Authentication and request processing model in high load modes for IoT components

Sergii Surkov [a], Oleksandr Martynyuk [a]

[a] *Odessa National Polytechnic University, Avenue Shevchenko, 1, Odessa, 65044, Ukraine*

Abstract

The paper is dedicated to the development and study of models and methods of authorization processes and high load modes for IoT components in the operation queue environment.

An essential part of authorization is the authentication of the transmitted data. In the trusted nodes, there's a risk of an unauthorized modification of payloads. It poses a threat because they may be infected with various viruses or botnets. Such nodes include trusted networks, load balancers, and proxy servers, where the data is transmitted in an unencrypted form. In the worst case, the attacker may even steal the private key of the TLS certificate. It enables him not only to analyze all the traffic but also to modify the payload. Situations of data spoofing at vulnerable points are poorly researched. A new "chunking" method was previously proposed and compared to the existing ones. Analytical studies have shown an increase in the performance of the new "chunking" method due to the more efficient usage of resources.

For efficient resource management in the multithreaded environment, it is necessary to load all the processor cores equally. A decent solution to take advantage of the multithreading environment is the operation queues. However, if an operation queue can't process accumulated operations efficiently, the system goes into high load mode. Such behavior may cause the system to run out of memory. The patterns of occurrence of high load modes have been researched previously. To prevent the high load modes, the load balancing of the server cluster method has been further developed, which can also be applied to a single server.

The new model of authentication and request processing in high load modes for IoT components has been created. It differs from the existing ones in the ability to prevent high load modes during the authentication of large payloads. It uses the limitation in the total bandwidth as well as the limitation of the number of requests. The information system for the IoT components, which implements the new model has been developed. In the test, several experiments were conducted, which show: high load mode, the effect of limitation of the bandwidth, effect of the limitation of the number of requests.

Keywords

authorization protocol, payload authentication, digital signature, information technology, server cluster, internet of things, smart car.

1. Introduction

The modern concept of web systems security focuses on systems without mandatory registration of users, where the main focus is to determine whether a user is a real person or not. In that area machine learning [1, 2] is the main focus of the study.

After user authorization, the main security aspect is the authentication of requests from the user to the server. In this paper we use term authentication as an act of validating that requests are what they

IntelITSIS'2021: 2nd International Workshop on Intelligent Information Technologies and Systems of Information Security, March 24–26, 2021, Khmelnytskyi, Ukraine

EMAIL: k1x0r@ukr.net (S. Surkov); anmartynyuk@ukr.net (O. Martynyuk);

ORCID: 0000-0001-9224-7526 (S. Surkov); 0000-0003-1461-2000 (O. Martynyuk);

claim to be. And the term authorization as the general term of authorization protocol which consists of authenticating user, authorizing or granting the permissions to the user, and authenticating requests.

With the increasing payload sizes, it's important to research authorization models and methods to improve the efficiency of payload authentication. For the authentication of large payloads, there are no protocols that verify it efficiently. To reduce resource consumption for authenticating payloads in the multithreaded environment, operation queues are used, which allow the developer to reuse previously created threads.

The most used protocol family of data transmission is HTTP[3, 4]. It's used for web browsers, desktop applications, mobile clients, and IoT devices. Data transmission can occur through many network nodes [5], which are trusted, but in some nodes, certificates can be substituted or data is transmitted in unencrypted form. This makes it necessary to authenticate request payloads for video streaming, document storage, database services with complex data center infrastructure, and more. The generally accepted protocol of encryption of the data during transmission is TLS[6, 7].

Proxy resources are controlled by the companies in which the people work, or by data centers, and generally they are considered to be trusted.

In the process of data verification in the operation queue environment, the high load modes may arise, leading to unstable operation[8, 9] of the information system, and these issues are poorly researched. This can lead to an increase in memory consumption and therefore to unstable operation and termination of the server process. The behavior of high load modes for authorization protocols of large payloads is poorly researched.

The formal analysis for existing authorization processes in information systems has been carried out. The existing methods and models for ensuring the security of authorization processes in information systems have been researched. It was found that the existing systems require a lot of resources for authentication of large payloads.

In a multithreaded environment, it is important to distribute the workload across the processor cores to efficiently manage resources. Operation queues [10, 11] are an efficient solution in a multithreaded environment, which helps to distribute the workload among all the processor cores. On the other hand, data changes during transmission can occur if no encryption is used or the data is decrypted in transit.

A high load mode occurs when operations in the queue accumulate and do not have time to be processed. It increases memory consumption, which can cause system instability. The means of detecting and preventing such modes does not allow to get rid of them completely. The study of these modes makes it possible to predict their impact on equipment and thereby increase the reliability[12, 13] of the system and reduce their negative effect.

The behavior of the high load modes themselves for authorization protocols for large requests is poorly understood, and their study is relevant.

The **purpose** of this work is to improve performance and reduce the harmful effect of high load modes in the operation queue environment.

To achieve this goal, the following tasks were completed:

- analyzed existing models, methods and information technologies, areas of application, advantages, disadvantages of methods for request authentication
- analyzed high load modes for authorization protocols in the operation queue environment
- developed a model and method of the authentication of large payloads for HTTP and HTTP/2 protocols for blocking and non-blocking sockets
- developed methods for ranking authentication payload mechanisms
- developed methods for identifying and studying the influence of high load modes on the system memory in the operation queue environment
- improved method of migration from a system with a single server to the server cluster
- authentication and request processing model in high load modes for IoT components has been further developed
- the method of building communication mobile systems for embedded car computers that do not have their own web browsers has been further developed

2. Related Works

To guarantee the immutability of the data during transmission, it's crucial to prevent MITM [14, 15] (Man in the Middle) attacks. The authentication of only the request headers by most authorization protocols leaves the payload prone to modification. Such attacks are becoming more relevant because of the increasing number of infected computers and botnets[16].

Protocols like OAuth 1.0a[17] and HAWK[18] use the HMAC method [19] to prevent the modification of payloads. These protocols work great for the authentication of small payloads.

It's essential for authorization protocols to authenticate the payload, although some rely on TLS encryption[20]. The generally used method is "filling the buffer" with its subtypes "buffering to memory" and "buffering to file" for large and small data sizes. Analysis of existing methods for verifying data integrity showed the necessity for scientific research - modeling existing authorization protocols for authentication of the payload, creating a new model and method for authentication of large payloads to improve the system performance.

A model and method of chunking authentication of large payloads have been developed, which allow to achieve an increase in performance by reducing the number of operations. They differ from existing ones by working directly with data chunks, and authenticating them as they appear.

2.1. The model of authentication of payloads by chunks

The model of authentication of payloads by chunks was developed [21] it differs from the existing ones in the representation of the payload as the data stream in the form of chunks. The digital signature is located in the end, which makes it possible to verify the data during transmission and removes the necessity to read it again. The size of the chunk and the metadata block must be a multiple of the block size of the hashing algorithm, and the remainder of the last chunk is filled with zeros.

An example of a chunking payload authentication model is shown in Figure 1.

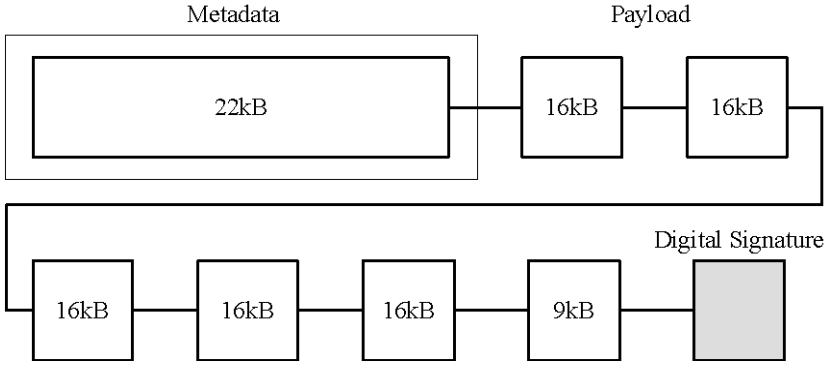

Figure 1: An example of a chunking payload authentication model

On the basis of the model, a new method of chunking payload authentication for large payloads has been developed, which makes it possible to achieve an increase the performance by reducing the number of operations. The new method has been implemented for HTTP authorization protocols, which use HMAC for payload verification. This allows web services to authenticate large payloads and not be exposed to denial-of-service attacks[22].

2.2. The method of authentication of payloads by chunks

In the "chunking" method, not all of the request data is buffered, but a small portion, which is used to update digital signature of the request. This has the advantage of reducing CPU utilization for both blocking and non-blocking sockets.

The "chunking" method gives an advantage in reduced CPU usage with both blocking and non-blocking sockets. It has the best effect in server applications with non-blocking sockets. Despite the

blocking sockets are less efficient than non-blocking ones, the method still gives a huge advantage for authentication of large payloads. And the blocking sockets might be feasible to use it in client applications with existing libraries.

Using our method, it's possible to create or modify an HMAC-based HTTP authorization protocol to enable support of large payload authentication. As a result, the "chunking" method provides an ability for the servers to efficiently authenticate large payloads, which makes them less prone to denial of service attacks.

The method of chunking payload authentication is shown in Figure 2.

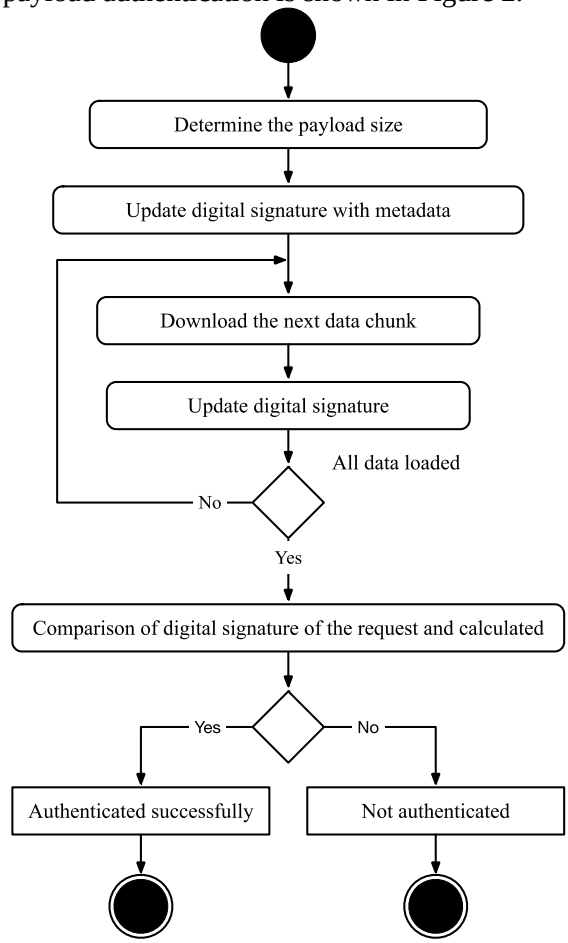

Figure 2: The method of authentication of payloads by chunks

After the chunk is received, the digital signature is updated and processed further. The incoming chunk size for the "Update digital signature" unit is irrelevant. It is possible because the mechanism of this unit structures the incoming data stream according to the hashing algorithm block size.

The data processing flow for the "chunking" method is shown in Figure 3.

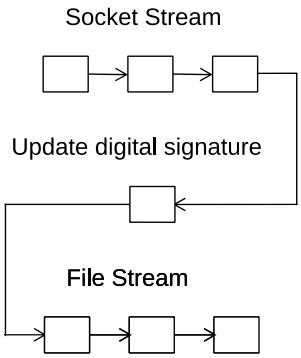

Figure 3: The data processing sequence for the "chunking" method

Because there's no necessity to process the "chunks" again after they were received, no intermediate storage is required. Therefore, the "chunking" method is limited only by the computational power of the CPU.

The handling of the data chunks is performed by the operation queues in the multithreaded environment[23, 24]. The most common use case of transmitting large payloads to the server is uploading a file to persistent storage. Worth noting, that the disk-related operations are usually distributed on a single logical disk. This is usually the weakest point in the system.

The model and method of authentication of payloads by chunks are used in the communication model of IoT components.

2.3. Communication model of IoT components

The communication model of IoT[25] components was further developed, combining the "request-response" and "publisher-subscriber" behavioral models, which differs from the existing ones in the ability to work and verify large payloads, which allows to achieve an increase in the speed and reliability[26] of the information system when working with large payloads.

The communication model consists of three components: client, server and module. A central server functions as a mediator between IoT modules and clients. In the case of a single client, the only advantage of the new model over the request-response model may be the amount of data sent and received. The advantage of the new model will be if multiple clients are connected to the central server. The new model allows transferring and verifying a large amount of data using them on low-power devices using the "chunking" method.

An example of IoT communication model is shown in Figure 4.

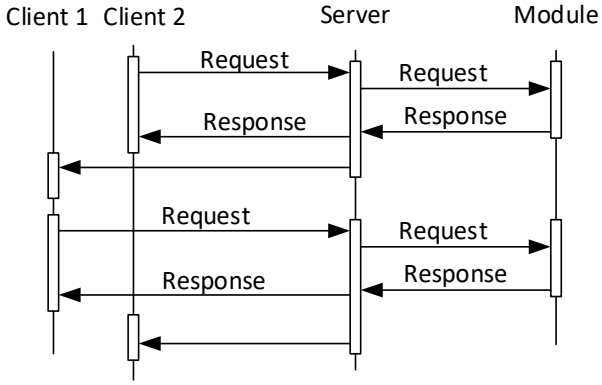

Figure 4: An example of an IoT communication model

The use of the "request-response" model makes it possible to perform individual requests. Taking advantage of the speed of receiving updates to the state of the publisher-subscriber model allows the clients to receive updates from the modules on the fly.

2.4. The influence of high load modes on data authorization mechanisms

The analysis showed the advantage of the "chunking" method. To validate the analysis, it is necessary to compare the researched methods. To do this, the method of comparison of data authentication mechanisms was developed. For the input of the new method, "Buffering to File", "Buffering to Memory" and "Chunking" data authentication mechanisms were created. The results have confirmed the analysis and proved the advantage of the "chunking" method by 13-22% for a large payloads and 1 percent for small payloads[27].

Based on the method of comparison of data authentication mechanisms, there were developed methods for identifying high load modes and studying the influence of high load modes on data authorization mechanisms in the operation queue environment. The results of this study can be used to find the optimal load for the equipment.

The original method differs from the existing ones in balancing according to the capabilities of the servers, which allows to achieve an increase in the performance and reliability[28, 29] of the server cluster by selecting the optimal load of the equipment in the cluster and the possibility of self-recovering. From the server side of the cluster, to confirm that the load balancer is available, it must periodically send heartbeat to the load balancer. Research has shown that these measures will allow self-recovering in emergency situations.

3. Proposed model and method

As a conclusion to the paper[27], a suggestion was made to counteract high load modes based on an active number of connections. Having a low number of maximum concurrent requests limit leaves a chance that a server might not be 100% loaded. Moreover, a denial-of-service attack might happen with a technique of starting a number of concurrent connections to the server with the minimal possible speed. To counteract high load modes, a method of migration from a single server to the server cluster has received further development. The main difference between the original method[32] and the modified one is the choice of the least loaded server for large requests.

In the original method, the server is selected by the least number of active requests. The improved method differs between small requests and large requests. The small requests are processed in the same manner in both methods. For requests with a large payload, the improved method uses the combination of the concurrent write speed and the number of active connections.

The data transmission speed per client is not constant and is changing over time. This might expose the server cluster to denial-of-service attacks. In order to prevent them, the maximum number of connections per server is also limited. If network interface speed is higher than maximum concurrent write speed, it's advisable to limit network bandwidth for the process to this value. Both values can be determined by the method of studying the influence of high load modes on data authorization mechanisms in the operation queue environment.

That's why in the new method for large requests, the server with the least concurrent write speed is selected. If all the servers are loaded at 100% then the server with the fewest active requests is selected. This allows determining which server is the least loaded more accurately.

The modified method improves the reliability, which is a crucial safety factor [30, 31] of cluster systems. Any request sent to the load balancer [32, 33] can be processed by any of the servers in the web cluster. It was improved by adding a step of verification of the current number of active requests and consists of the following steps:

1) the load balancer finds the server with the least concurrent write speed or the fewest requests in the last minute.

2) check whether the current number of active requests can put the system into high load mode and reject it if it's higher than it was found by the method of identifying high load modes

3) the incoming request is transmitted to the selected server.

4) process the request

The method might be simplified for a single server:

1) to get the current number of active requests in the server

2) check whether the current number of active requests can put the system into high load mode and reject it if it's higher than it was found by the method of identifying high load modes

3) process the request

In this work, a new model for processing requests in high load modes was developed, which differs from the existing ones in the ability to work and verify large payloads and to prevent high load modes. This makes it possible to achieve an increase in the speed and reliability of the information system when working with large payloads. Authentication and request processing model in high load modes for IoT components is shown in figure 5.

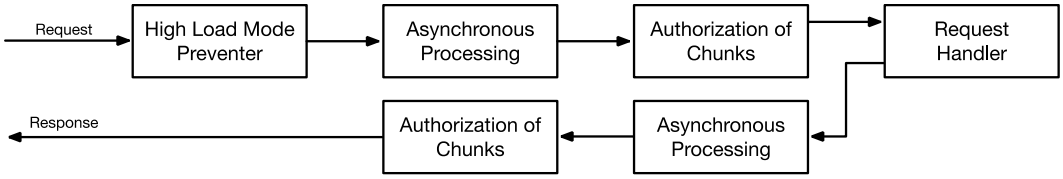

Figure 5: Authentication and request processing model in high load modes for IoT components

In this model, it is assumed that it is known which "Request Handler" processes a new request or a data chunk of the request.

The first stage is "High Load Mode Preventer", which limits the number of requests in the operation queue. Its mechanism uses a simplified version of the method of migration from single server to the server cluster. The maximum number of requests is obtained by the method of identifying the high load modes.

The second stage is "Asynchronous Processing", in which the IoT communication model is implemented, operations are created for processing a data chunk that is added to the operation queue.

The third stage is the authentication of a data chunk according to the "chunking" method. After that, the data goes to the "Request Handler".

On the way back, the Request Handler creates operations for sending chunks of data that are signed before sending. The "High Load Mode Preventer" step is not needed for sending data.

4. Results

Based on the authentication and request processing model in high load modes for IoT components, an information system for the communication of IoT components was created. The central server is responsible for communication between clients and modules. To discover the new modules, the Bonjour discovery protocol is used, after which the server connects to them using the HTTP/2(3) protocol.

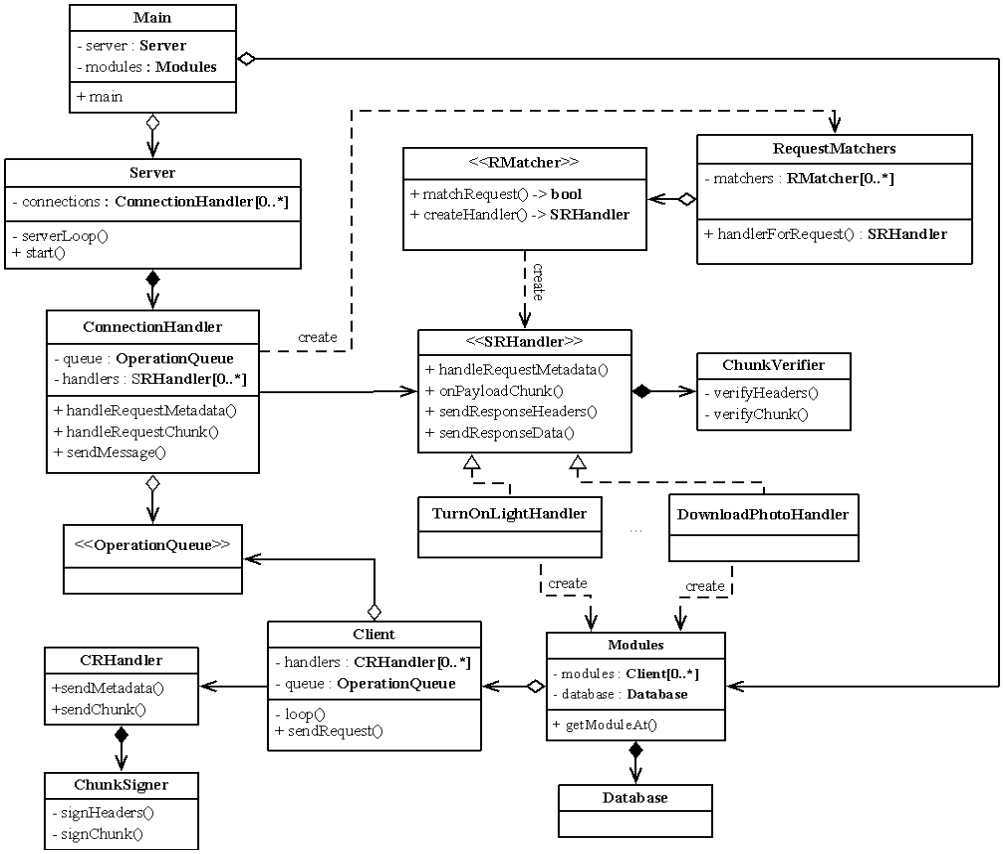

Figure 6: Class diagram of the developed library

For the HTTP/2(3) protocols, a method of chunking authentication of large payloads is implemented in the ChunkVerifier and ChunkSigner classes.

The class diagram of the central server is shown in Figure 6.

To test the new model four experiments were carried out. The model uses the modified method of "method of migration from single server to the server cluster" to implement the "High Load Mode Preventer" component. To control the environment more precisely, the client and server were running on a single computer.

The test computer has the following configuration:

OS: Ubuntu 20.04 LTS

CPU: Intel Core i7 8700K

RAM: 32G

SSD: Samsung 970 EVO Plus 1Tb

The first experiment aims to show the high load mode without any measures of preventing it. As input, 20 clients were chosen with a speed of 1Gbps per client. The results of the experiment are shown in figure 7.

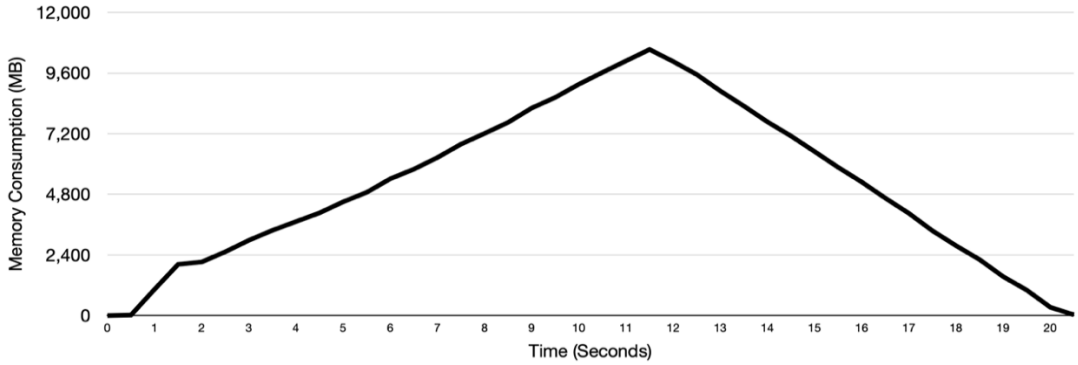

Figure 7: The high load mode without any measures of preventing it

The plot shows linear growth and linear decrease. The peak memory consumption has reached 10534 megabytes, which might potentially cause the server to crash.

The network bandwidth is limited to the maximum drive concurrent write speed for the next series of experiments to prevent high load modes. In the experiments, the maximum concurrent write speed and the maximum number of active clients were determined by the method of studying the influence of high load modes on data authorization mechanisms in the operation queue environment. For SSD, which is used in the test configuration the maximum concurrent write speed is measured around 680 MB/s.

To show the effect of the limitation of network bandwidth in the second experiment the same number of clients is used, but the total network speed is limited to 680 MB/s (5.4 Gbit/s). The results are shown in figure 8.

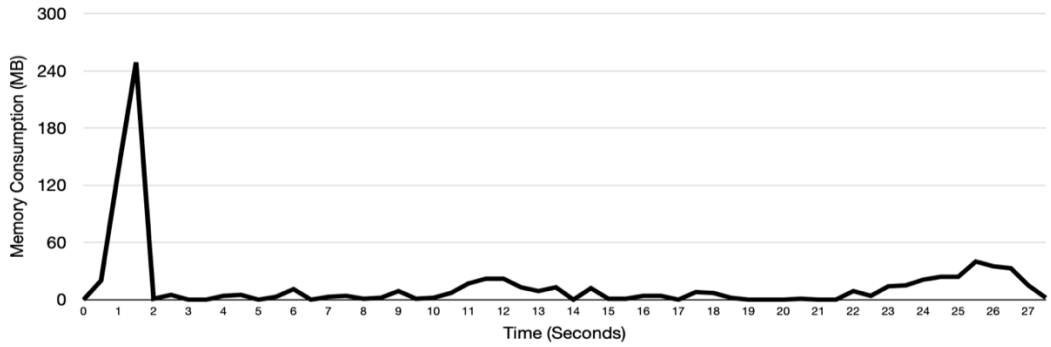

Figure 8: The effect of the limitation of network bandwidth

With the limitation of the network speed, the situation might seem to be stabilized. But the server isn't limited to process a large number of clients. To show the effect of the increased number of clients it was increased to 400. The network speed limit of 680 MB/s (5.4 Gbit/s) remains the same. The results are shown in figure 9.

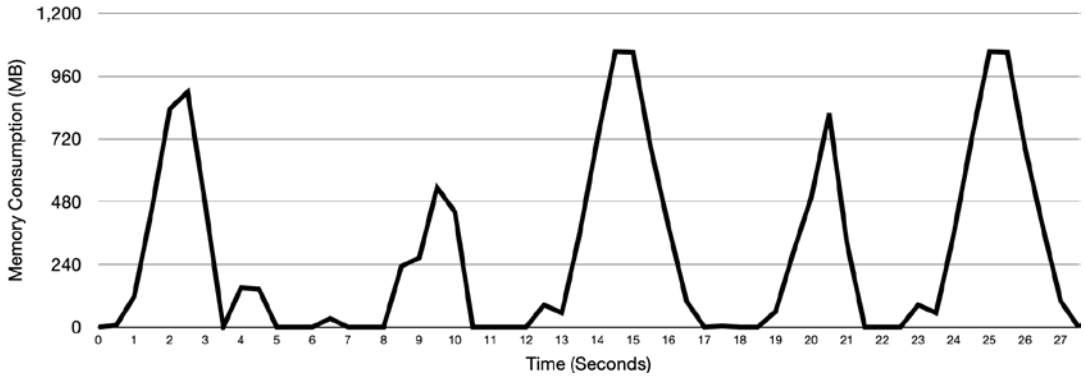

Figure 9: The impact of an increased number of clients with the limitation of network bandwidth

To show the effect of the limitation of the number of active clients it was limited to 121. The network speed limit of 680 MB/s (5.4 Gbit/s) remains the same. The results of the experiment are shown in figure 10.

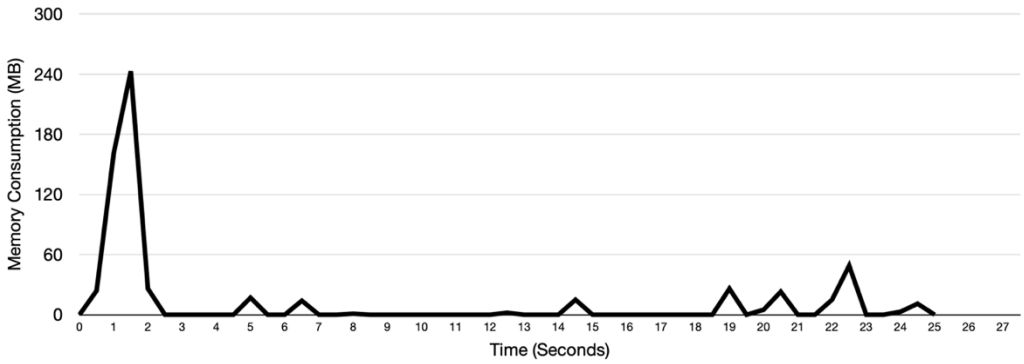

Figure 10: The effect of the limitation of the number of active clients

The modified method of migration from a single server to a server cluster has added the set of measures for preventing high load modes. The series of experiments have proven the effectiveness of them.

The resulting communication model was used in the development of a module for remote control of a PC, which controls "POWER" and “RESET” switches and reads “HDD LED” and “POWER LED” signals using opto-isolators.

The method of building communication mobile systems based on the OAuth 1.0 protocol was further developed, which differs from the existing ones in the ability to work with embedded car computers that do not have their own web browser and allows to achieve an increased security by working directly with standard development tools for third-party applications.

The new method is characterized by the use of the communication model of the Internet of Things and the "chunking" method of data verification for communication between a device with a web browser and a car computer. The method consists of the following steps:

1) On the car computer, get the URL from the web service for authorization in the web browser
2) Initiate a launch on a device a web browser with the received URL
3) After successful authorization by the user, transfer the access token to the car computer
4) On the car computer, verify the validity of the token

The interdependence between the car computer, web browser device, and web service are shown in Figure 11.

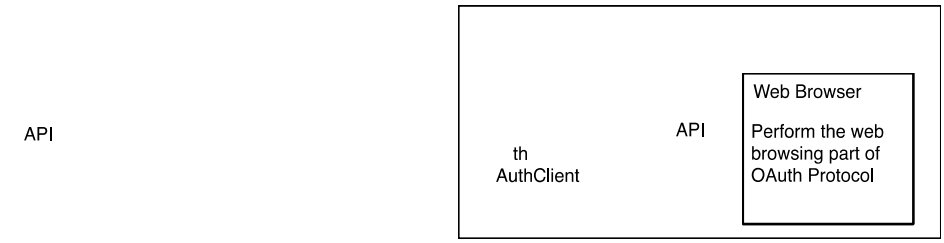

Figure 11: Interdependence between car computer, device with web browser and web service

The introduction of the new method significantly expands the capabilities of car computers.

5. Conclusion

In this work, a relevant scientific and practical problem is solved, associated with increasing the speed and reducing the harmful effect of high load modes for authentication processes and data processing in an operation queue environment. The main results are as follows:

1. Based on the analysis of existing authorization methods with the capability of authenticating large payloads in the operation queue environment, it is concluded that developing and researching new authentication methods of large payloads in the operation queue environment is a relevant task.

2. A model and method for chunking data authentication for large payloads have been developed. Analysis of the new method showed an advantage in performance compared to the existing ones due to the absence of the necessity to read the data again.

3. Based on the analysis of existing models, new models and methods for authentication of large payloads for blocking and non-blocking sockets were created.

4. A method for ranking payload authentication mechanisms was developed, on the basis of which experiments were carried out, which confirmed the advantage of the "chunking" method by 13-22% for a large payloads and 1 percent for small payloads.

5. Methods have been developed to identify the high load modes and study the effect of high load modes on the use of system memory, which made it possible to increase the accuracy of predicting high load modes and improve system reliability by limiting the equipment load.

6. To limit the load on the equipment, the method of load balancing in the server cluster was further developed, which made it possible to reduce the negative effect of the influence of high load modes on the system by improving the distribution of the load between the servers in the cluster.

7. Experiments were carried out to evaluate the new protocol, to establish the requirements for the server cluster operating with the new authorization protocol, which showed that the improvements in the load balancing mechanism prevented falling into high load modes in the researched situations

8. A method has been developed for building communication mobile systems based on the OAuth protocol for embedded car computers that do not have their own web browsers. Based on the method, a module of the Internet of Things information system was created for authorization, built-in by car computers that do not have their own web browsers.

9. The authentication and request processing model in high load modes for IoT components was further developed, on the basis of which the information system of IoT components has been created.

6. References

[1] A. Kupin, I. Muzyka, R. Ivchenko, Information Technologies of Processing Big Industrial Data and Decision-Making Methods, in International Scientific-Practical Conference on Problems of Infocommunications Science and Technology, PIC S and T, (2019), doi: 10.1109/INFOCOMMST.2018.8632096.

[2] O. Pomorova, T. Hovorushchenko, Artificial neural network for software quality evaluation based on the metric analysis, in IEEE East-West Design & Test Symposium, EWDTS'2013, Kharkiv, (2013), pp. 200-203, doi: 10.1109/EWDTS.2013.6673193.

[3] R. Fielding, J. Reschke, Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing, IETF RFC 7230, 2015. URL: https://tools.ietf.org/html/rfc7230

[4] M. Belshe, R. Peon, Hypertext Transfer Protocol Version 2 (HTTP/2), IETF RFC 7540, 2015. URL: https://tools.ietf.org/html/rfc7540

[5] J. M. Kizza, Guide to Computer Network Security Third Edition. Springer, 2015.

[6] Rescorla E., HTTP over TLS, IETF RFC 2818 (Informational), 2014. URL: http://tools.ietf.org/html/rfc2818

[7] Dierks T., The Transport Layer Security (TLS) Protocol. – IETF RFC 5246 2014. URL: http://tools.ietf.org/html/rfc5246

[8] S. Farooqi, F. Zaffar, N. Leontiadis, Z. Shafiq, Measuring and mitigating OAuth access token abuse by collusion networks, in: Communications of the ACM, New York, NY, United States, (2020), vol. 63, pp. 103-111, doi: 10.1145/3387720.

[9] A. Mukherjee, Physical-Layer Security in the Internet of Things: Sensing and Communication Confidentiality Under Resource Constraints, Proceedings of the IEEE, vol. 103, no. 10, pp. 1747-1761, (2015), ISSN: 1558-2256, doi: 10.1109/JPROC.2015.2466548

[10] Apple Inc., Grand Central Dispatch, 2020. URL: https://github.com/apple/swift-corelibs-libdispatch

[11] S. Grosch, Concurrency by Tutorials (Second Edition): Multithreading in Swift with GCD and Operations. McGaheysville, VA, United States Razeware LLC, 2020, p. 100.

[12] D. Maevsky, V. Kharchenko, M. Kolisnyk, E. Maevskaya, Software reliability models and assessment techniques review: Classification issues, in 2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Bucharest, Romania, (2017), pp. 894-899, doi: 10.1109/IDAACS.2017.8095216.

[13] Hovorushchenko T., Pavlova O., Method of Activity of Ontology-Based Intelligent Agent for Evaluating the Initial Stages of the Software Lifecycle., Advances in Intelligent Systems and Computing, vol. Vol. 836, pp. 169-178, (2019), doi: 10.1007/978-3-319-97885-7_17

[14] Y. Feng, M. Sathiamoorthy, A security analysis of the OAuth protocol, in IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), Victoria, BC, Canada, (2013), pp. 271-276, doi: 10.1109/PACRIM.2013.6625487.

[15] A. Lombardi, WebSocket Lightweight Client-Server Communications. O'Reilly Media, 2016.

[16] S. Lysenko, K. Bobrovnikova, O. Savenko, A Botnet Detection Approach Based on The Clonal Selection Algorithm, in Proceedings of 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DeSSerT-2018), Kyiv, Ukraine, (2018).

[17] Hammer-Lahav E., The OAuth 1.0 Protocol. IETF RFC 5849, 2010. URL: http://tools.ietf.org/html/rfc5849.

[18] E. Hammer, HAWK / HTTP Holder-Of-Key Authentication Scheme, 2019. URL: https://github.com/hueniverse/hawk

[19] H. Krawczyk, M. Bellare, HMAC: Keyed-Hashing for Message Authentication, 2011. URL: https://tools.ietf.org/html/rfc2104

[20] E. Hammer, OAuth 2.0 and the Road to Hell, 2012. URL: http://hueniverse.com/2012/07/oauth-2-0-and-the-road-to-hell/

[21] S. S. Surkov, Model and method of chunk processing of payload for HTTP authorization protocols, Proceedings of 2020 IEEE 15th International Conference on Advanced Trends in

Radioelectronics, Telecommunications and Computer Engineering (TCSET), Slavske, Ukraine, pp. 317-321, (2020), doi: 10.1109/TCSET49122.2020.235447
[22] Lysenko S., Bobrovnikova K., Matiukh S., Hurman I., Savenko O., Detection of the botnets' low-rate DDoS attacks based on self-similarity, International Journal of Electrical and Computer Engineering (Q2), vol. 10., №4, pp. 3651-3659, (2020), ISSN: 2088-8708
[23] O. Drozd, M. Kuznietsov, O. Martynyuk, M. Drozd, A method of the hidden faults elimination in FPGA projects for the critical applications, in Proceedings of 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT'2018), Kyiv, Ukraine, (2018), pp. 231 – 234, doi: 10.1109/DESSERT.2018.8409131.
[24] A. Drozd, V. Kharchenko, S. Antoshchuk, J. Sulima, M. Drozd, Checkability of the digital components in safety-critical systems: problems and solutions, in IEEE East-West Design & Test Symposium, Sevastopol, Ukraine, (2011), doi: 10.1109/EWDTS.2011.6116606.
[25] V. M. Kuntsevich, et al. (Eds), Control Systems: Theory and Applications. Book Series in Automation, Control and Robotics: River Publishers, Gistrup/Delft, 2018, pp. 1-326.
[26] A. Kupin, A. Senko, Principles of intellectual control and classification optimization in conditions of technological processes of beneficiation complexes, in CEUR Workshop Proceedings, (2015), vol. 1356, pp. 153–160. URL: http://ceur-ws.org/Vol-1356/paper_34.pdf
[27] S. Surkov, Reduction of the impact of critical modes for authorization protocols for large requests in operation queue enviroment, Applied Aspects of Information Technology, vol. Vol.3 No.3, (2020), doi: 10.15276/aait.03.2020.3
[28] A. Drozd, J. Drozd, S. Antoshchuk, V. Nikul, M. Al-dhabi, Objects and Methods of On-Line Testing: Main Requirements and Perspectives of Development, in Proc. IEEE East-West Design & Test Symposium, Yerevan, Armenia, (2016).
[29] Drozd O., Perebeinos I., Martynyuk O., Zashcholkin K., Ivanova O., Drozd M., Hidden fault analysis of FPGA projects for critical applications, in IEEE International Conference TCSET, Lviv-Slavsko, Ukraine, (2020), doi: 10.1109/TCSET49122.2020.235591.
[30] H. Sanae, Security Requirements and Model for Mobile Agent Authentication, Smart Network Inspired Paradigm and Approaches in IoT Applications, Singapore, Republic of Singapore, pp. 179-189, (2019), doi: 10.1007/978-981-13-8614-5_11
[31] K. Saini, Squid Proxy Server 3.1: Beginner's Guide Paperback. Birmingham, United Kingdom: Packt Publishing, 2011, p. 332.
[32] D. Wessels, Squid: The Definitive Guide. Sebastopol, CA, United States: O'Reilly Media, 2010, p. 472.
[33] M. Rash, Linux Firewalls: Attack Detection and Response with iptables, psad, and fwsnort. San Francisco, CA, United States: No Starch Press, 2007, p. 336.