MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages: 2232-2239

PATAC: A pattern analysis model for securing web application against broken authentication & access control vulnerabilities.

Miss.Amrita A.Athalye[1*], Dr.Madhuri N. Gedam[2], Mrs.Supriya Singh[3]
Asst.Prof, Dept. of Information Technology, Shree L.R Tiwari College of Engineering, Mumbai, India
*Corresponding Author Email: amrita.pathak@slrtce.in[1]
Associate.Prof, Dept. of Information Technology, Shree L.R Tiwari College of Engineering, Mumbai, India
Email:madhuri.gedam@slrtce.in[2]
Asst.Prof, Dept. of Information Technology, Shree L.R Tiwari College of Engineering, Mumbai, India
Email:supriya.singh@slrtce.in [3]

Abstract

Authentication & access control are the initial line of defense against attacks on software-as-a-service (SaaS) cloud deployments. But due to improper software design, these defenses have various loop holes, which allows the attacker to inject attacks like Trojan horse, conditional vulnerabilities, etc. In order to reduce the probability of these attacks, this text proposes design of a novel header-scanning based pattern analysis model for detecting and mitigating broken authentication & access control attacks. The model is deployed at header of the cloud, which allows it to analyze all incoming requests before they are processed by the SaaS model. Due to this header-level deployment, the model is able to detect and mitigate almost 99.8% of authentication & access control vulnerabilities, including elastic attacks. A testbed for attack injection and analysis is also proposed, which allows the system model to be tested & validated for different kinds of vulnerabilities. The proposed model is also equipped with detection & removal of SQL (structured query language) injection, XSS (cross site scripting), and distributed denial of service (DDoS) attacks. In order to estimate performance of the proposed model, it was compared with various state-of-the-art attack detection & mitigation methods. Due to header-level pattern analysis, the proposed model was able to outperform recent state-of-the-art methods in terms of accuracy of attack detection, precision, recall, and delay performance. The pattern analysis layer utilizes temporal behaviour of client nodes along with their real time request-response behaviour in order to estimate probability of attacks. In order to further improve attack detection performance & scalability of the proposed model, this text proposes various improvements that can be used to extend the existing pattern analysis approach.

Keywords: Authentication, access control, pattern analysis, header-level, SaaS

Introduction

Cloud deployments require a wide variety of security measures, which include but are not limited to, authentication, access control, key-exchange, data privacy, user privacy, route privacy, authenticity of data, etc. In order to design these security measures, a wide variety of computationally complex security models are designed. Performance of these models depends upon number of attacks detected & mitigated by them, and their response time. Out of these attack detection models; authentication & access control provide an initial line of defense against a wide variety of attacks. It is observed that, a cloud deployment with proper authentication & access control mechanisms can avoid almost 80% of all attacks [1] before they affect working of the cloud. A typical authentication & access control model can be observed from figure 1, wherein 4 types of operations are performed,

i. Client requests for login page, and passes credentials like username, password, one time password (OTP), biometric information, etc.
ii. Server responds with a success code & stores client information in the form of session variables, or rejects the requests & redirects client to the login page.
iii. Upon successful authentication, an access token is generated & shared with the client.
iv. Client uses this access token to perform read/write operations on the cloud. If the access token is invalid, client is redirected to the login page, and its server session is terminated.

Based on this design, a wide variety of system models are proposed by researchers, which modify type of login, session storage, access token usage, etc. A brief survey of some of the recently proposed models is done for estimating their attack detection & mitigation performance, and can be observed from the next section of this text. This section is followed by design of the proposed PATAC: A pattern analysis model for securing web application against broken authentication & access control vulnerabilities. Performance evaluation & comparison of the proposed model is done in order to estimate its attack detection & quality of service (QoS) capabilities w.r.t. standard access control & authentication methods. Finally, this text concludes with some interesting observations about the proposed model, and recommends various methods to further improve its performance.

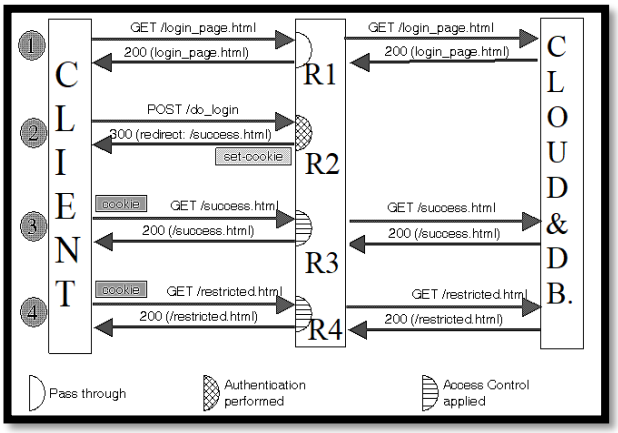


Fig 1. A typical application-level authentication and access control implementation

Literature Review

Assessing verification and access control vulnerabilities requires planning of numerous solicitations, checking and approval models, which should work to identify and moderate these vulnerabilities. For example, the work in [1] proposes an unchanging, straightforward, recognizable, and dispersed block chain model that utilizations trust access verification. This model reduces number of vulnerabilities during validation, and can follow inappropriate post confirmation client conduct to further develop framework security. The proposed model uses Proof of Concept (PoC), and plays out all agreement with the assistance of verifier hubs that have high trust levels. These conventions can be tried for weakness examination utilizing the work in [2], wherein different test models are executed for the framework, and its security execution is assessed. This work can be combined with [3], wherein a Multidimensional access control model is characterized. This model uses synergistic confirmation, through User Type Approval (UTA), and consolidates it with separated help security for access control. The model exhibits high precision, and moderate postponement of validation and access control when contrasted with cutting-edge models. This framework can recognize security weaknesses in a wide assortment of conditions, including Internet of Things (IoT) as talked about in [4]. The proposed model in [4] uses elliptic curve cryptography (ECC) and combines it with key arrangement convention for fusing access control and secure confirmation. This model must be combined with the work in [5], wherein character-based marks are utilized for access confirmation in IPv6 Networks. The proposed model is applied to vehicular deployments, however can be utilized basically for any continuous cloud network.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages: 2232-2239

This work can be further developed utilizing biometric confirmation structures as proposed in [6], wherein Telecare Medicine Information System (TMIS) is utilized as an application. The biometric model has an accuracy of more than 95%, which is modestly secure, and can be utilized for household security frameworks. The model in [7] gives a decent access control system design, yet can be worked on as far as validation through the work in [7] is performed, wherein multifaceted models are depicted. These models incorporate username and secret key mix, email and secret word mix, telephone number and OTP mix, and so forth, which permits cloud deployments to approve clients through standard components of verification. Comparable models are proposed in [8][9][10][11][12] wherein trust based verification, multimodal biometric validation, practically unclonable functions (PUFs), climate mindful zero-exertion two-factor confirmation systems, and fuzzy vector marks are proposed to give high security validations, by means of coordinating client input design.

Elite confirmation models can likewise be seen from [13][14] [15][16] wherein normal validation models utilizing incremental developments, block chain-based verification models utilizing high detect ability, shared confirmation for limitation arranged deployments, and two-way correspondence models for programming characterized optical deployments are discussed. These models assist in further developing cloud security by means of novel validation techniques that give detectability, high trust, and low postponement for a wide assortment of utilization areas. These models can be combined with the work in [17][18][19][20] for giving better access control and better verification by means of a characteristic level plan, utilizing block chains, information conglomeration through access control instruments, dynamic keystroke investigation, and key arrangement plans. A use of these models on electronic medical care records can be seen from [20], wherein an access control model is portrayed. This model can work on general productivity of access control and confirmation by means of hub level information checking, and segregation. The security of this model can be additionally further developed utilizing the work in [22][23][24][25] wherein different methods for confirmation and access control are examined. These methods incorporate three factor client controlled single sign on, unique exceptionally reconfigurable believed client gathering model by means of traits and meta graphs hypothesis, contingent protection saving confirmation model for side channel assault evacuation, and different other security designs. Numerical changes like Phase Truncated Fourier Transform (PTFT) can likewise be utilized for cancellable biometric confirmation as observed in [26], wherein precision of validation is exceptionally high. Utilization of these models for real time deployments can be observed from [27][28][29][30], wherein secure cloud models, limitless medical care administrations by means of block chain, secure gathering of correspondences, and cosmology information for access control are depicted. These applications use high security confirmation, and fast access control to further enhances cloud security. Comparative models for validation and access control that use secret sharing, and area-based verification with access control can likewise be observed from [31][32]. These models target further developing security by means of getting to and investigating client inputs utilizing various guidelines. Outline of different validation and access control systems can likewise be seen from [33], wherein investigation of these models dependent on attacks and attack tokens is depicted. Accordingly, it tends to be seen that not many methodologies have utilized transient investigation for access control, and the vast majority of them rely upon rule-based examination of client inputs. Spurred by this methodology, the proposed approach utilizes design investigation for moderation of cloud access control and confirmation weaknesses using temporal analysis, and is depicted in the following section

PATAC: Proposed pattern analysis model for securing Web Application against broken Authentication & Access Control.

Analysis of variations in user input plays a very important role during design of security patches for cloud deployments. From the review, it is observed that majority of researchers utilize these user input variations in order to design highly efficient attack mitigation systems. These systems are inherently 2-class classifiers that divide user input into malicious and normal categories. Majority of these engines are dependent upon instantaneous analysis of user requests, which limits their capability while detecting& mitigating authentication & access control attacks. This capability can be enhanced via use of temporal pattern analysis, and tracking, which would allow vulnerability to differentiate between normal and malignant user requests. To perform this task, this section proposes design of 3-layered security model, which includes,

- A header-level authentication layer, which utilizes rule engines for granting or blocking user entry into the system.
- An extended header-level access control layer, which uses temporal pattern analysis for evaluation& classification of user requests.
- A highly efficient logging engine, that assists in IP tracing & future attack mitigation.

All these layers are deployed on cloud header as observed from figure 2, which indicates that any GET, POST, PUT, or DELETE request passes through this layer before entering the cloud.

This is done so that all requests are pre-processed, validated, and checked before being actually processed by the cloud. The first level of checks is performed by authentication layer, which assists in validating the user's identity. The design of authentication layer is described in section 3.1 of this text, which is followed by design of the extended header-level access control layer, which uses temporal pattern analysis for evaluation & classification of user requests. These layers assist in marking the incoming requests as malicious or non-malicious. The non-malicious requests are passed to the cloud for processing, while their counterparts are given to a highly efficient logging engine that assists in IP tracing & future attack mitigation. Design of this high-efficiency logging engine is described in section 3.3 of this text, wherein various tracked entities, and their significance of tracking is mentioned in details.

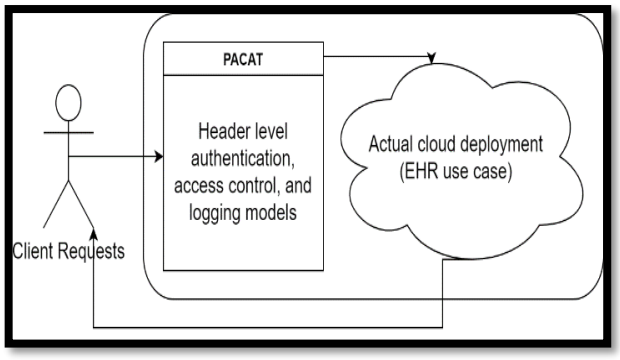


Fig 2. Data flow using the designed PATAC model

A header-level authentication layer, which utilizes rule engines for granting or blocking user entry into the system.

As observed from figure 2, each incoming request is given to a header level authentication layer. This layer assists in classification of incoming requests into authenticated and non-authenticated ones. The process of verification is described as follows, and utilizes pattern validation and regular expression-based matching for validation of the incoming requests,

- All incoming requests must either have a SESSION_ID, or an authentication data packet. This data packet can consist of username/password combination, phone number/OTP combination, OAuth token, etc. In this case, a username/password combination along with captcha is used.
- The captcha assists in reducing probability of brute force attacks.
- Once the login request is accepted, then a SESSION_ID is given to the client, which can be used for any future authorizations.
- To validate every incoming request, it is passed through the following checks,
  - Check if the request has any hypertext data, or hypertext tags. (<html>, <script>, etc.).

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages: 2232-2239

- Check if the user input has any single quotes (‘), which is one of the major reasons of SQL Injection attacks.
- These attacks can cause database to malfunction, and provide access tokens to attacking users.
- Check if the user input has (--) sign, which indicates comments for SQL, and can assist attackers to get authorized.
- Check if the user input has any of the following JavaScript keywords,
  - Wscript, javascript, jar, applescript, jscript, behavior, mocha, vbscript, vbs, livescript, or view-source.
- Check for presence of following regular expressions in the user input,
  - form, style, xlink:href, xmlns:xdp, seek segment Time, form action, or FSCommand
- Check if input encoding type is utf8mb4
- Check for presence of non-visible ASCII characters, which might inject cross-side-scripting (XSS) attacks. ASCII values between 00-08, 11, 12, 14, 15, and 16-31 are invisible characters.

- In case any of these checks are true, then the identified characters are replaced with a blank character (‘’), and request is reported to the logging engine. The following details are added to the report.
  - Type of attack: XSS or SQLi
  - Client’s IP address.
  - The timestamp at which request had arrived.
- The processed & validated data is passed to the cloud, where authentication is performed. If cloud authenticates the validated data, then a SESSION_ID is given to the client, which is used for future interactions with the cloud.
- After authentication, a session is started, and details including client ID, name, email, etc. are stored on the server in an encrypted format. This information can be accessed using an unique SESSION_ID, which is given to the client, and stored on their browser.
- If requested credentials are not matched by the cloud, then request is marked as an ‘Invalid Authentication’ request, and given to logging engine with the following parameters,
  - After Attack Type: Invalid Authentication
  - Client IP address
  - The timestamp at which request had arrived
  - All request parameters (username/password, phone number/OTP, etc.) used during login.

In order to access the cloud, client’s browser is required to pass the SESSION_ID with every consecutive request. This request is mapped & accompanied with access control mechanics, and processed via application-specific access-control engines. Design of the proposed access control engine is described in the next sub-section of this section.

An extended header-level access control layer, which uses temporal pattern analysis for evaluation & classification of user requests

After authentication, an extended header-level access control layer is activated. This layer uses temporal pattern analysis for evaluation & classification of user requests. Each user request is categorized into the following classes,

- Requests sent by admin, which have the capability of accessing all pages, and performing all operations.
- Requests sent by non-admin users, which are further divided into the following sub-categories,
  - Insert requests, to add data to the cloud
  - Modify requests, to update data on the cloud
  - Delete requests, to remove data from the cloud
  - Search requests, to view stored data

Based on these categories, the following access control mechanisms are activated, and temporal patterns are analyzed,

- All incoming client requests, after authentication, are captured by this header analysis layer, and GET, POST; PUT & DELETE variables are tracked.
- These variables are classified based on cloud action, and stored in an internal hash-table. Layout of this hash-table can be observed from table 1, wherein SESSION_ID of each user is stored along with suggested cloud action.

TABLE I. MAPPING TABLE FOR VARIABLE TO REQUESTED ACTIONS

Sess. ID	Type of request	Var. Name	Name of Page	Cloud Action
1234	*GET*	*selUser*	*User Record*	*Read*

- Check the passed SESSION_ID, and compare it with the ID on server.
- If, SESSION_ID∈Server$_{SESSION_IDS}$, then checking of access control is done, otherwise user is logged out of the system, its SESSION_ID is nullified, and this incident report is given to the logging engine with following details,
  - Attack Type: Invalid Authentication
  - Client IP address
  - The timestamp at which request had arrived
  - All passed request variables with their corresponding values
- If the request is authenticated, then another hash table is generated. This table consists of all the incoming requests, and their validated values as observed from table 2,

TABLE 2. GENERATED HASH TABLE FOR EACH INCOMING ACCESS CONTROL REQUEST

Sess. ID	Req. Type	Var. Name	Page Name	Valid Value
1234	*GET*	*selUser*	*User Record*	*5*

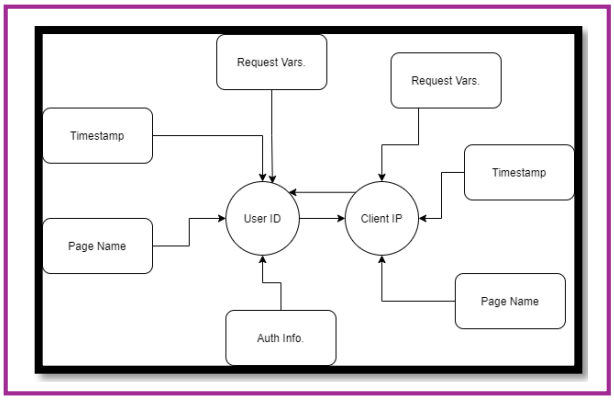
  - Name of the accessed page
- In case of a match, a rule engine is activated, and the following conditions are checked for access control,
- Both the hash tables 1, and 2 are matched on the basis of SESSION_ID, request type, variable name, and page name.
- In case of non-matching requests, the user is logged out, its SESSION_ID is nullified, and the following details are given to the logging engine-
  - Attack Type: Invalid Access
  - Client IP address
  - The timestamp at which request had arrived
  - All passed request variables with their corresponding values

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages: 2232-2239

Table 3. Conditions for access control

Request	Given Access	Request Passed
Page Write	Page Write	Yes
Page Read	Page Read	Yes
Page View	Page View	Yes
Page Write	Page Read and/or Page View	No
Page Read	Page Write and/or Page View	No
Page View	Page write and/or Page read	No

- If the conditions granted to the user are verified, then the requested page is displayed to the user, else the request is rejected, and logged via the logging engine.
- Thus, if any of the NOT granted conditions are satisfied, then invalid access incidence is reported using the following parameters,
  - Attack Type: Invalid Access
  - Client IP address
  - The timestamp at which request had arrived
  - All passed request variables with their corresponding values
  - Name of the accessed page
- For temporal analysis, each request variable to checked, and its corresponding response is evaluated. This evaluation is logged, and a temporal validity percentage is evaluated using equation (1) as follows,



$$P_{valid} = \left(\frac{\sum_{T=t_1}^{t_2} P_{invalid_T}}{\sum_{T=t_1}^{t_2} P_{valid_T}}\right) \quad (1)$$

Where, P_{valid} is temporal validity percentage for the given time instances t_1 & t_2, while, P_{validT} and $P_{invalid\_T}$ represents number of valid & invalid requests identified during the underlying temporal interval.

- In case the value of $P_{valid} > 0.1$, then it indicates that over 10% of all requests are invalid. This means the requesting user is trying to continuously access the system with improper credentials. In such a condition, the user ID is blocked, and following information is stored via logging engine,
  - Attack Type: Invalid Temporal Access
  - Client IP address
  - The timestamp at which request had arrived
  - All passed request variables with their corresponding values
  - Name of the accessed page

This interval is modified as per user requests, and checked continuously. User accounts which show malicious activities are temporarily blocked. But as the number of malicious activities increases, these accounts are blocked permanently. Tracking of these malicious activities is facilitated using a highly efficient logging engine that assists in IP tracing & future attack mitigation. Use of this engine is already mentioned multiple times in current and previous sub-section. The design of this engine is described in the next sub-section, wherein data storage & tracking operations are mentioned in details.

A highly efficient logging engine, that assists in IP tracing & future attack mitigation

Due to inconsistency of user inputs, a relational database is insufficient for logging user requests. Thus, the proposed high efficiency logging engine uses graph representations to store all the logs. The data captured by this engine is described in table 4, wherein various tracking parameters are mentioned. These parameters are represented in the form of a graph (as observed from figure 3) for efficient visualization. Using this graphical visualization, cyber experts can understand which users are sending continuous malicious requests, and thus assist in blocking them out of the system.

Moreover, this graph engine is useful for a wide variety of data analytic operations, including design of cyber recommender systems. This comprehensive data storage format assists in warehousing & mining data related to any kind of network events & attack. These events are represented using figure 3. The User ID and client IP are central points of identification for each request. All data can be linked with these 2 entities, and can be used for future tracings. For instance, if a user is accessing a given page, then it's UserID & timestamp information can be used for evaluation of access time.

This information can be extended via mining the request variables, which allows for checking request type, and page access type. For instance, if a user is currently performing record updation, and requires large delay for this operation, then it can be marked as an outlier event, and can be later analyzed for malicious activities. Such operations allow for future risk mitigations, and assist in tracking the given user based on their ID and IP information. In most of the cases, where user is continuously trying to access the system with malicious intent, its ID and IP are blocked across entire cloud deployment. This assists in improving accuracy of access control & authorization attack detection, accuracy of access control check, and reducing delay of authorization & access control checks. Analysis of these parameters, along with their comparison with different state-of-the-art models is described in the next section of this text.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages: 2232-2239

TABLE 4. DATA STORAGE DESIGN FOR THE HIGH EFFICIENCY LOGGING ENGINE

UID	Client IP	Type of attack	Timestamp	Service Name
Req. Vars.	Vals. Of req. vars.	Auth. Data	Temporal service access info.	Other Meta data

IV. Results analysis and comparison

An electronic healthcare record (EHR) management application was designed to test the given authentication & access control check engine. Design of this EHR management application is described using the following steps,

- Users are required to provide a username, password, and captcha combination during login. The captcha allows for removal of dictionary attacks.
- Users are able to upload, delete, modify, and view their own healthcare records.
- Users (patients) are able to view different doctors, and grant them access to view these records.
- Doctors are able to view records, and use them for diagnostic purposes.
- Admin are able to perform the following operations,
    - Add, update, delete and view patients
    - Add, update, delete and view doctors
    - Add, update, delete and view access control permissions to users

Based on this EHR design, the following scenarios were simulated, and performance metrics were evaluated,

- Simulate over 10 million authentication requests, and verify authentication performance.
- For each of these requests, perform access control using the following,
    - Try to access pages which are not granted access by the admin.
    - Try to upload EHR records for other patients.
    - Try to view EHR records for other patients.

Total 10 million authentications & access control requests were sent to the system, and security performance was measured. This performance is measured using the following metrics,

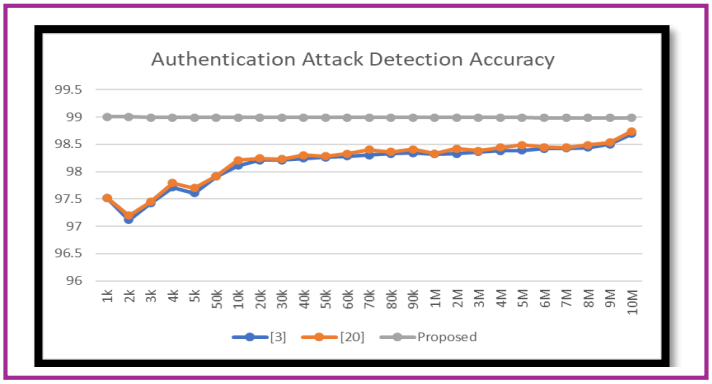
- Accuracy for detection of authentication attack (AAuth)
- Delay needed for checking authentication attack check (DAuth)
- Delay needed for access control check (DAcc)
- Accuracy of access control check (AAcc)
- Accuracy of user input-based attacks (AUser)

Each of these parameters along with their respective values were tabulated, and visualized for different access control & authorization models. For instance, accuracy of authorization attack detection (AAuth) can be observed from table 4 as follows,

Using the results in table 5 and its visualization in figure 4, it can be observed that the proposed model has comparable performance for authorization attack detection & removal when compared with [3], and [20].

TABLE 5. ACCURACY OF AUTHORIZATION ATTACK DETECTION FOR DIFFERENT MODELS

Num. Requests	AAuth(%)[3]	AAuth (%)[20]	AAuth Proposed
1k	97.52	97.52	99.00
2k	97.12	97.20	99.00
3k	97.42	97.45	98.99
4k	97.71	97.79	98.99
5k	97.61	97.70	98.99
50k	97.91	97.92	98.99
10k	98.11	98.21	98.99
20k	98.21	98.24	98.99
30k	98.21	98.23	98.99
40k	98.24	98.30	98.99
50k	98.26	98.28	98.99
60k	98.28	98.33	98.99
70k	98.30	98.40	98.99
80k	98.33	98.36	98.99
90k	98.34	98.41	98.99
1M	98.32	98.33	98.99
2M	98.33	98.42	98.99
3M	98.36	98.38	98.99
4M	98.38	98.44	98.99
5M	98.39	98.49	98.99
6M	98.42	98.45	98.98
7M	98.43	98.44	98.98
8M	98.44	98.49	98.98
9M	98.50	98.53	98.98
10M	98.69	98.74	98.98



This showcases that the proposed model is deployment ready, and can be used for Web based applications. Similar observations were made for delay of authorization check (DAuth), and can be observed from table 6 as follows Using the analysis in table 6, it can be observed that the proposed PACAT modelhas63% lower delay when compared with models proposed in [3], and [20]. This is due to the light weight sanitization and session hashed based authorization process followed by the model, thereby showcasing its superior performance. Similar

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages: 2232-2239

observations were made for accuracy of access control check (AAcc), and can be observed from table 7 as follows,
Based on this analysis and visualization in figure 5, it can be observed that the proposed model has 46% better access control detection accuracy when compared with models proposed in [3], and [20].

TABLE 6. DELAY OF AUTHORIZATION ATTACK DETECTION FOR DIFFERENT MODELS

Num. Requests	DAuth(ms) [3]	DAuth(ms) [20]	DAuth(ms) Proposed
1k	10.05	10.05	1.98
2k	50.46	50.42	9.90
3k	100.61	100.58	19.80
4k	200.61	200.45	39.60
5k	245.99	245.77	48.51
50k	307.31	307.28	60.79
10k	368.63	368.26	73.07
20k	430.13	430.00	85.35
30k	492.01	491.91	97.62
40k	553.71	553.38	109.90
50k	615.45	615.32	122.18
60k	677.15	676.81	134.46
70k	738.84	738.09	146.73
80k	800.41	800.17	159.01
90k	862.13	861.52	171.29
1M	924.11	924.01	183.56
2M	985.81	984.92	195.84
3M	1047.30	1047.09	208.12
4M	1108.86	1108.19	220.40
5M	1170.51	1169.33	232.67
6M	1231.90	1231.53	244.98
7M	1293.51	1293.38	257.25
8M	1355.11	1354.43	269.53
9M	1415.99	1415.42	281.81
10M	1430.15	1429.58	284.63

TABLE 7. ACCURACY OF ACCESS CONTROL DETECTION FOR DIFFERENT MODELS

Num. Requests	AAcc(%)[3]	AAcc (%)[20]	AAcc (%)Proposed
1k	52.72	54.71	97.46
2k	57.47	49.59	97.46
3k	57.62	51.70	97.74
4k	52.82	49.89	97.74
5k	50.79	55.79	97.54
50k	50.94	52.93	97.45
10k	50.04	58.01	97.64
20k	52.07	59.02	97.84
30k	57.02	58.02	97.64
40k	59.02	56.08	97.45
50k	59.03	52.11	97.64
60k	52.11	57.08	97.54
70k	50.14	51.18	97.64
80k	56.09	55.12	97.45
90k	51.15	59.10	97.54
1M	52.13	59.06	97.84
2M	54.11	55.15	97.54
3M	51.16	53.15	97.74
4M	55.13	58.13	97.15
5M	53.15	52.21	97.25
6M	53.17	57.14	97.24
7M	58.12	50.21	97.93
8M	53.18	58.15	97.93
9M	51.23	52.24	97.24
10M	51.74	52.76	98.21

There by s howcasing its superior performance under a large number of requests. Similar observations were made for delay of access control check (DAcc), and can be observed from table 8 as follows.Based on this analysis and the visualization in figure 6, it can be observed that the proposed model showcases 55% reduction in delay when compared with models proposed in [3], and [20]. This is due to the light-weight pattern analysis for access control check process followed by the model, thereby showcasing its superior performance. Similar observations were made for accuracy of user attack (AUser), which includes SQL injection, XSS, access control, DoS attacks& cookie hijacking, and can be observed from table 9 as follows,
Based on this analysis, it can be observed that the proposed model has similar performance for user attack detection accuracy when compared with models proposed in [3], and [20], thereby making it deployable for real time SaaS based cloud environments

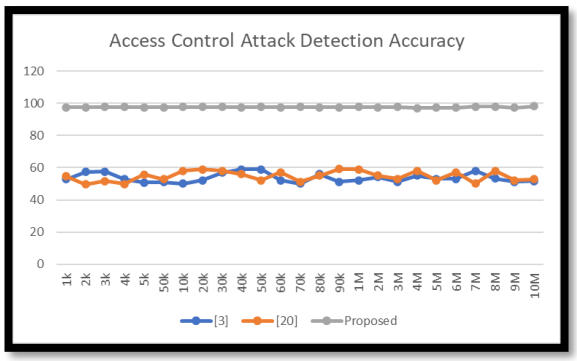


Fig 5. Access control accuracy performance

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages: 2232-2239

.V. Conclusion and future scope

Design of the proposed 3-layered model for access control & broken authentication vulnerabilities showcases good performance in terms of accuracy of access control attack detection, while its performance for authentication & user-initiated attacks is at par with standard models proposed in [3], and [20]. This performance improvement is mainly due to the temporal analysis engine, which assists in blocking users based on their temporal request patterns, and logging their uniquely identifiable information for future tracking. The proposed model also outperforms existing methods in terms of delay of authentication checks, and delay of access control checks, which is due to light-weight rule-based design. This reduction in delay assists the model to be applied for real time, high speed, and high throughput SaaS cloud deployments that demand high security. Moreover, the model also proposes design of a high efficiency logging system, which utilizes graph databases for improving temporal mining performance. The proposed logging engine can be combined with data analytic tools in order to visualize user access & authentication patterns, thereby assisting system admin. This work can be extended via use of block chain-based models for better tracing, and transparency. The system can also be extended to detect ownership transfer attacks, which can be handled using a combination of block chain & deep learning methods. Visualization capability of proposed model must be enhanced via testing the logging engine on various data analytics & visualization tools.

TABLE 8. DELAY OF ACCESS CONTROL CHECK DETECTION FOR DIFFERENT MODELS

Num. Requests	DAcc(ms) [3]	DAcc (ms) [20]	DAcc (ms) Proposed
1k	5.36	5.65	1.95
2k	29.80	25.25	9.77
3k	53.53	51.51	19.57
4k	115.05	105.01	38.99
5k	138.49	123.73	47.66
50k	158.11	170.40	60.20
10k	186.34	190.02	72.21
20k	234.85	226.24	84.44
30k	248.96	278.47	96.29
40k	285.80	330.07	108.07
50k	323.88	366.94	120.14
60k	403.82	342.88	132.48
70k	396.35	440.60	145.16
80k	469.52	429.49	156.67
90k	488.53	454.02	169.45
1M	468.10	468.10	180.86
2M	538.85	499.41	192.76
3M	562.14	530.72	205.47
4M	584.20	628.51	218.04
5M	686.97	616.70	228.32
6M	624.63	661.58	240.61
7M	746.49	759.41	253.45
8M	755.00	700.78	266.35

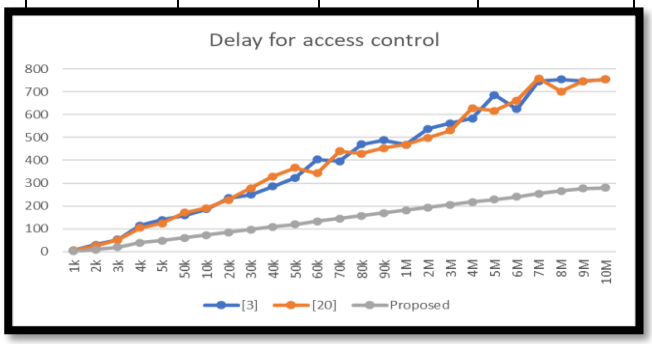


Fig 6. Delay for access control detection for various models

TABLE 9. ACCURACY OF USER ATTACK DETECTION FOR DIFFERENT MODELS

Num. Requests	AUser(%)[3]	AUser (%)[20]	AUser (%)Proposed
1k	98.43	97.94	98.19
2k	98.43	97.94	98.19
3k	98.72	98.23	98.48
4k	98.72	98.23	98.48
5k	98.52	98.03	98.28
50k	98.42	97.93	98.18
10k	98.62	98.13	98.38
20k	98.82	98.33	98.57
30k	98.62	98.13	98.38
40k	98.42	97.93	98.18
50k	98.62	98.13	98.38
60k	98.52	98.03	98.28
70k	98.62	98.13	98.38
80k	98.42	97.93	98.18
90k	98.52	98.03	98.28
1M	98.82	98.33	98.57
2M	98.52	98.03	98.28
3M	98.72	98.23	98.48
4M	98.12	97.63	97.88
5M	98.22	97.73	97.98
6M	98.21	97.72	97.97
7M	98.91	98.42	98.66
8M	98.91	98.42	98.66
9M	98.21	97.72	97.97
10M	99.19	98.70	98.95

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages: 2232-2239

References

[1] T. Cai, Z. Yang, W. Chen, Z. Zheng and Y. Yu, "A Blockchain-Assisted Trust Access Authentication System for Solid," in IEEE Access, vol. 8, pp. 71605-71616, 2020, doi: 10.1109/ACCESS.2020.2987608.
[2] M. Alhaidary et al., "Vulnerability Analysis for the Authentication Protocols in Trusted Computing Platforms and a Proposed Enhancement of the OffPAD Protocol," in IEEE Access, vol. 6, pp. 6071-6081, 2018, doi: 10.1109/ACCESS.2017.2789301.
[3] Z. Mehmood, A. Ghani, G. Chen and A. S. Alghamdi, "Authentication and Secure Key Management in E-Health Services: A Robust and Efficient Protocol Using Biometrics," in IEEE Access, vol. 7, pp. 113385-113397, 2019, doi: 10.1109/ACCESS.2019.2935313.
[4] B. Maciej, E. F. Imed and M. Kurkowski, "Multifactor Authentication Protocol in a Mobile Environment," in IEEE Access, vol. 7, pp. 157185-157199, 2019, doi: 10.1109/ACCESS.2019.2948922.
[5] X. Zhang, D. Cheng, P. Jia, Y. Dai and X. Xu, "An Efficient Android-Based Multimodal Biometric Authentication System With Face and Voice," in IEEE Access, vol. 8, pp. 102757-102772, 2020, doi: 10.1109/ACCESS.2020.2999115.
[6] A. A. S. AlQahtani, H. Alamleh and J. Gourd, "0EISUA: Zero Effort Indoor Secure User Authentication," in IEEE Access, vol. 8, pp. 79069-79078, 2020, doi: 10.1109/ACCESS.2020.2990604.
[7] H. Wang, T. Chen, X. Liu and J. Chen, "Exploring the Hand and Finger-Issued Behaviors Toward Natural Authentication," in IEEE Access, vol. 8, pp. 55815-55825, 2020, doi: 10.1109/ACCESS.2020.2981828.
[8] M. Adil et al., "MAC-AODV Based Mutual Authentication Scheme for Constraint Oriented Networks," in IEEE Access, vol. 8, pp. 44459-44469, 2020, doi: 10.1109/ACCESS.2020.2978303.
[9] S. Ding, J. Cao, C. Li, K. Fan and H. Li, "A Novel Attribute-Based Access Control Scheme Using Blockchain for IoT," in IEEE Access, vol. 7, pp. 38431-38441, 2019, doi: 10.1109/ACCESS.2019.2905846.
[10] B. S. Saini et al., "A Three-Step Authentication Model for Mobile Phone User Using Keystroke Dynamics," in IEEE Access, vol. 8, pp. 125909-125922, 2020, doi: 10.1109/ACCESS.2020.3008019.
[11] K. Riad, R. Hamza and H. Yan, "Sensitive and Energetic IoT Access Control for Managing Cloud Electronic Health Records," in IEEE Access, vol. 7, pp. 86384-86393, 2019, doi: 10.1109/ACCESS.2019.2926354.
[12] Y. Hou, S. Garg, L. Hui, D. N. K. Jayakody, R. Jin and M. S. Hossain, "A Data Security Enhanced Access Control Mechanism in Mobile Edge Computing," in IEEE Access, vol. 8, pp. 136119-136130, 2020, doi: 10.1109/ACCESS.2020.3011477.
[13] C. -S. Park and H. -M. Nam, "Security Architecture and Protocols for Secure MQTT-SN," in IEEE Access, vol. 8, pp. 226422-226436, 2020, doi: 10.1109/ACCESS.2020.3045441.
[14] M. A. Al Sibahee et al., "Lightweight Secure Message Delivery for E2E S2S Communication in the IoT-Cloud System," in IEEE Access, vol. 8, pp. 218331-218347, 2020, doi: 10.1109/ACCESS.2020.3041809.
[15] Kumar, V, Kumar, R, Pandey, SK. A secure and robust group key distribution and authentication protocol with efficient rekey mechanism for dynamic access control in secure group communications. Int J Commun Syst. 2020; 33:e4465. https://doi.org/10.1002/dac.4465
[16] Tiwari, D, Chaturvedi, GK, Gangadharan, GR. ACDAS: Authenticated controlled data access and sharing scheme for cloud storage. Int J Commun Syst. 2019; 32:e4072. https://doi.org/10.1002/dac.4072
[17] Boonkrong, Sirapat. (2021). Authentication and Access Control: Practical Cryptography Methods and Tools. 10.1007/978-1-4842-6570-3.
[18] Z. Ai, Y. Liu, L. Chang, F. Lin and F. Song, "A Smart Collaborative Authentication Framework for Multi-Dimensional Fine-Grained Control," in IEEE Access, vol. 8, pp. 8101-8113, 2020, doi: 10.1109/ACCESS.2019.2962247.
[19] A. K. Das, M. Wazid, A. R. Yannam, J. J. P. C. Rodrigues and Y. Park, "Provably Secure ECC-Based Device Access Control and Key Agreement Protocol for IoT Environment," in IEEE Access, vol. 7, pp. 55382-55397, 2019, doi: 10.1109/ACCESS.2019.2912998.
[20] T. Gao, X. Deng, Y. Wang and X. Kong, "PAAS: PMIPv6 Access Authentication Scheme Based on Identity-Based Signature in VANETs," in IEEE Access, vol. 6, pp. 37480-37492, 2018, doi: 10.1109/ACCESS.2018.2841008.
[21] S. A. Chaudhry, K. Yahya, F. Al-Turjman and M. -H. Yang, "A Secure and Reliable Device Access Control Scheme for IoT Based Sensor Cloud Systems," in IEEE Access, vol. 8, pp. 139244-139254, 2020, doi: 10.1109/ACCESS.2020.3012121.
[22] J. Zhao et al., "A Secure Biometrics and PUFs-Based Authentication Scheme With Key Agreement For Multi-Server Environments," in IEEE Access, vol. 8, pp. 45292-45303, 2020, doi: 10.1109/ACCESS.2020.2975615.
[23] M. Seo, J. Y. Hwang, D. H. Lee, S. Kim, S. Kim and J. H. Park, "Fuzzy Vector Signature and Its Application to Privacy-Preserving Authentication," in IEEE Access, vol. 7, pp. 69892-69906, 2019, doi: 10.1109/ACCESS.2019.2919351.
[24] J. K. Mudhar, S. Kalra and J. Malhotra, "An Efficient Blockchain Based Authentication Scheme to Secure Fog Enabled IoT Devices," 2020 Indo – Taiwan 2nd International Conference on Computing, Analytics and Networks (Indo-Taiwan ICAN), 2020, pp. 75-80, doi: 10.1109/Indo-TaiwanICAN48429.2020.9181356.
[25] Y. Tang, T. Liu, X. He, J. Yu and P. Qin, "A Lightweight Two-Way Authentication Scheme Between Communication Nodes for Software Defined Optical Access Network," in IEEE Access, vol. 7, pp. 133248-133256, 2019, doi: 10.1109/ACCESS.2019.2941084.
[26] A. A. Jasim et al., "Secure and Energy-Efficient Data Aggregation Method Based on an Access Control Model," in IEEE Access, vol. 7, pp. 164327-164343, 2019, doi: 10.1109/ACCESS.2019.2952904.
[27] Q. Lyu, N. Zheng, H. Liu, C. Gao, S. Chen and J. Liu, "Remotely Access "My" Smart Home in Private: An Anti-Tracking Authentication and Key Agreement Scheme," in IEEE Access, vol. 7, pp. 41835-41851, 2019, doi: 10.1109/ACCESS.2019.2907602.
[28] C. -L. Hsu, T. -V. Le, M. -C. Hsieh, K. -Y. Tsai, C. -F. Lu and T. -W. Lin, "Three-Factor UCSSO Scheme With Fast Authentication and Privacy Protection for Telecare Medicine Information Systems," in IEEE Access, vol. 8, pp. 196553-196566, 2020, doi: 10.1109/ACCESS.2020.3035076.
[29] J. S. Alshudukhi, B. A. Mohammed and Z. G. Al-Mekhlafi, "An Efficient Conditional Privacy-Preserving Authentication Scheme for the Prevention of Side-Channel Attacks in Vehicular Ad Hoc Networks," in IEEE Access, vol. 8, pp. 226624-226636, 2020, doi: 10.1109/ACCESS.2020.3045940.
[30] A. Alarifi, M. Amoon, M. H. Aly and W. El-Shafai, "Optical PTFT Asymmetric Cryptosystem-Based Secure and Efficient Cancelable Biometric Recognition System," in IEEE Access, vol. 8, pp. 221246-221268, 2020, doi: 10.1109/ACCESS.2020.3043689.
[31] J. Indumathi et al., "Block Chain Based Internet of Medical Things for Uninterrupted, Ubiquitous, User-Friendly, Unflappable, Unblemished, Unlimited Health Care Services (BC IoMT U6 HCS)," in IEEE Access, vol. 8, pp. 216856-216872, 2020, doi: 10.1109/ACCESS.2020.3040240.
[32] Kiran, GM, Nalini, N. Enhanced security-aware technique and ontology data access control in cloud computing. Int J Commun Syst. 2020; 33:e4554. https://doi.org/10.1002/dac.4554
[33] Yuan, H, Maple, C, Lu, Y, Watson, T. Peer-assisted location authentication and access control for wireless networks. Internet Technology Letters 2019; 2:e71. https://doi.org/10.1002/itl2.71