დაჯავშნის სერვისის არსებობა უცხოელებისთვის საქართველოში და ვებ აპლიკაციების უსაფრთხოების აუცილებლობა

გიორგი ბიძიშვილი
giorgi.bidzishvili285@ens.tsu.ge

რეზიუმე — ონლაინ დაჯავშნის ვებ აპლიკაციების წილი დღითიდღე იზრდება რაც მათი განვითარების და მომხმარებლებზე ორიენტირებულობას აუცილებლობას წარმოშობს, ონლაინს სერვისებიც ვითარდება და უფრო და უფრო კომპლექსური ხდება მათი ინტერფეისი, რაც ნამდვილად უსიამოვნოს ხდის მომხმარებლისთვის პროცესს ძიების, ასევე დღევანდელი სამყაროსთვის დიდ გამოწვევას წამოადგენს ვებ სერვისების დაცულობა, ვინაიდან ყოველდღიურად ათასობით შემოტევა ხორციელდება ჰაკერების მხრიდან, უამრავი მცდელობაა სხვადასხვა ვებ აპლიკაციის მომხმარებლების პირადი ინფორმაციის მოპოვებისა, რათა მოხდეს მათი გამოყენება მზაკვრული მიზნებისათვის, რასაკვირველია, თანამედროვე საიტები მაქსიმალურად დაცული უნდა იყოს მსგავსი შემოტევებისაგან, და ამისათვის უნდა მოხდეს თანამედროვე სტანდარტებისა და მეთოდების გამოყენება. მცირე ონლაინს სერვისებმა საკუთარ თავზე არ უნდა აიღონ მომხმარებელთა ავტორიზაციის სერვერების მართვა, თითოეული მომხმარებლის მოქმედება უნდა იყოს დაფარული და მიუწვდომელი გარე პირებისთვის, რის მიღწევის საშუალებებზეც ოდნავ მოგვიანებით ვისაუბრებთ. რაც შეეხება ჩვენს კონკრეტულ შეთხვევას, შევეხებით დაჯავშნის სერვისის არსებობას უცხოელებისთვის რაც მათ გაუმარტივებს საქართველოში მოგზაურობას, ასევე ვისაუბრებთ ვებ აპლიკაციის უსაფრთხოების აუცილებლობის შესახებ.

Summary — The share of online booking web applications is increasing day by day
What is their development and customer orientation
Necessity arises, online services are also developing and more and more
Their interface becomes complex, which makes it really unpleasant for the user
The web presents a great challenge to the process of search, as well as to today's world
Security of services, as thousands of attacks are carried out every day
On the part of hackers, there are many attempts to impersonate users of various web applications
obtaining information in order to use it for fraudulent purposes,
Of course, modern sites should be protected as much as possible
from attacks, and this should be done according to modern standards and methods
use. Small online services should not take on customers
Management of authorization servers should be an action of each user
Covered and inaccessible to outsiders, little more than the means of achieving it
We will talk later. As for our specific case, we will touch on the existence of a reservation service for foreigners, which will make it easier for them to travel to Georgia.
We will also talk about the need for web application security.

გასაღები სიტვები -ონლაინ სეერვისები, უსაფრთხოება, ფაირვოლი, booking სერვისები HTTPS, Oauth, მიკროსერვისები, მონოლითური არქიტექტურა
Keywords —. firewall, microservices, monolithic architecture, security, booking services, HTTPS, OAuth

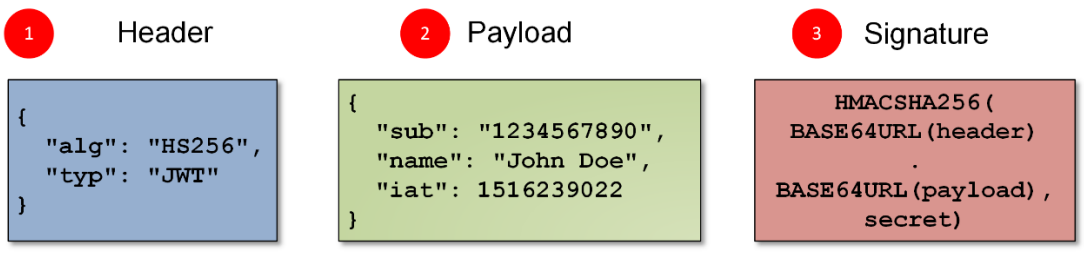
შესავალი

დღევანდელ ვითარებაში უფრო და უფრო აქტუალური ხდება ონლაინ სერვისების [1]

გამოყენება, ონლაინ სერვისი ინტერნეტის საშუალებით მომხმარებლებს აძლევს წვდომას სხვადასხვა სახის ინფორმაციაზე, მსგავსი სერვისი მაგალითია Online Banking, საგანმანათლებლო, shopping, სიახლეების საიტები, სოციალური ქსელები, e-mail-ები და სხვა. ზოგიერთი მათგანის გამოყენება, როგორიცაა Netflix [2] და Microsoft Office [3] , ფასიანია ზოგიც უფასოა, მაგალითად Wikipedia [4]. რასაკვირველია, 21-ე საუკუნეში ბევრი სახის ონლაინ სერვისი არსებობს, მაგრამ მოცემულ სიტუაციაში ჩვენი განხილვის თემას წარმოადგენს online booking მიმართულება მაგალითად როგრიცაა Booking.com, სადაც მომხმარებლებს შეუძლიათ შეუფერხებლად დაგეგმონ თავიანთი მოგზარუბა.
სწორი არქიტექტურის შერჩევა გადამწყვეტია ვებ აპლიკაციის წარმატებისთვის. მიუხედავად იმისა, რომ მიკროსერვისები გვთავაზობენ მასშტაბურობას და მოდულარობას, მონოლითური არქიტექტურა უზრუნველყოფს მართვის სიმარტივეს, ამასთან ერთად უნდა აღინიშნოს ისიც რომ მიკროსერვისლი არქიტეტურის მქონე აპლიკაციის შენარჩუნება დაკავშირებული დიდ თანხებთან. სწორედ ამიტომ ჩვენი კონკრეტული შემთხვევისთვის, მონოლითური არქიტექტურის არჩევანი მიზანმიმართულია.
ის აადვილებს დეველოპმენტს, აპლიაკციის გაშვებას და მის შენარჩუნებას, რაც უზრუნველყოფს ბაზარზე გასვლის ეფექტურ სისწრაფეს.
მონოლითის ერთიანი კოდის ბაზა ხელს უწყობს JWT(JSON Web Token) ტოკენების შეუფერხებელ ინტეგრაციას, რაც უზრუნველყოფს აუთენთიფიკაციისა და ავტორიზაციის ცენტრალიზებულ და მარტივ მიდგომას.

პრობლემების გადაჭრის გზები

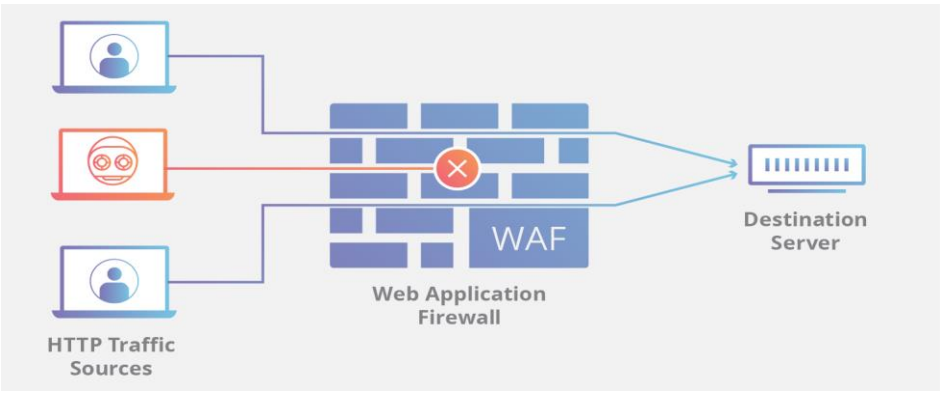
1.1 უსაფრთხო ინფრასტრუქტურის გამართვა ვებ საიტებზე
ამ პრობლემაზე მუშაობის გადაწყვეტილების მიღება არც თუ ისე რთული იყო ეს კი განპირობებულია რამდენიმე მიზეზით ერთ ერთი უმთავრესი არის ის რომ ონლაინ სერვისებს პოპულარობა დღითიდღე იმატებს და ამასთან ერთად მომხმარებელთა მოთხოვნილებებიც.
რა თქმა უნდა, მომხმარებლების ნდობის მოსაპოვებლად საჭიროა, რომ აპლიკაცია იყოს უსაფრთხო მომხმარებლებისთვის და რაც შეძლება დაბალი იყოს მათ მიერ მოწოდებული პირადი ინფორმაციის არასასურველი პირებისთვის ხელში ჩაგდების რისკი. ამიტომ საჭიროა აპლიკაციის შექმნის დროს მაქსიმალურად შევეცადოთ გავხადოთ იგი დაცული, მაგრამ როგორ მოვახერხოთ ეს ?
1.2 JWT(JSON Web Token) არის უსაფრთხო და მოქნილი, ის იკრიპტება, კრიპტი შედგება 3 ნაწილისგან: 1) Header - სადაც წერია შიფრირების ალგორითმი 2) Payload - ში ინახება მონაცემები რომლებიც სურვილისამებრ შეგვიძლია გავატანოთ ტოკენს, მაგალითად მომხმარებლის სახელი, გვარი და როლები 3) Signature(ხელმოწერა) - რომელიც იქმენა შემდეგი პრინციპით:
ენკოდირებული ჰედერი + ენკოდირებული Payload + secretkey
გამოიყენება იმის დასადასტურებლად, რომ JWT-ის გამგზავნი არის ის, ვინც ამბობს, რომ არის.



1.3 ავტორიზაციის მოთხოვნა გარკვეულ რესურსებზე წვდომისათვის მცირე გამონაკლისი შემთხვევების გარდა თითოეულ მომხმარებელს უნდა მოვთხოვოთ ავტორიზაცია გაიაროს, რათა მოახდინოს გარკვეული მოქმედებები და ჩვენც მოვახდინოთ ვალიდაცია აქვს თუ არა ამ კონკრეტულ მომხმარებელს უფლება განახორციელოს სასურველი მოქმედება. აუცილებელია მომხმარებლებს არ მივანიჭოთ იმაზე მეტი უფლება ვიდრე საჭიროა მისთვის რათა თავი დავიზღვიოთ არასასურველი შემთხვევებისგან. თანამედროვე ტექნოლოგიებში მომხმარებელთა ავტორიზაციისა და ავთენტიფიკაციისათვის მენეჯმენტი უმჯობესია პატარა ორგანიზაციებმა, დიდი გამოცდილების მქონე კომპანიებს მიანდონ, მაგალითად Google-ს ან Red Hat-ს და Single-sign on [5] ტექნოლოგის გამოყენებით OAuth-ით [6] დავაკავშიროთ მომხმარებლები ჩვენს აპლიკაციასთან. ეს მცირე ორგანიზაციისთვის გაამარტივებს მომხმარებელთა მენეჯმენტს და ამავდროულად ვებ აპლიკაციასაც უფრო დაცულს გახდის.

1.4 გამოიყენეთ Firewall

ვებსაიტის დაცულობისათვის აუცილებელი გამოვიყენოთ Firewall-ები, WAF [7] ან SDN [8] ვებ აპლიკაციის ყველა ფენაზე შემომავალ რექვესთებზე, რათა მაქსიმალურად მოვახდინოთ საფრთხის შემცველი რექვესთების ამოცნობა და დაუყოვნებლივ ბლოკირება.

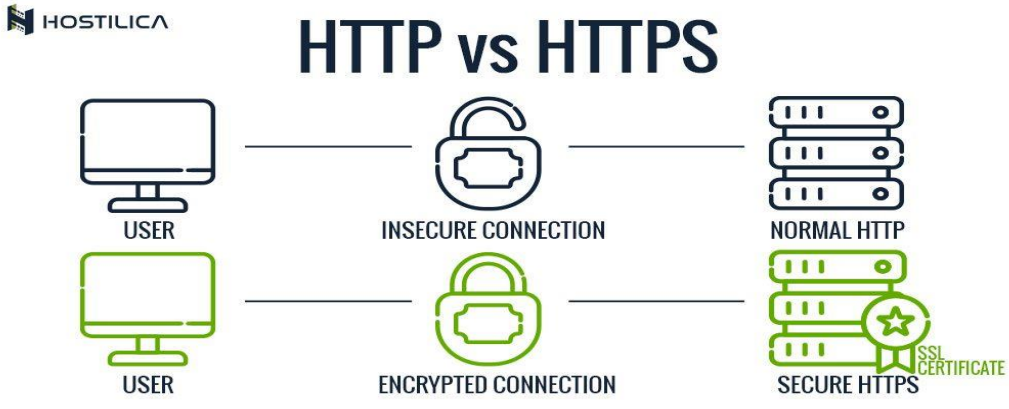


1.5 Multi-factor Authentication (MFA)

ვებსაიტის დაცულობისათვის საჭიროა მომხმარებლებს მაქსიმალურად მოვუწოდოთ მრავალ დონიანი ავთენტიფიკაციის გამოყენებისაკენ, რადგან ამ მიდგომის გამოყენება საგრძნობლად შეუწყობს ხელს მათი ანგარიშების დაცვას მრავალი სახის ავტორიზაციის შემოტევებისაგან, იმ შემთხვევაშიც კი თუ რაიმე გზით ჰაკერი მოიპოვებს მათი ანგარიშის პაროლს ის ვერ მოახერხებს ანგარიშზე შესვლას დამატებითი ინფორმაციის წარდგენის გარეში, რაც სპეციფიკიდან გამომდინარე შესაძლოა იყოს მომხმარებლის თითის ანაბეჭდი ან სულაც ერთჯერადი კოდი რომელიც მომხმარებელს შეტყობინების სახით მისდის.

1.6 Enable HTTPS on All Pages

დაცულობის კიდევ ერთ ძირითად გზას წარმოადგენს ყველა გვერდზე კავშირისთვის HTTPS პროტოკოლის გამოყენება, რათა უზრუნველყოფილი იყოს საიტის ვიზიტორების დაცულობა საფრთხის შემცველი Cookie შემოტევებისგან, პაროლების მოპარვის მცდელობისა თუ სახვა მრავალი საფრხისაგან. ამ ყველაფრის გასაკეთებლად აუცილებელია დააყენოთ ვებსაიტის უსაფრთხოების სერთიფიკატი ანუ SSL/TCL სერთიფიკატი თქვენს სერვერზე.



მომავალი სამუშაო

მომავალ სამუშაოებს რაც შეეხება ვფიქრობ რომ უახლოეს მომავალში booking ვებ აპლიკაციების ჭრილში დაემატება ფუნცქიონალი რისი საშუალებითაც მოხმარებელბი შეძლებენ მოგზაურობასთან ერთად დაგეგმონ ქემფინგიც სასურველ ლოკაციაზე.
უსაფრთხოების კუთხით უახლეს მომავალში წინგადადგმულ ნაბიჯად ჩაითვლება, მცირე მასშტაბის საიტების მიერ არსებული გადახდის სისტემების Google Pay [9] ჩანაცვლებით, რაც შედარებით გაამარტივებს ამ პროცეს და ამავდროულად საკრედიტო ინფორმაციის გამჟღავნების ნაკლები რისკის წინაშე დავდგებით ვინაიდან Google ტრანსფერამდე ახდენს საკრედიტო ინფორმაციის ჰეშირებას[10], ანუ პირდაპირი მონაცემები არ გამოიყენება შესაბამისად რომც მოხდეს აპლიკაციიდან ამ ინფორმაციის გაჟონვა ინფორმაციის მომპოვებელი მაინც ვერ მოახერხებს მის გამოყენებას, რადგან ის დაჰეშილი იქნება და მის პირვანდელ სახეზე დაბრუნება თითქმის შეუძლებელი.

დასკვნა

დაჯავშნის სერვისის არსებობა უცხოელებს გაუმარტივებს მოგზარუობის დაგეგმის პროცესს საქართველოში, ისინი შეძლებენ დაჯავშნონ საიტზე განთავსებული სასურველი სასტუმრო და ოთახი. ამავდროულად საიტის სიმარტივისა და მომხმარებელზე მორგებულობის გარდა მნიშვნელოვანია უსაფრთხოების ასპექტებიც, ვინაიდან თუ გინდა მომხმარებლისთვის სანდო იყო უნდა გქონდეს უსაფრთხო უახლესი მიდგომებით დაცული სისტემა, რომელიც სრულებით უზრუნველყოფს იმას რომ მომხმარებელმა კომფორტულად და დაცულად იგრძნოს თავი ჩვენს მიერ შეთავაზებული სერვისის გამოყენების დროს.

ლიტერატურა

[1] B. Vangie, „Online Services,“ https://www.webopedia.com/, [ინტერნეტი]. Available: https://www.webopedia.com/definitions/online-service/.

[2] „Netflix,“ Netflix, [ინტერნეტი]. Available: https://www.netflix.com/.

[3] „Microsoft Office,“ [ინტერნეტი]. Available: https://www.microsoft.com/en-us/microsoft-365/get-started-with-office-2021.

[4] „Wikipedia,“ [ინტერნეტი]. Available: https://www.wikipedia.org/.

[5] jacquelinelam98, „What is SIngle-sign on?,“ [ინტერნეტი]. Available: https://www.geeksforgeeks.org/introduction-of-single-sign-on-sso/.

[6] M. Raible, „What the Heck is OAuth?,“ [ინტერნეტი]. Available: https://developer.okta.com/blog/2017/06/21/what-the-heck-is-oauth.

[7] cloudflare, „What is a WAF? | Web Application Firewall explained,“ [ინტერნეტი]. Available: https://www.cloudflare.com/learning/ddos/glossary/web-application-firewall-waf/.

[8] vmware, „What is Software-Defined Networking (SDN)?,“ [ინტერნეტი]. Available: https://www.vmware.com/topics/glossary/content/software-defined-networking.html.
სს

[9] „Google Pay,“ [ინტერნეტი]. Available: https://pay.google.com/intl/en_in/about/.

[10] P. Greenamyer, „Hashing vs encryption vs salting: what's the difference?,“ [ინტერნეტი]. Available: https://cybernews.com/security/hashing-vs-encryption/.