentication Failure

Description:
The product transmits or stores authentication credentials, but it uses an insecure method that is susceptible to unauthorized interception and/or retrieval.

Business Impact:
The business impact of Insufficiently Protected Credentials, such as a web server transmitting cleartext credentials, can result in severe security breaches, data theft, and reputation damage. This vulnerability can lead to unauthorized access to sensitive information and compromise customer trust, potentially causing financial losses and legal consequences.

Vulnerability Path: http://testfire.net/
Vulnerability Parameter: http://testfire.net/login.jsp

Steps to Reproduce:
Identify a login or authentication form in the target application.
Attempt to log in using an incorrect username and password.
Capture the network traffic or observe the response.
If the application's response provides clear information indicating which part of the login credentials was incorrect (e.g., "Invalid username"), you may have found insufficiently protected credentials.

Recommendation:
The recommendation for "Insufficiently Protected Credentials (Web Server Transmits Cleartext Credentials)" is to implement secure communication protocols like HTTPS and employ encryption to safeguard sensitive login credentials transmitted between the client and the web server.

7. Vulnerability Name:

Improper Handling of an Exceptional Condition (Web Server Transmits Cleartext Credentials)

CWE: CWE-928
OWASP Category: A04:2021 – Vulnerable and Outdated Components

Description:

software weakness where exceptional conditions, such as errors or unexpected events, are not handled or managed appropriately within the application's code. This weakness can manifest in various ways depending on the specific application and programming language being used.

Business Impact:

The business impact of improper handling of an exceptional condition where a web server transmits cleartext credentials can lead to significant security vulnerabilities, potential data breaches, damage to the organization's reputation, legal liabilities, and financial losses due to remediation and compensation.

Vulnerability Path: http://testfire.net/

Steps to Reproduce:

This CWE does not correspond to a specific security issue that can be easily reproduced in the same way as the others. It relates to improper error handling and exception management, which can manifest in various ways depending on the application. To identify such issues, you would need to analyse the application's code and error-handling mechanisms.

Recommendation:

The recommended solution for improper handling of an exceptional condition where a web server transmits cleartext credentials is to implement secure communication protocols, such as HTTPS, to encrypt the transmission of sensitive information and prevent unauthorized access.

8. Vulnerability Name:

Improper Handling of an Exceptional Condition (Web Server Transmits Cleartext Credentials)

CWE: CWE-930
OWASP Category: A02:2021 – Cryptographic Failures

Description:

software weakness where an application makes the incorrect assumption that certain data is immutable (i.e., it cannot be modified or changed) when, in reality, this data can be altered by external entities, including attackers, or even by the program itself.

Business Impact:

The business impact of improper handling of an exceptional condition, such as a web server transmitting cleartext credentials, can result in severe security breaches, damage to the organization's reputation, potential legal repercussions, and financial losses due to compromised user data and trust.

Vulnerability Path: http://testfire.net/

Steps to Reproduce:

This CWE relates to verifying cryptographic signatures properly and securely. Reproducing this CWE would involve analysing an application's cryptographic signature verification process to determine if it's performed incorrectly or inadequately. This may require code analysis and testing within the specific context of the application.

Recommendation:

The recommended solution for improper handling of an exceptional condition where a web server transmits cleartext credentials is to implement secure encryption protocols (e.g., HTTPS) to protect sensitive data during transmission and to handle exceptions gracefully, providing informative error messages without exposing sensitive information.

9. Vulnerability Name:

Insecure Direct Object Reference (IDOR)

CWE: cwe-639

OWASP Category: A01:2021 – Broken Access Control

Description:

The system's authorization functionality does not prevent one user from gaining access to another user's data or record by modifying the key value identifying the data.

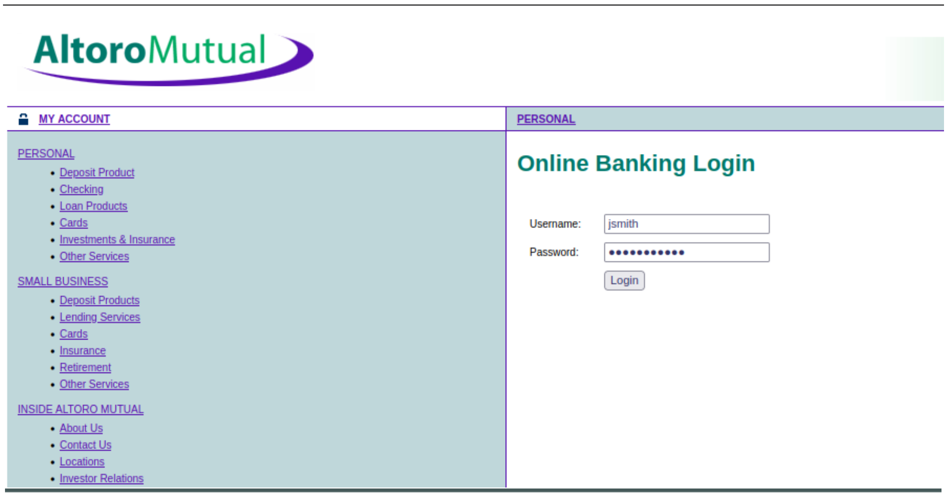
Business Impact:

IDOR can lead to unauthorized access to sensitive data or resources, potentially resulting in data breaches, privacy violations, financial losses, and damage to an organization's reputation. It can also lead to legal and regulatory consequences, impacting the overall trust and confidence in the business.

Vulnerability Path: http://testfire.net/

Steps to Reproduce:

Navigate into the given URL and login using john smith credentials. Click on “Go” to view John Smith’s savings. Change the listAccount=800002 to 800003 to view account history of other customers.



PERSONAL

- Deposit Product
- Checking
- Loan Products
- Cards
- Investments & Insurance
- Other Services

SMALL BUSINESS

- Deposit Products
- Lending Services
- Cards
- Insurance
- Retirement
- Other Services

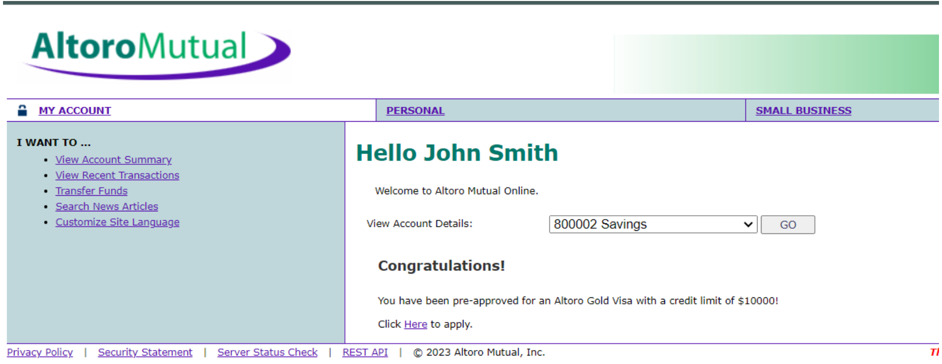
INSIDE ALTORO MUTUAL

- About Us
- Contact Us
- Locations
- Investor Relations

Online Banking Login

Username: jsmith

Password: ●●●●●●●●●●●



I WANT TO ...

- View Account Summary
- View Recent Transactions
- Transfer Funds
- Search News Articles
- Customize Site Language

Hello John Smith

Welcome to Altoro Mutual Online.

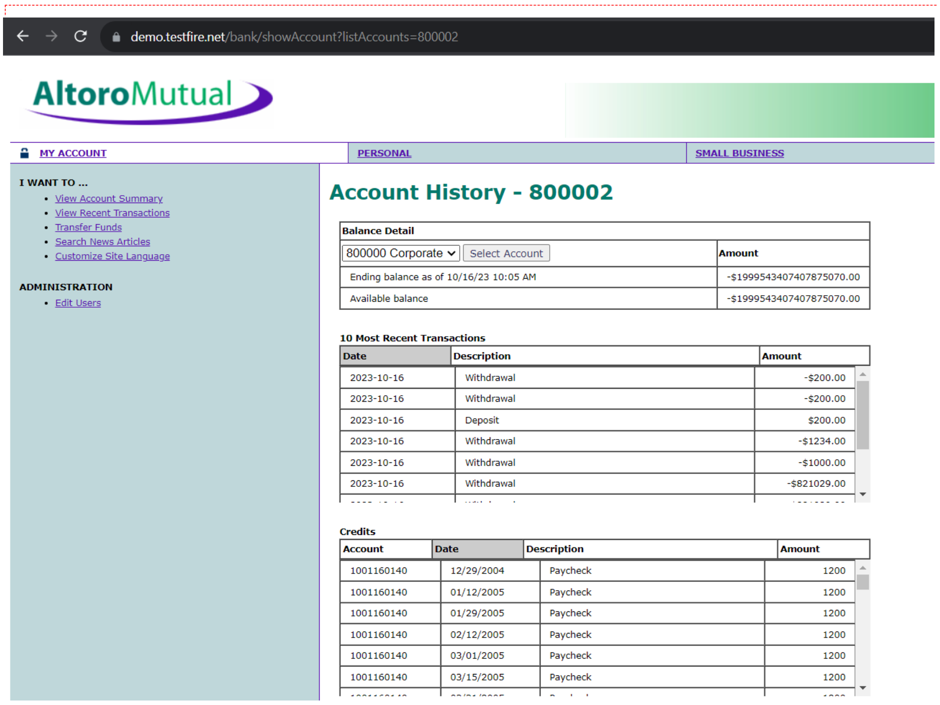
View Account Details: 800002 Savings

Congratulations!

You have been pre-approved for an Altoro Gold Visa with a credit limit of $10000!

Click Here to apply.

Privacy Policy | Security Statement | Server Status Check | REST API | © 2023 Altoro Mutual, Inc. Tl



demo.testfire.net/bank/showAccount?listAccounts=800002

I WANT TO ...

- View Account Summary
- View Recent Transactions
- Transfer Funds
- Search News Articles
- Customize Site Language

ADMINISTRATION

- Edit Users

Account History - 800002

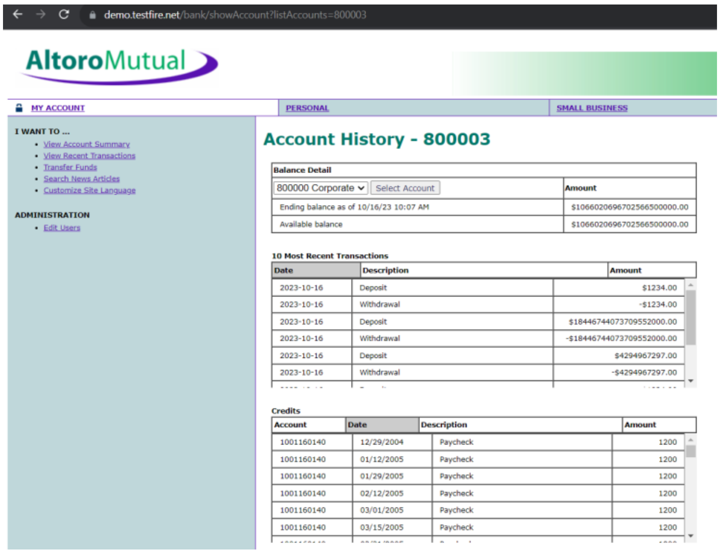
Balance Detail	
800000 Corporate ⌄ Select Account	Amount
Ending balance as of 10/16/23 10:05 AM	-$1999543407407875070.00
Available balance	-$1999543407407875070.00

10 Most Recent Transactions

Date	Description	Amount
2023-10-16	Withdrawal	-$200.00
2023-10-16	Withdrawal	-$200.00
2023-10-16	Deposit	$200.00
2023-10-16	Withdrawal	-$1234.00
2023-10-16	Withdrawal	-$1000.00
2023-10-16	Withdrawal	-$821029.00

Credits

Account	Date	Description	Amount
1001160140	12/29/2004	Paycheck	1200
1001160140	01/12/2005	Paycheck	1200
1001160140	01/29/2005	Paycheck	1200
1001160140	02/12/2005	Paycheck	1200
1001160140	03/01/2005	Paycheck	1200
1001160140	03/15/2005	Paycheck	1200



demo.testfire.net/bank/showAccount?listAccounts=800003

Account History - 800003

Balance Detail	
800000 Corporate	Amount
Ending balance as of 10/16/23 10:07 AM	$10660206967025665000000.00
Available balance	$10660206967025665000000.00

10 Most Recent Transactions

Date	Description	Amount
2023-10-16	Deposit	$1234.00
2023-10-16	Withdrawal	-$1234.00
2023-10-16	Deposit	$18446744073709552000.00
2023-10-16	Withdrawal	-$18446744073709552000.00
2023-10-16	Deposit	$4294967297.00
2023-10-16	Withdrawal	-$4294967297.00

Credits

Account	Date	Description	Amount
1001160140	12/29/2004	Paycheck	1200
1001160140	01/12/2005	Paycheck	1200
1001160140	01/29/2005	Paycheck	1200
1001160140	02/12/2005	Paycheck	1200
1001160140	03/01/2005	Paycheck	1200
1001160140	03/15/2005	Paycheck	1200

Recommendation:

The recommendation or solution for Insecure Direct Object Reference (IDOR) is to implement proper access controls and validation mechanisms to ensure that users can only access or modify the data and resources they are authorized to, and not directly reference internal objects. This can be achieved through role-based access control, session management, and thorough input validation.

10. Vulnerability Name:

Information Disclosure

CWE: cwe-200

OWASP Category: A03:2021 - Cryptographic Failures

Description:

The product exposes sensitive information to an actor that is not explicitly authorized to have access to that information.

Business Impact:

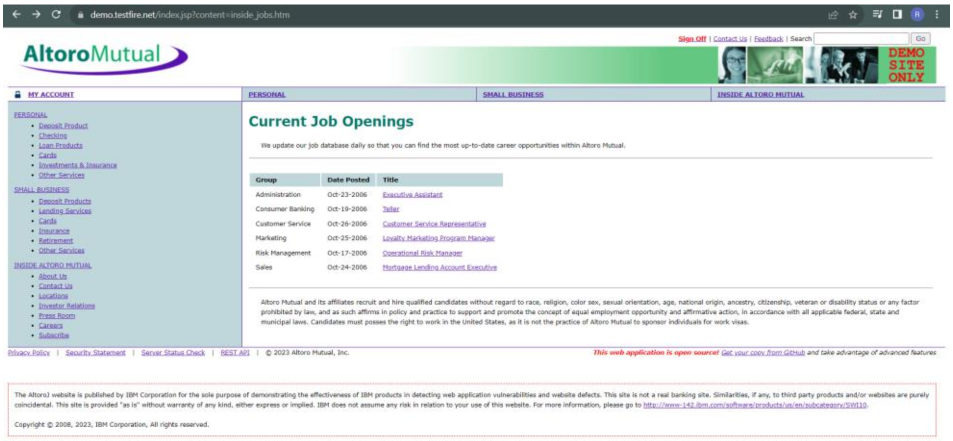
Information disclosure jeopardizes privacy, competitive standing, and trust, potentially resulting in legal actions, financial losses, and reputational harm, undermining an organization's security, prosperity, and image.

Vulnerability Path:

https://demo.testfire.net/index.jsp?content=inside_jobs.html

Steps to Reproduce:

Navigate to the URL and then the details are clearly visible.



demo.testfire.net/index.jsp?content=inside_jobs.htm

Sign Off | Contact Us | Feedback | Search Go

AltoroMutual

DEMO SITE ONLY

MY ACCOUNT | PERSONAL | SMALL BUSINESS | INSIDE ALTORO MUTUAL

PERSONAL
- Deposit Product
- Checking
- Loan Products
- Cards
- Investments & Insurance
- Other Services

SMALL BUSINESS
- Deposit Products
- Lending Services
- Cards
- Insurance
- Retirement
- Other Services

INSIDE ALTORO MUTUAL
- About Us
- Contact Us
- Locations
- Investor Relations
- Press Room
- Careers
- Subscribe

Current Job Openings

We update our job database daily so that you can find the most up-to-date career opportunities within Altoro Mutual.

Group	Date Posted	Title
Administration	Oct-23-2006	Executive Assistant
Consumer Banking	Oct-19-2006	Teller
Customer Service	Oct-26-2006	Customer Service Representative
Marketing	Oct-25-2006	Loyalty Marketing Program Manager
Risk Management	Oct-17-2006	Operational Risk Manager
Sales	Oct-24-2006	Mortgage Lending Account Executive

Altoro Mutual and its affiliates recruit and hire qualified candidates without regard to race, religion, color sex, sexual orientation, age, national origin, ancestry, citizenship, veteran or disability status or any factor prohibited by law, and as such affirms in policy and practice to support and promote the concept of equal employment opportunity and affirmative action, in accordance with all applicable federal, state and municipal laws. Candidates must posses the right to work in the United States, as it is not the practice of Altoro Mutual to sponsor individuals for work visas.

Privacy Policy | Security Statement | Server Status Check | REST API | © 2023 Altoro Mutual, Inc. *This web application is open source! Get your copy from GitHub and take advantage of advanced features*

The AltoroJ website is published by IBM Corporation for the sole purpose of demonstrating the effectiveness of IBM products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. IBM does not assume any risk in relation to your use of this website. For more information, please go to http://www-142.ibm.com/software/products/us/en/subcategory/SWI10.

Copyright © 2008, 2023, IBM Corporation, All rights reserved.

Recommendation:

Information disclosure refers to the unintentional release of sensitive or confidential data. To mitigate this risk, implement robust data security measures, including encryption, access controls, and employee training, and regularly audit and update security protocols to prevent unauthorized access and data leaks.

11. Vulnerability Name:

Web Application Potentially Vulnerable to Clickjacking

CWE: cwe-451

OWASP Category: A04:2021 – Insecure Design

Description:

The user interface (UI) does not properly represent critical information to the user, allowing the information - or its source - to be obscured or spoofed. This is often a component in phishing attacks.

Business Impact:

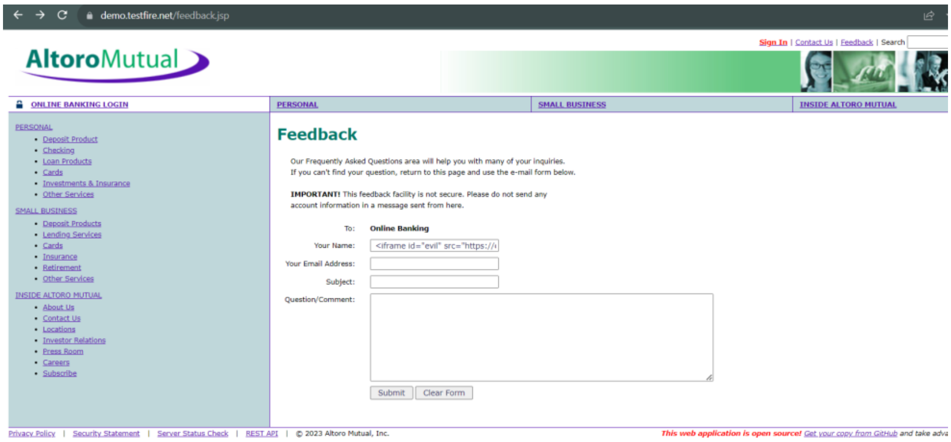
The Impact of Clickjacking The hacker has several ways they can use the redirected clicks for their own gain. A common form of clickjacking involves mirroring a login and password form on a website.

Vulnerability Path:

https://demo.testfire.net/feedback.jsp

Steps to Reproduce:

Navigate to the above-mentioned URL. Enter the payload “<iframe id="evil" src=https://evil.com sandbox="allow-forms"></iframe>.

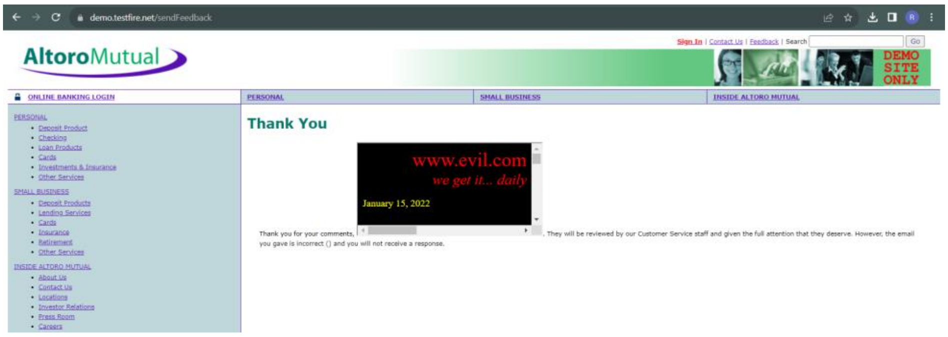


Feedback

Our Frequently Asked Questions area will help you with many of your inquiries.
If you can't find your question, return to this page and use the e-mail form below.

IMPORTANT! This feedback facility is not secure. Please do not send any account information in a message sent from here.

To: **Online Banking**
Your Name: <iframe id="evil" src="https://
Your Email Address:
Subject:
Question/Comment:



Thank You

Thank you for your comments, . They will be reviewed by our Customer Service staff and given the full attention that they deserve. However, the email you gave is incorrect () and you will not receive a response.

Recommendation:

Information disclosure refers to the unintentional release of sensitive or confidential data. To mitigate this risk, implement robust data security measures, including encryption, access controls, and employee training, and regularly audit and update security protocols to prevent unauthorized access and data leaks.

Vulnerabilities Report for Main Website

12. Vulnerability Name:

Protection Mechanism Failure (Web Application Potentially Vulnerable to Clickjacking)

CWE: CWE-693
OWASP Category: A04:2021 – Broken Access Control

Description:

This weakness covers three distinct situations. A "missing" protection mechanism occurs when the application does not define any mechanism against a certain class of attack. An "insufficient" protection mechanism might provide some defences - for example, against the most common attacks - but it does not protect against everything that is intended. Finally, an "ignored" mechanism occurs when a mechanism is available and in active use within the product, but the developer has not applied it in some code path.

Business Impact:
Protection Mechanism Failure in web applications potentially vulnerable to clickjacking can result in severe business impacts. Clickjacking is a malicious technique where attackers deceive users into performing unintended actions. When protection mechanisms fail, it can lead to unauthorized data exposure, financial losses, and reputational damage. This, in turn, can erode trust, harm customer relationships, and lead to legal consequences, making robust security measures crucial for preserving a business's integrity and success.

Vulnerability Path: https://vtop2.vitap.ac.in/vtop/initialProcess

Steps to Reproduce:
Identify a security feature or protection mechanism in the application (e.g., authentication, access control).
Attempt to bypass or defeat the security mechanism.
Observe whether the application allows unauthorized access or circumvention.
If you successfully bypass the protection mechanism, you have reproduced a protection mechanism failure issue.

Recommendation:
Return the X-Frame-Options or Content-Security-Policy (with the 'frame-ancestors' directive) HTTP header with the page's response.
This prevents the page's content from being rendered by another site when using the frame or iframe HTML tags.

13. Vulnerability Name:
Improper Input Validation (CGI Generic XSS)

CWE: CWE-20
OWASP Category: A04:2021 – Injection

Description:
The product receives input or data, but it does not validate or incorrectly validates that the input has the properties that are required to process the data safely and correctly.

Business Impact:
Improper input validation, specifically in the context of CGI scripts, can lead to Cross-Site Scripting (XSS) vulnerabilities. The business impact of this vulnerability includes

risks of data theft, unauthorized access, and damage to a company's reputation, potentially resulting in financial losses, legal liabilities, and loss of customer trust.

Vulnerability Path: https://vtop2.vitap.ac.in/vtop/initialProcess

Steps to Reproduce:

Identify a user input field in the target application, such as a search box or login form.
Attempt to enter malicious input, like SQL code, HTML tags, or special characters, in the input field.
Submit the input and observe if the application processes it without validation or sanitization.
If the application accepts and processes the input without any validation or sanitization, you have successfully reproduced the issue.

Recommendation:

The recommended solution for Improper Input Validation (CGI Generic XSS) is to implement robust input validation and sanitization mechanisms to prevent malicious code injection in web applications. This involves validating and encoding user inputs, utilizing security libraries, and employing web application firewalls to mitigate Cross-Site Scripting (XSS) vulnerabilities.

14. Vulnerability Name:

Deserialization of Untrusted Data ('Cross-site Scripting') (Apache Tomcat 9.0.0.M1)

CWE: CWE-502
OWASP Category: A08:2021 – Software & Data Integrity Failure

Description:

The product deserializes untrusted data without sufficiently verifying that the resulting data will be valid.

Business Impact:

The Deserialization vulnerabilities can lead to remote code execution, potentially allowing an attacker to execute arbitrary code on the server. This can result in data breaches, unauthorized access, and data manipulation, leading to reputational damage and legal consequences.

Vulnerability Path: https://vtop2.vitap.ac.in/vtop/initialProcess

Steps to Reproduce:

I. Identify the Target Application: Find a web application that uses deserialization to process incoming data, e.g., session data stored in cookies.

II. Create a Malicious Serialized Object: Craft a malicious serialized object that contains code you want to execute on the server.
III. Inject the Malicious Object: Inject the malicious serialized object into the application by sending it in a request. This might involve setting a cookie with the malicious data.
IV. Trigger Deserialization: Send a request to the application to trigger the deserialization process.
V. Exploit the Vulnerability: If the application does not properly validate the deserialized data, it may execute the malicious code contained within the serialized object.

Recommendation:

The recommendation for deserialization of untrusted data (Cross-site Scripting or XSS) is to implement strong input validation, sanitize user inputs, and use secure serialization libraries to prevent malicious code execution when processing data from untrusted sources.

15. Vulnerability Name:

Unrestricted Upload of File with Dangerous Type (Apache Tomcat 9.0.0.M1)

CWE: CWE-434
OWASP Category: A05:2021 – Security Misconfiguration

Description:

The product allows the attacker to upload or transfer files of dangerous types that can be automatically processed within the product's environment.

Business Impact:

Allowing users to upload and execute dangerous file types can lead to server compromise, data loss, and unauthorized access to sensitive information. This may result in legal and regulatory consequences.

Vulnerability Path: https://vtop2.vitap.ac.in/vtop/initialProcess

Steps to Reproduce:

I. Identify the File Upload Functionality: Find a web application that allows users to upload files, such as images or documents.
II. Create a Malicious File: Create a malicious file, such as a PHP script, and give it an innocuous name and extension that the application allows (e.g., "innocent.jpg").
III. Upload the Malicious File: Use the application's file upload functionality to upload the malicious file. The application may not properly validate the file type.

IV. Access the Uploaded File: Access the uploaded file through the application. Depending on the application's security controls, you may be able to execute the malicious code contained in the uploaded file, e.g., by accessing it through a URL.

Recommendation:

The recommendation or solution for unrestricted file uploads of dangerous types is to implement strict file type validation and content scanning to prevent malicious files from being uploaded.

16. Vulnerability Name:

Improper Certificate Validation (Apache Tomcat 9.0.0.M1)

CWE: CWE-295

OWASP Category: A02:2021 – Cryptographic failures

Description:

The product does not validate, or incorrectly validates, a certificate.

Business Impact:

The Improper certificate validation can expose users to man-in-the-middle attacks and unauthorized data access. This can harm trust in the application, leading to loss of customers and legal consequences.

Vulnerability Path: https://vtop2.vitap.ac.in/vtop/initialProcess

Steps to Reproduce:

I. Set Up a Man-in-the-Middle Attack: Intercept network traffic between a client and the server using tools like Wireshark to act as a man-in-the-middle.

II. Present a Forged Certificate: Generate a self-signed or forged SSL/TLS certificate to impersonate the server. Present this certificate during the SSL/TLS handshake.

III. Intercept and Modify Traffic: Intercept the SSL/TLS handshake between the client and server, presenting the forged certificate during the handshake.

IV. Exploit the Vulnerability: If the application does not properly validate the server's certificate, the client may accept the forged certificate, allowing you to intercept and modify traffic between the client and server.

Recommendation:

The recommendation for improper certificate validation is to ensure that digital certificates are thoroughly validated during authentication processes, using up-to-date and secure methods, to prevent unauthorized access and protect against security vulnerabilities.

17. Vulnerability Name:

Uncontrolled Resource Consumption (Apache Tomcat 9.0.0.M1)

CWE: CWE-400
OWASP Category: A08:2021 – Software & Integrity Failure

Description:
The product does not properly control the allocation and maintenance of a limited resource, thereby enabling an actor to influence the number of resources consumed, eventually leading to the exhaustion of available resources.

Business Impact:
Uncontrolled resource consumption can lead to denial of service (DoS) attacks, causing system downtime, loss of revenue, and reputational damage.

Vulnerability Path: https://vtop2.vitap.ac.in/vtop/initialProcess

Steps to Reproduce:

I. Identify the Target Application: Find an application that handles resource-intensive operations.

II. Generate Resource-Intensive Requests: Create and send a large number of resource-intensive requests to the application. For example, sending multiple requests that require significant CPU, memory, or network bandwidth.

III. Observe Resource Exhaustion: Monitor the application to observe how it handles the resource-intensive requests. If the application doesn't have proper resource controls, it may become slow or unresponsive, leading to a potential DoS condition.

Recommendation:
The recommendation for uncontrolled resource consumption is to implement strict resource monitoring and management protocols to ensure that resources such as CPU, memory, and bandwidth are used efficiently and within predefined limits. This may involve using resource management tools, setting usage quotas, and optimizing code or configurations to prevent excessive resource consumption.

18. Vulnerability Name:

Improper Locking (Apache Tomcat 9.0.0.M1)

CWE: CWE-667
OWASP Category: A07:2021 – Identification &Authentication Failure

Description:
The product does not properly acquire or release a lock on a resource, leading to unexpected resource state changes and behaviours.

Business Impact:

Using weak or broken cryptographic algorithms can lead to data breaches, compromised confidentiality, and loss of trust in the security of the application.

Vulnerability Path: https://vtop2.vitap.ac.in/vtop/initialProcess

Steps to Reproduce:

Reproducing this vulnerability typically involves analysing the application's use of cryptography and identifying instances where weak or broken algorithms are used. Exploiting it would depend on the specific cryptographic context in the application.

Recommendation:

The Improper locking is a software issue were multiple threads or processes access shared resources without proper synchronization. The recommended solution is to implement thread-safe mechanisms, such as mutexes or semaphores, to ensure exclusive access to shared data, preventing data corruption or race conditions.

19. Vulnerability Name:

Session Fixation (Apache Tomcat 9.0.0.M1)

CWE: CWE-384
OWASP Category: A07:2021 – Identification &Authentication Failure

Description:

Authenticating a user, or otherwise establishing a new user session, without invalidating any existing session identifier gives an attacker the opportunity to steal authenticated sessions.

Business Impact:

Session fixation can lead to unauthorized access, data exposure, and identity theft, resulting in reputational damage and legal consequences.

Vulnerability Path: https://vtop2.vitap.ac.in/vtop/initialProcess

Steps to Reproduce:

I. Identify the Target Application: Find a web application that uses session management.
II. Obtain a Session ID: Obtain a valid session ID from the application, either by registering an account or using an existing session.
III. Share the Session ID: Share the valid session ID with the victim, for example, by sending a link with the session ID as a URL parameter.
IV. Victim Uses the Session: Encourage the victim to use the session by clicking on the link or accessing the application with the provided session ID.

V. Exploit the Vulnerability: If the application does not properly handle session management and associates the session ID provided by the attacker with the victim's account, the attacker can impersonate the victim.

Recommendation:
The Session fixation is a security vulnerability where an attacker can fix or set a user's session ID, potentially gaining unauthorized access to the user's account. To mitigate session fixation, implement robust session management practices, such as generating a new session ID after login, using secure session cookies, and regularly rotating session IDs during user interactions to make it harder for attackers to predict or fix a session ID.

20. Vulnerability Name:

Information Disclosure

CWE: CWE-200
OWASP Category: A03:2021 - Cryptographic Failures

Description:
The product exposes sensitive information to an actor that is not explicitly authorized to have access to that information.

Business Impact:
Information disclosure jeopardizes privacy, competitive standing, and trust, potentially resulting in legal actions, financial losses, and reputational harm, undermining an organization's security, prosperity, and image.

Vulnerability Path: https://vtop2.vitap.ac.in/vtop/initialProcess

Steps to Reproduce:

I. Identify the vulnerable system or code.
II. Reproduce the issue by capturing relevant data or using known attack vectors.
III. Analyse the results to confirm the information disclosure.
IV. Fix the vulnerability by patching, modifying code, or adjusting configurations.
V. Test the fix thoroughly, monitor for any recurrence, and document the process.

Recommendation:
Information disclosure refers to the unintentional or unauthorized release of sensitive data. To prevent it, organizations should implement robust cybersecurity measures, including encryption, access controls, regular security audits, and employee training, to safeguard sensitive information and ensure compliance with data protection regulations.

Proposed Solution:

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	1. The prevalence of malware attacks and their impact on organizations and individuals. 2. The limitations of existing malware detection methods. 3. The need for a more efficient, accurate, and scalable solution to combat malware threats.
2.	Idea / Solution description	1. The use of machine learning and deep learning models for malware classification. 2. Real-time data acquisition and processing to identify malware in network traffic and files. 3. Integration with existing security infrastructure for proactive threat detection. 4. User-friendly reporting and alerting interfaces to facilitate quick responses to potential threats.
3.	Novelty / Uniqueness	1. Our solution leverages state-of-the-art deep learning architectures, such as custom-built Convolutional Recurrent Neural Networks (CRNN), optimized for the unique features of malware files. 2. We have developed novel ensemble models that combine multiple machine learning algorithms to achieve higher accuracy in classifying both known and unknown malware variants. 3. Our proprietary threat intelligence database contains a vast repository of malware samples, allowing us to enrich data and detect patterns that other solutions may miss. 4. Adaptability to Evolving Malware Tactics 5. Low False Positive Rate 6. In addition to its technical capabilities, our solution provides a user-friendly web-based interface with interactive dashboards that allow security teams

		to visualize and manage detected threats easily 7. 7. We offer scalable deployment options and can seamlessly integrate with cloud services to accommodate growing data loads and ensure uninterrupted protection for organizations of all sizes.
4.	Social Impact / Customer Satisfaction	1. Effective Threat Prevention: Our solution prevents data breaches and financial losses by proactively identifying and mitigating malware threats before they can cause harm. 2. Efficiency in Detection: It reduces the time and effort required for manual malware analysis through automated and high-throughput processes. 3. 3.Improved Productivity: Users can focus on their core tasks, leading to increased productivity and reduced stress, as your solution takes care of routine malware detection.
5.	Business Model (Revenue Model)	1. 1.Subscription-based model: Charging users on a recurring basis for access to your service. 2. 2.Licensing model: Licensing the software to organizations for a one-time fee. 3. Freemium model: Offering a basic version for free and charging for premium features. 4. Service-based model: Providing ongoing support and maintenance services for a fee.
6.	Scalability of the Solution	Cloud-Based Infrastructure: Our solution is built on a robust cloud-based infrastructure that offers exceptional scalability. It leverages the elasticity of cloud computing, allowing us to easily accommodate surges in data loads and user activity. With this approach, we can dynamically allocate computing resources as needed, ensuring optimal performance during peak periods. Multi-Environment Deployment: We have engineered our solution to be highly flexible, enabling deployment across a wide range of

		environments. It caters to the diverse needs of small businesses, large enterprises, and everything in between. Whether deployed on a single server or across a distributed network of servers, our solution adapts seamlessly to its surroundings.

Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Building an efficient detection system requires gathering samples of both malicious and benign software from several sources and properly classifying them with regard to the different categories of malware.
- To improve model performance, it is crucial to extract pertinent features from malware samples and carry out feature selection/extraction.
- Malware detection is based on the creation and training of machine learning models, which can combine deep learning and conventional ML models.
- Reliable results are guaranteed when cross-validation is used in conjunction with proper assessment measures to assess model performance.
- For the detection system to be available, a suitable deployment platform—such as on-premises servers or cloud services—must be chosen.
- It is crucial to have strong security measures in place to guard against adversarial assaults on the models and private information.
- System dependability depends on continuously checking the deployed models for performance deterioration or concept drift and setting up warning mechanisms for quick response.
- Ensuring smooth operation under changing loads is achieved by applying load balancing and designing the architecture to manage varying traffic volumes.
- Usability and functionality are increased by creating an intuitive user interface, offering choices for uploading files or URLs for scanning, and including APIs for integration.
- Legal and operational integrity depend on a number of factors, including making sure that all applicable rules and regulations are followed, keeping accurate records, and educating analysts and administrators.

Example - Solution Architecture Diagram:

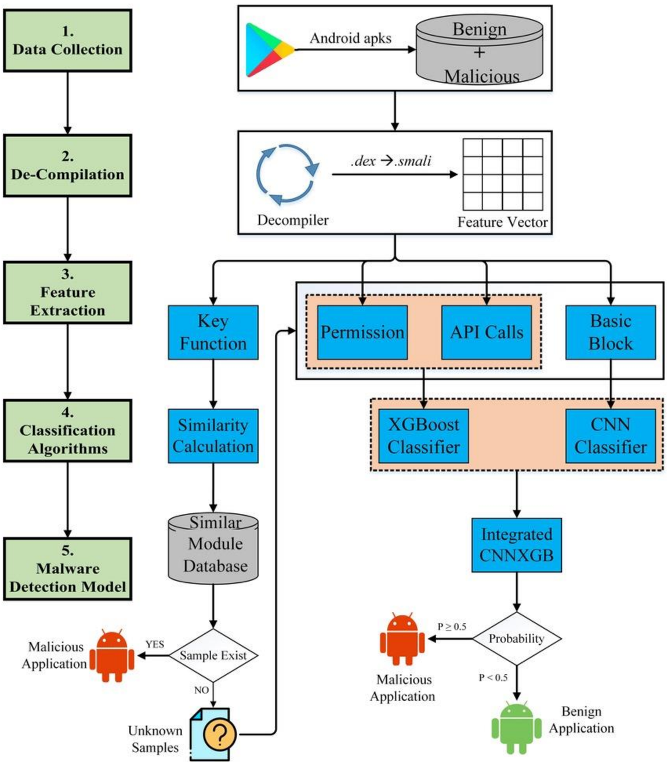


Figure 1: Architecture and data flow of the Malware Detection and Classification

Reference: https://www.researchgate.net/figure/System-Architecture-of-the-Android-malware-detection-using-hierarchical_fig1_349162676

Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the rightamount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

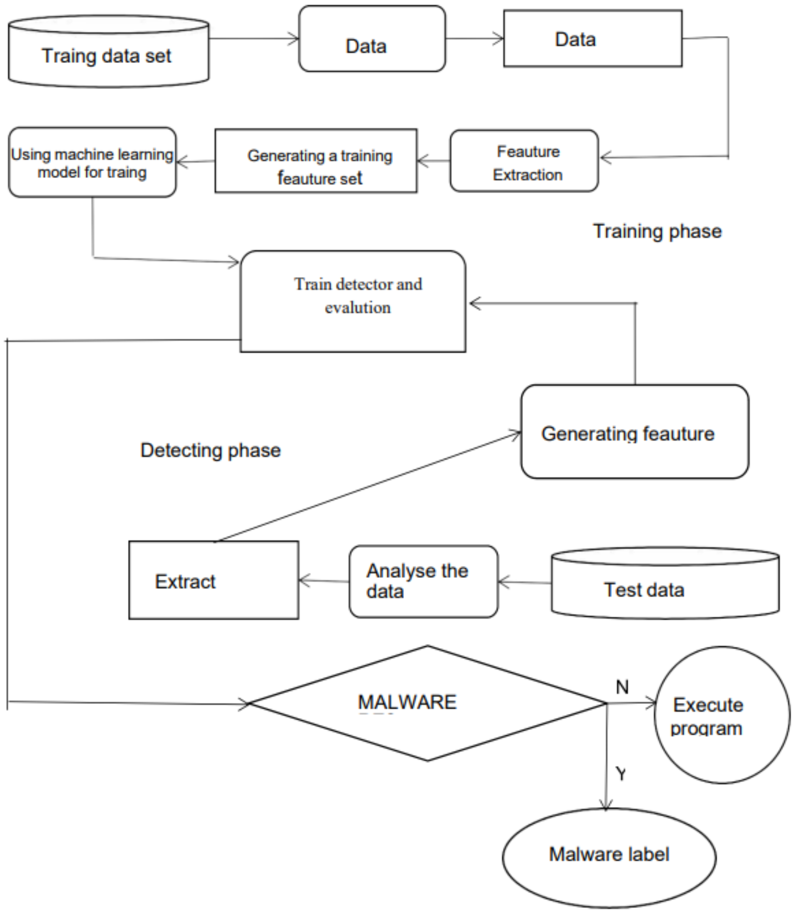
DATA FLOW:

- Data is acquired from various sources, including network traffic logs, file systems, and data feeds.
- The acquired data undergoes preprocessing to clean, transform, and normalize it.
- In the feature extraction layer, relevant features are extracted and transformed into feature vectors.
- These feature vectors are fed into the machine learning model layer, where the

classification takes place.

- Real-time analysis and detection determine whether the incoming data contains malware.
- If malware is detected, the reporting and alerting interface issues alerts and notifications to relevant parties.
- The deployment and scalability layer ensures the system can accommodate growing data loads and users.
- Security and compliance measures are maintained throughout the process to protect the system and data.
- The research and development layer ensures the system's adaptability to evolving malware threats through continuous updates and improvements.

DATA FLOW DIAGRAM OF PROJECT:



User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Customer_Registration	USN-1	I may register for the application as a user by providing my email address, creating a password, and verifying it.	Can login to my account	High	Sprint-1
		USN-2	Setting of Multi factor authentication (MFA) for assuring more security	Downloading respective app (Microsoft or google Auth)	High	Sprint-1
		USN-3	As a user, I will receive confirmation email once I have registered for the application	Redirecting to given link	High	Sprint-1
	Login	USN-4	As a user, I can log into the application by entering email, password and MFA	Render to home page	High	
	Scan	USN-5	Selecting options and uploading of selected files	Upload from drive or device and can scan	High	Sprint-2
		USN-6	Completion of scan and report generation	View and download the report	High	Sprint-2
		USN-7	Downloading and backing up the report	Report backup	Medium	Sprint-2
	Awareness	USN-8	As a user, I can know the details of the malware in the file uploaded. I can be aware of the malware.	Malware awareness	High	Sprint-3
		USN-9	Suggesting of related articles and blogs of malware.	Knowledge regarding malware	Low	Sprint-3
	Exploring	USSN-10	Exploring various features in web	Knowing about more accessibilities	low	Sprint-3

	Logout	USN-9	Logging out user account	I can logout of the website.	Medium	Sprint-3
Customer (Mobile)	Customer_Registration	USN-1	I may register for the application as a user by providing my email address, creating a password, and verifying it.	Can login to my account	High	Sprint-1
		USN-2	Setting of Multi factor authentication (MFA) for assuring more security	Downloading respective app (Microsoft or google Auth)	High	Sprint-1
		USN-3	As a user, I will receive confirmation email once I have registered for the application	Redirecting to given link	High	Sprint-1
	Login	USN-4	As a user, I can log into the application by entering email, password and MFA	Render to home page	High	Sprint-1
	Scan	USN-5	Selecting options and uploading of selected files	Upload from drive or device and can scan	High	Sprint-2
Corporate Companies	Upload	USN-11	As a corporate user, I can receive real-time alerts and notifications when potential threats are detected in the files I've uploaded.	This enhances the security of my uploaded data.	High	Sprint-4
Admin	Maintenance	Admin	As an admin, I can perform regular security audits and vulnerability assessments on the website to ensure it remains resilient against emerging threats	Global modulator	High	Sprint-1

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

Project-Malware detection and classification

Reference: https://www.researchgate.net/figure/Process-flowchart-of-the-malware-analysis-process-during-the-experiment-based-on-Banin-et_fig1_337901093

FLOWCHART:

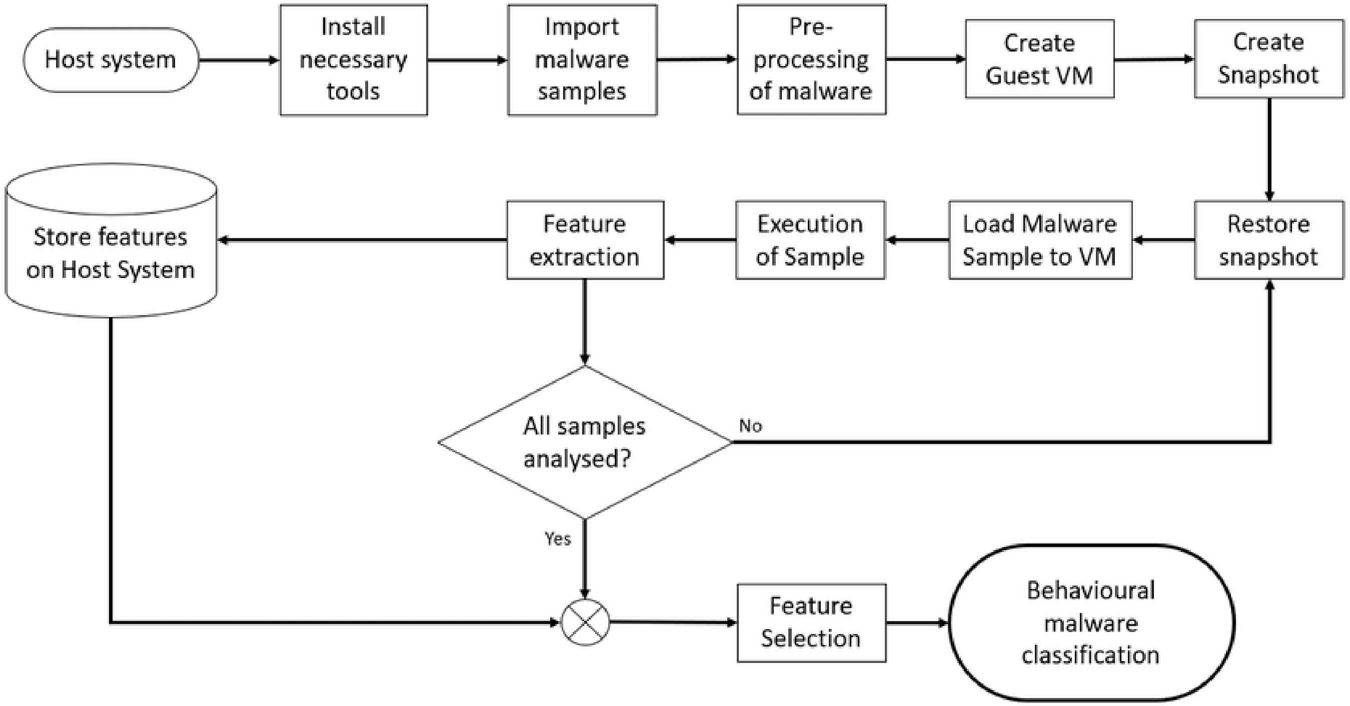


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web app	Python, Django
2.	Data Collection	Gathering raw data sources	Malware samples, network traffic, Kaggle
3.	Data Preprocessing	Cleaning and feature extraction	Removing duplicates, extracting features
4.	Machine Learning	Developing and serving ML models	Python, scikit-learn, TensorFlow
5.	Database	Data Type, Configurations etc.	MySQL
6.	Cloud Database	Database Service on Cloud	AWS Cloud services
7.	File Storage	File storage requirements	AWS S3 Bucket
8.	External API-1(REST)	Acts as a bridge for malware data	Rest API, etc.
9.	External API-2(SOAP)	Acts as a bridge for malware data	SOAP API, etc.
10.	Machine Learning Model	Training and evaluating model	TensorFlow, karas, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / CloudLocal Server Configuration: Cloud Server Configuration	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Malware Detection Algorithm	Utilizes deep learning neural networks to identify malware based on behavioral patterns.	Convolutional Neural Networks (CNNs)
2.	Security Implementations	Data encryption, firewall, Secure file handling, Access control	e.g., SHA-256, Encryptions, IAM Controls, OWASP etc.
3	Feature Extraction	Extracts file headers, byte sequences, and API call sequences for analysis.	Heuristic Analysis, Behavior-based Analysis

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Project setup & Infrastructure	USN-1	Set up the development environment with the required tools and frameworks to start the garbage classification project.	1	High	Abhiram
Sprint-1	development environment	USN-2	Gather a diverse dataset of images containing different types of garbage (plastic, paper, glass, organic) for training the deep learning model.	2	High	Hema Sagar
Sprint-2	Data collection	USN-3	Preprocess the collected dataset by resizing images, normalizing pixel values, and splitting it into training and validation sets.	2	High	Sruta Keerthi

3.	Scalable Architecture	The system is built on a cloud-based infrastructure, allowing automatic scaling based on demand.	AWS, Kubernetes for container orchestration
4.	Availability	Ensures high availability through redundant servers and load balancing. Also employs WAFs to protect against attacks.	Redundant Servers, F5 BIG-IP WAF, Load Balancers
5.	Performance	Uses in-memory caching for frequently accessed data, employs parallel processing for faster analysis, and has optimized intrusion detection algorithms	Redis for caching, Multi-threading, Snort IDS with custom rulesets

Sprint-2	data preprocessing	USN-4	Explore and evaluate different deep learning architectures (e.g. ., CNNs) to select the most suitable model for garbage classification.	3	High	Navya
Sprint-3	model development	USN-5	train the selected deep learning model using the pre-processed dataset and monitor its performance on the validation set.	4	High	Hema Sagar
Sprint-3	Training	USN-6	implement data augmentation techniques (e.g., rotation, g) flipping to improve the model's robustness and accuracy.	6	medium	Navya
Sprint-4	model deployment & Integration	USN-7	deploy the trained deep learning model as an API or web service to make it accessible for garbage classification. integrate the model's API into a user-friendly web interface for users to up images and receive garbage classification results.	1	medium	Abhiram

Sprint-5	Testing & quality assurance	USN-8	conduct thorough testing of the model and web interface to identify and report any issues or bugs. fine-tune the model hyperparameters and optimize its performance based on user feedback and testing results.	1	medium	Sruta Keerthi

References:

https://c4model.com/

https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/
https://www.ibm.com/cloud/architecture
https://aws.amazon.com/architecture

https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d

Product Backlog, Sprint Schedule, and Estimation:

Use the below template to create product backlog and sprint schedule

Project Tracker, Velocity & Burndown Chart:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	3	2 Days	4 Oct 2023	6 Oct 2023	20	3 Sep 2023
Sprint-2	5	4 Days	6 Oct 2023	10 Oct 2023		
Sprint-3	10	6 Days	10 Oct 2023	16 Oct 2023		

Sprint-4	2	8 Days	16 Oct 2023	24 Oct 2023		
Sprint-5	1	6 Days	24 Oct 2023	30 Oct 2023		

Velocity:

Imagine we have a 29-days sprint duration, and the velocity of the team is 20 (points per sprint). Let’s calculate the team’s average velocity (AV) per iteration unit (story points per day)

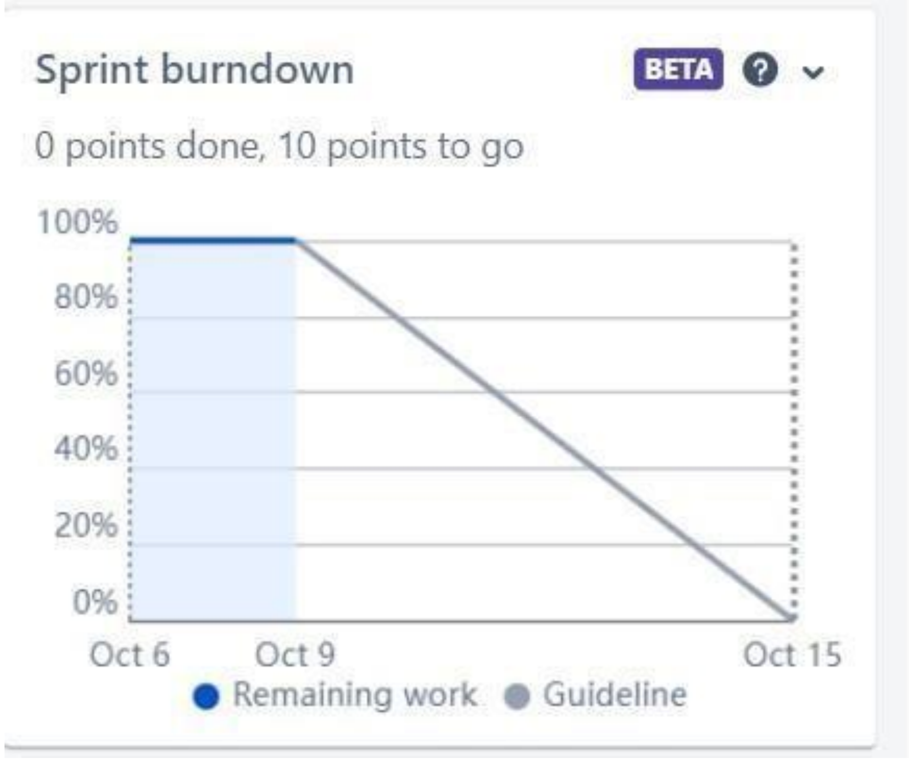
$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

AV= 26/20 = 1.35

Burndown Chart:

A burndown chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

Burndown Chart:

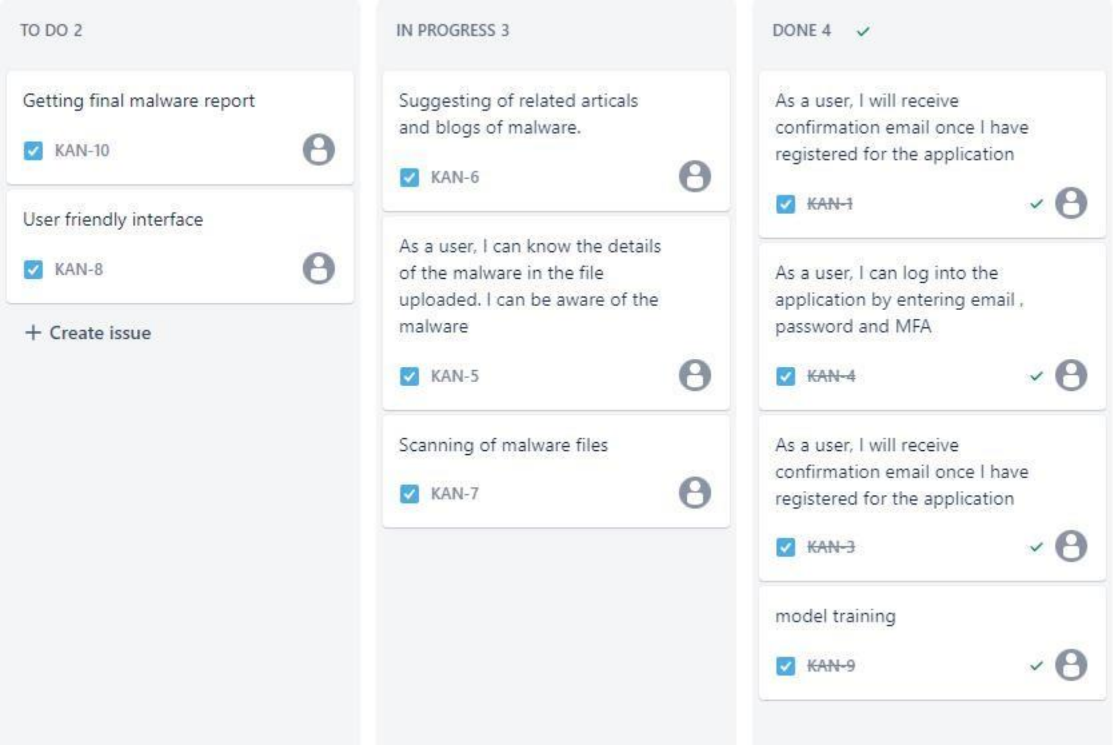


Sprint burndown

0 points done, 10 points to go

Board section.

We have completed sprint 1 and 2. So we can see the remaining tasks on board.



TO DO 2

Getting final malware report

☑ KAN-10

User friendly interface

☑ KAN-8

IN PROGRESS 3

Suggesting of related articals and blogs of malware.

☑ KAN-6

As a user, I can know the details of the file uploaded. I can be aware of the malware

☑ KAN-5

Scanning of malware files

☑ KAN-7

DONE 4 ✓

As a user, I will receive confirmation email once I have registered for the application

☑ ~~KAN-1~~ ✓

As a user, I can log into the application by entering email , password and MFA

☑ ~~KAN-4~~ ✓

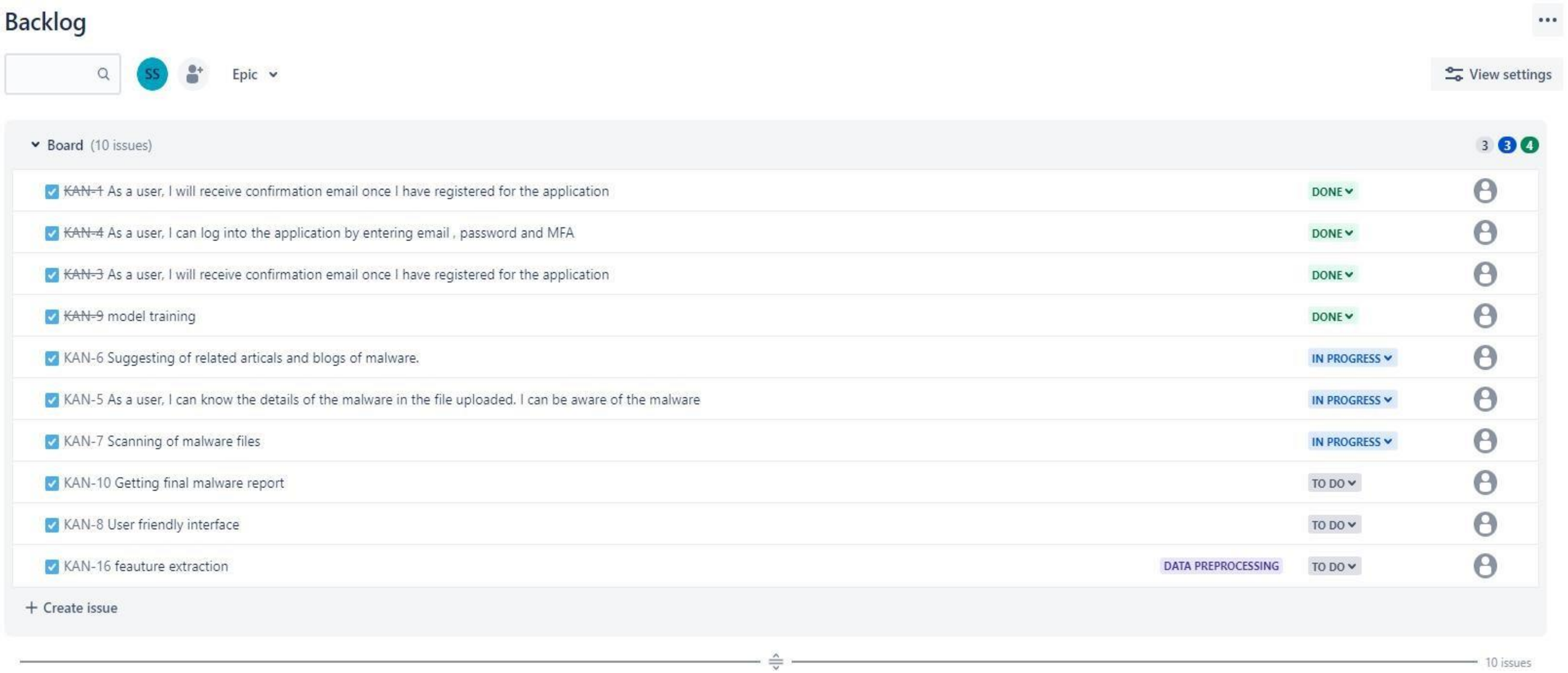
As a user, I will receive confirmation email once I have registered for the application

☑ ~~KAN-3~~ ✓

model training

☑ ~~KAN-9~~ ✓

Backlog section



Backlog

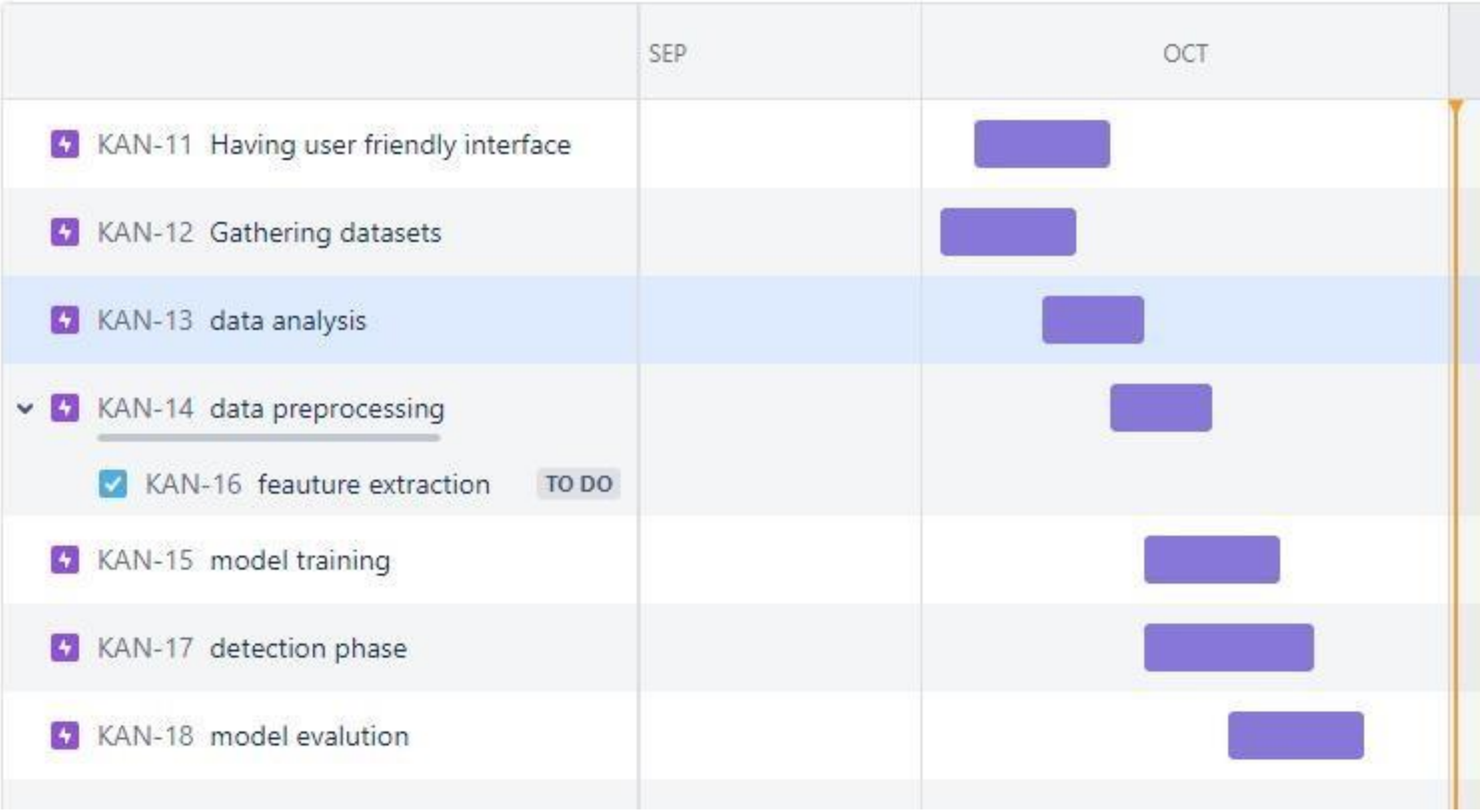
Board (10 issues) 3 3 4

☑	~~KAN-1~~ As a user, I will receive confirmation email once I have registered for the application		DONE
☑	~~KAN-4~~ As a user, I can log into the application by entering email , password and MFA		DONE
☑	~~KAN-3~~ As a user, I will receive confirmation email once I have registered for the application		DONE
☑	~~KAN-9~~ model training		DONE
☑	KAN-6 Suggesting of related articals and blogs of malware.		IN PROGRESS
☑	KAN-5 As a user, I can know the details of the malware in the file uploaded. I can be aware of the malware		IN PROGRESS
☑	KAN-7 Scanning of malware files		IN PROGRESS
☑	KAN-10 Getting final malware report		TO DO
☑	KAN-8 User friendly interface		TO DO
☑	KAN-16 feauture extraction	DATA PREPROCESSING	TO DO

+ Create issue

10 issues

Timeline



KAN-11 Having user friendly interface

KAN-12 Gathering datasets

KAN-13 data analysis

KAN-14 data preprocessing

☑ KAN-16 feauture extraction TO DO

KAN-15 model training

KAN-17 detection phase

KAN-18 model evalution

Team Members:

Naladala Navya

Sanisetty Hema Sagar

Kurra Naveen Abhiram

Vishnubhatla V L Sruta Keerthi