## A ტიპის ქსელური მარშუტიზატორი 2(ორი ცალი)

მარშრუტიზატორს უნდა გააჩნდეს შემდეგი ტექნიკური მახასიათებლები:

1. მარშრუტიზატორი უნდა იყოს მულტისერვისული, მომავალში პროგრამული და აპარატურული კომპონენტების დამატებით, საკუთრივ მარშრუტიზატორს უნდა გააჩნდეს:
	1. **ხმოვანი მარშრუტიზატორის მხარდაჭერა:**
		1. უნდა შეეძლოს ხმოვანი სიგნალიზაციის (SIP და H323) ოქმებით განხორციელებული ზარების მარშრუტიზაცია;
		2. უნდა შეეძლოს ხმოვანი სიგნალიზაციის (SIP და H323) ოქმების უსაფრთხოების უზრუნველყოფა;
		3. საჭიროების შემთხვევაში უნდა გააჩნდეს სპეციალური აპარატურული მოდულების მხარდაჭერა, რათა აღნიშნული მოდულების დონეზე განხორციელდეს სატელეფონო ზარების დამუშავება. მათ შორის ხმოვანი კოდეკების ტრანსლირება, DTMF -ის კონვერტირება, ციფრული PSTN ქსელიდან TDM ტექნოლოგიით მიღებული ზარის კონვერტაცია IP ქსელში;
	2. **IP სათელეფონო სადგურის როლის შესრულების შესაძლებლობა:**
		1. IP ტელეფონების მიერთება და მათი მართვა;
		2. სატელეფონო ნომრების მინიჭება, IP ტელეფონებზე;
		3. IP სათელეფონო სადგურის მართვა შესაძლებელი უნდა იყოს გრაფიკული ინტერფეისიდან;
	3. **ახალი თაობის NGFW/NGIPS ბრანდმაუერის მხარდაჭერა:**
		1. დაცვა რეალურ დროში;
		2. პასიური და აქტიური რეჟიმების მხარდაჭერა;
		3. ქსელის ტოპოლოგიის აღმოჩენა;
		4. დაცვის რეჟიმი: 2-დან 7-ე შრემდე (Layer);
		5. SSL ტრაფიკის დეშიფრაციის მხარდაჭერა და მისი შიგთავსისი ინსპექტირების საშუალება;
		6. უნდა უზრუნველყოფდეს ე.წ IDS/IPS Tuning-ს, ადამიანის მინიმალური ჩარევით (მაგ.: წესების შერჩევის, პოლიტიკების კონფიგურაცია, პოლიტიკების განახლება და ა.შ.);
		7. უნდა შეეძლოს ფაილების აღმოჩენა და კონტროლი. აღნიშნული ოპერაციების განხორციელება შესაძლებელი უნდა იყოს ფაილების ტიპების, გამოყენებული პროტოკოლისა და ე.წ „ატვირთვა“ „ჩამოტვირთვის“ მიხედვით
		8. უნდა შეეძლოს პასიურად შეაგროვოს ინფორმაცია ქსელის ჰოსტებისა და მათი ქცევის შესახებ. მათ შორის: ოპერაციული სისტემები, სერვისები, ღია პორტები, აპლიკაციები, სისუსტეები( vulnerabilities). აღნიშნული ინფორმაციის გამოყენება შესაძლებელი უნდა იყოს, უსაფრთხოების ინციდენტების კორელაციისთვის, ე.წ „false positives“ ინციდენტების შესამცირებლად და პოლიტიკების შესაბამისობაში მოყვანისთვის;
		9. გადაწყვეტილებამ უნდა უზრუნველყოს რეალურ დროში ქსელური აქტივების გზამკვლევის აგება. მას უნდა შეეძლოს ისეთი დეტალური ინფორმაციის დადგენა აქტივების შესახებ, როგორიცაა, აქტივის მომხმარებლები, აქტივის მიერ უზრუნველყოფილი სერვისები და გამოყენებული აპლიკაციები.
		10. გადაწყვეტილებას უნდა შეეძლოს ისეთი მობილური მოწყობილობების აღმოჩენა და კლასიფიცირება, როგორებიცაა iPad, iPhone, Blackberry. გადაწყვეტილებას უნდა შეეძლოს აღმოჩენილი მოწყობილობების დაკავშირება მომხმარებლებთან და აპლიკაციებთან. გადაწყვეტილებას უნდა შეეძლოს ამოიცნოს Apple ის ე.წ „Jailbroken“ მოწყობილობები;
		11. გადაწყვეტილებას უნდა შეეძლოს, ქსელში აღმოჩენილ აპლიკაციებს მიუსადაგოს რისკის დონე და ბიზნესის რელევანტობა;
		12. გადაწყვეტილებას უნდა შეეძლოს ქსელური ნაკადებში ამოიცნოს სხვადასხვა აპლიკაციები და თვალყური ადევნოს მათ. გადაწყვეტილებას უნდა შეეძლოს ამოიცნოს, მომხმარებლის აპლიკაციები, სერვერის აპლიკაციები, ვებ აპლიკაციები და აპლიკაციების ოქმები;
		13. გადაწყვეტილებას უნდა შეეძლოს პასიურად განახორციელოს ინფორმაციის შეგროვება ქსელური ნაკადების შესახებ, მათ შორის სესიის დაწყებისა და დამთავრების დრო, კომუნიკაციის პორტები, სერვისები, გადაცემული ინფორმაციის მოცულობა;
		14. გადაწყვეტილებას უნდა შეეძლოს პასიურად განახორციელოს მომხმარებლების მაიდენტიფიცირებელი ინფორმაციის შეგროვება. მას უნდა შეეძლოს IP მისამართების მისადეგება მომხმარებლების სახელებზე, აღნიშნული ინფორმაცია ხელმისაწვდომი უნდა იყოს შეტყობინების მენეჯმენტისთვის;
		15. გადაწყვეტილებას უნდა შეეძლოს დაადგინოს თუ როგორ გამოიყენებენ ქსელურ რესურსებს მომხმარებლები, მოწყობილობები და აპლიკაციები;
		16. გადაწყვეტილებას უნდა შეეძლოს დროში გავრცობილად ახორციელოს ქსელის პროფილირება და შესაბამისი რეკომენტაციების გაწევა ახალი და განახლებული უსაფრთხოების წესებისთვის. გადაწყვეტილებას უნდა შეეძლოს ახალი სიგნატურების ავტომატურად მიღება და უცნობი აპლიკაციების ბლოკირება;
		17. ქსელური შეტევების აღმოჩენის წესები უნდა ეფუძნებოდეს გაფართოებად ღია სტანდარტს, რომლის მეშვეობითაც მომხმარებელს უნდა შეეძლოს როგორც მწარმოებლის მიერ შემუშავებული წესების მოდიფიცირება ასევე ახალი, მისთვის სასურველი წესების შექმნა;
		18. გადაწყვეტილებას უნდა გააჩნდეს SNORT ენაზე დაფუქნებული სრულფასოვანი წესების მხარდაჭერა. გადაწყვეტილებაში შესაძლებელი უნდა იყოს არანაკლებ 2000 SNORT წესის გამოყენება;
		19. სისტემას უნდა ჰქონდეს შესაძლებლობა მოახდინოს ტრაფიკის ინსპექტირება ქცევის მიხედვით;
		20. შეტევების აღმომჩენ ძრავს უნდა შეეძლოს როგორც ცნობილი საფრთხეების, ასევე ახალი, ჯერ უცნობი საფრთხეების აღმოჩენა;
		21. შეტევების აღმომჩენ ძრავს უნდა გააჩნდეს შეტევების აღმოჩენის რამოდენიმე მექანიზმი, მათ შორის ე.წ „exploit” ებზე დაფუძნებული სიგნატურები, სისტემების ე.წ „სისუსტეებზე“ დაფუძნებული წესები, ოქმების ანომალიები და ქცევის ანომალიების მიხედვით;
		22. გადაწყვეტილებას უნდა შეეძლოს ქსელის სხვადასხვა სეგმენტების ტრაფიკის ინსპექტირება სხვადასხვა პოლიტიკებით. ერთ ინტერფეისზე შესაძლებელი უნდა იყოს სხვადასხვა უსაფრთხოების პოლიტიკების გამართვა;
		23. გადაწყვეტილებას უნდა შეეძლოს IPv6 ოქმით მიმდინარე შეტევების აღმოჩენა;
		24. გადაწყვეტილებას უნდა შეეძლოს პასიურად, უწყვეტ რეჟიმში ახორციელოს ქსელური ჰოსტების შესახებ ინფორმაციის შეგროვება. მათ შორის კლიენტის ოპერაციული სისტემების, ე.წ „ვებ ბრაუზერი“-ს, ვირტუალური გარემოს და მობილური მოწყობილობების აღმოჩენა;
		25. გადაწყვეტილებას უნდა შეეძლოს გარე შეერთებების რეპუტაციის დადგენა. გადაწყვეტილება რეგულარულად და ავტომატურად უნდა ანახლებდეს რეპუტაციის ინფორმაციას. სისტემის ადმინისტრატორს უნდა შეეძლოს საკუთრივ განსაზღვროს ე.წ „blacklists/whitelists“ სიები. გადაწყვეტილებას უნდა შეეძლოს საჯარო და კერძო რეპუტაცის ე.წ „Feed” სერვისებთან მიერთება;
		26. გადაწყვეტილებას უნდა შეეძლოს სხვადასხვა მომხმარებლებისთვის ან მათი ჯგუფებისთვის განისაზღვროს სხვადასხვა უსაფრთხოების პოლიტიკები. მომხმარებლების მაიდენტიფიცირებელი ინფორმაციის დადგენა შესაძლებელი უნდა იყოს Active Directory, LDAP სერვერებიდან;
		27. გადაწყვეტილებას უნდა გააჩნდეს უნივერსალური და მოქნილი უსაფრთხოების პოლიტიკების აგების შესაძლებლობა. პოლიტიკებში შესაძლებელი უნდა იყოს სხვადასხვა სახის ინფორმაციის მითითება მათ შორის: ქსელების, ზონების, აპლიკაციების კონტროლის, მომხმარებლების კონტროლის, ფაილების კონტროლის, ჰოსტების წვდომის და IPS სისტემის პოლიტიკების;
2. მოთხოვნები ქსელური ინტერფეისების მიმართ:
	1. ინტერფეისების რაოდენობა არანაკლებ 3(სამი)10/100/1000მბ/წმ წარმადობის:
		1. მათ შორის - 2 (ორი)Rj-45 ტიპის, 1 (ერთი)SFP ტიპის; (ორმაგი დანიშნულების);
3. მარშრუტიზატორის წარმადობა, არანაკლებ 100მბ. მომავალში მხოლოდ პროგრამული ლიცენზიის დამატებით შესაძლებელი უნდა იყოს მარშრუტიზატორის წარმადობის გაზრდა 300მბ მდე.
4. DRAM ტიპის მეხსიერება არანაკლებ 4გბ. მომავალში გაფართოებადი 16 გბ მდე.
5. FLASH ტიპის მეხსიერება არანაკლებ 4გბ. მომავალში გაფართოებადი 16 გბ მდე.
6. მარშრუტიზატორს უნდა გააჩნდეს ქსელური ბრანდმაუერის ფუნქციონალი :
	1. მარშრუტიზატორს უნდა გააჩნდეს ე.წ „stateful” ბრანდმაუერის ფუნქციონალი;
	2. ბრანდმაუერზე შესაძლებელი უნდა იყოს, რამოდენიმე უსაფრთხოების ზონების განსაზღვრა და მათი გათვალისწინებით ტრაფიკის ინსპექტირება;
	3. ბრანდმაუერს უნდა გააჩნდეს აპლიკაციის დონეზე(Layer 7) ოქმების ინსპექტირებისა და კონტროლის შესაძლებლობა, ე.წ “Application Layer Gateway (ALG)” და „Application Inspection and Control(AIC)” ფუნქციონალი;
7. მომავალში შესაძლებელი უნდა იყოს მარშრუტიზატორის ფუნქციონალის გაზრდა, მხოლოდ პროგრამული ლიცენზიის დამატების საშუალებით, რის შემთხვევაშიც მას გაუაქტიურდება ე.წ “Application Visibility and Control(AVC)” ფუნქციონალი:
	1. AVC ფუქნციონალს უნდა შეეძლოს ქსელური აპლიკაციების ამოცნობა „Deep Packet Inspection (DPI)” ტექნოლოგიით. განურჩევლად იმისა თუ რომელ პორტზე მუშაობს აპლიკაცია. მარშრუტიზატორს უნდა გააჩნდეს არანაკლებ 800 აპლიკაციის ამოცნობის შესაძლებლობა. აპლიკაციების სიგნატურების განახლება არ უნდა აფერხებდეს მარშრუტიზატორის მუშაობას;
	2. აღნიშნული ფუნქციონალის ფარგლებში შესაძლებელი უნდა იყოს აპლიკაციების გამოყენებისა და ქსელური წარმადობის შეფასების სტატისტიკის შეგროვება;
	3. აღნიშნული ფუნქციონალის ფარგლებში შესაძლებელი უნდა იყოს თითოეული აპლიკაციისთვის QoS ქსელური რესურსების გამოყენების პოლიტიკების დაწესება. ისეთი პოლიტიკების მხარდაჭერა, როგორებიცაა shape, police, priority;
	4. შესაძლებელი უნდა იყოს აპლიკაციების გონიერი და ოპტიმალური მარშრუტიზაცია. მარშრუტიზაცია უნდა ხორციელდებოდეს მარშრუტის მიმდინარე წარმადობაზე დაყრდნობით;
8. მარშრუტიზატორს საკუთრივ უნდა გააჩნდეს სპეციალური მექანიზმები, რათა განახორციელოს ქსელში კავშირის ხარისხის შეფასება, სხვადასხვა სახის აპლიკაციებისათვის. შესაძლებელი უნდა იყოს აღნიშნული მექანიზმის გაშვება პერიოდული განრიგის მიხედვით ან მუდვიმად:
	1. აღნიშნულ მექანიზმებს უნდა გააჩნდეს, შემდეგი სახის ქსელის წარმადობის პარამეტრების დადგენის შესაძლებლობა:
		1. Delay (round-trip და one-way)
		2. Jitter (directional)
		3. Packet loss (directional)
		4. Packet sequencing (packet ordering)
		5. Path (per hop)
		6. Connectivity (directional)
		7. Server or website download time
	2. აღნიშნული მექანიზმით შესაძლებელი უნდა იყოს შემდეგი სახის ოქმებისა და აპლიკაციებისთვის, კავშირის ხარისხის შეფასება და წარმადობის პარამეტრების დადგენა:
		1. TCP
			1. response time
		2. UDP
			1. round-trip delay
			2. one-way delay
			3. one-way jitter
			4. one-way packet loss
			5. connectivity
		3. ICMP
			1. response time
			2. end-to-end და hop-by-hop response time
		4. HTTP
			1. response time between to retrieve a web page from HTTP server
		5. FTP
			1. response time to retrieve a file from FTP server
		6. DHCP
			1. measure the response time to obtain an IP address from DHCP server
		7. DNS
			1. difference between the time taken to send a DNS request and receive a reply
	3. აღნიშნულ მექანიზმს უნდა შეეძლოს, ზუსტად შეაფასოს ქსელის მუშაობის შემდეგი პარამეტრები:
		1. Per-direction jitter (source to destination and destination to source)
		2. Per-direction packet loss
		3. Per-direction delay (one-way delay)
		4. Round-trip delay (average round-trip time)
	4. აღნიშნული მექანიზმის მონიტორინგი შესაძლებელი უნდა იყოს SNMP ოქმის საშუალებით
9. მარშრუტიზატორს უნდა გააჩნდეს ტრაფიკის ოპტიმალური მარშრუტიზაციის ტექნოლოგიის მხარდაჭერა. იგი რეალურ დროში უნდა აფასებდეს მარშრუტის ხარისხის ისეთ პარამეტრებს, როგორებიცაა: delay, loss, jitter, throughput, available bandwidth, reachability და მათზე აგებული პოლიტიკების მიხედვით უზრუნველყოფდეს აპლიკაციების ტრაფიკის ოპტიმალურ მარშრუტიზაციას:
	1. მარშრუტიზატორს უნდა შეეძლოს, გონივრული, თითოეული აპლიკაციისთვის ოპტიმალური მარშრუტიზაციის განხორციელება, ისეთი პარამეტრების მიხედვით, როგორებიცაა: delay, loss, jitter. მარშრუტიზატორს უნდა შეეძლოს მე - 7 პუქტში აღწერილი, ფუნქციონალის გამოყენება რათა განახორციელოს ქსელური აპლიკაციების ამოცნობა;
	2. მარშრუტიზატორს უნდა შეეძლოს პასიურად განახორციელოს მარშრუტის შეფასება. ამისათვის ის პასიურად უნდა ახდენდეს ტრანზიტული ტრაფიკის ანალიზს და ისეთი სახის პარამეტრების დადგენას, როგორებიცაა delay, packet loss, reachability, throughput. აღნიშნული სახის პარამეტრების გამოყენებით მარშრუტიზატორს უნდა შეეძლოს ოპტიმალურად განახორციელოს მარშრუტიზაცია;
	3. მარშრუტიზატორს უნდა შეეძლოს აქტიურად განახორციელოს მარშრუტის შეფასება, მე 7 პუნქტში აღწერილი ტექნოლოგიის გამოყენებით, რათა მის შედეგებზე დაყრდნობით განახორციელოს ტრაფიკის ოპტიმალური მარშრუტიზაცია;
	4. მარშრუტიზატორს უნდა შეეძლოს კომბინირებულად, როგორც აქტიური ასევე პასიური მარშრუტის შეფასების ტექნოლოგიების გამოყენება ოპტიმალური მარშრუტის ასარჩევად
10. მარშრუტიზატორს უნდა გააჩნდეს შემდეგი QoS ტექნოლოგიების მხარდაჭერა: HQoS, Classification, marking, Weighted Random Early Detection (WRED), policing, shaping, priority, and Class-Based Weighted Fair Queuing (CBWFQ) scheduling;
11. საკუთრივ მარშრუტიზატორს უნდა გააჩნდეს სპეციალური დიაგნოსტიკური საშუალება, რომლის მეშვეობითაც ადმინისტრატორს შეეძლება: ტრანზიტული, მოწყობილობაზე მიმართული და მოწყობილობიდან გამოსული ტრაფიკის შეგროვება. შეგროვებული ტრაფიკის ანალიზი შესაძლებელი უნდა იყოს ლოკალურად ან მისი ექსპორტირებით გარე სამუშაო სადგურზე. შეგროვებული ტრაფიკის ანალიზი შესაძლებელი უნდა იყოს Wireshark პროგრამული უზრუნველყოფით;
12. საკუთრივ მარშრუტიზატორში მოთხოვნისამებრ ადმინისტრატორს უნდა შეეძლოს ე.წ „სკრიპტი“ -ის პოლიტიკების დაწერა, აღნიშნული ფუნქციონალით შესაძლებელი უნდა იყოს გარკვეული ამოცანების ავტომატიზირება მარშრუტიზატორში. პოლიტიკებს უნდა შეეძლოს მარშრუტიზატორში გარკვეული ინფორმაციის მონიტორინგი და მათ საფუძველზე წინასწარ განსაზღვრული ქმედებების შესრულება:
	1. აღნიშნულ პოლიტიკებს უნდა შეეძლოს შემდეგი სახის ინფორმაციის მონიტორინგი და წინასწარ განსაზღვრული პოლიტიკის მიხედვით მარშრუტიზატორში სკრიპტის აქტივაცია:
		1. Counter - წინასწარ განსაწღვრული მთვლელის მონიტორინგი
		2. Interface Counter - ინტერფეისზე სხვადასხვა სტატისტიკური მონაცემისთვის განსაზღვრული მთვლელების მონიტორინგი
		3. Timer - დროის პოლიტიკის მონიტორინგი
		4. Watchdog System Monitor - CPU ან მეხსიერების მონიტორინგი
		5. CLI - ტექსტური ინტერფეისით შეყვენილი ბრძანებების მონიტორინგი
		6. None - შესაძლებელი უნდა იყოს სკრიპტის პოლიტიკის ხელით გაშვება
		7. OIR - Online insertion and removal (OIR) ის მონიტორინგი
		8. SNMP– SNMP Object და SNMP Trap ის მონიტორინგი
		9. Routing – მარშრუტიზაციის ცხრილის მონიტორინგი
		10. მე 7 პუნქტში მოთხოვნილი ტექნოლოგიის მონიტორინგი
	2. აღნიშნულ ტექნოლოგიას უნდა გააჩნდეს ე.წ “if” და ”goto” ლოგიკური ოპერაციების მხარდაჭერა
	3. სკრიპტით შესაძლებელი უნდა იყოს შემდეგი ქმედებების განხორციელება:
		1. წინასწარ განსაზღვრული ბრძანებების გაშვება
		2. შეტყობინების გენერაცია SYSLOG და SNMP ოქმების საშუალებით
13. მარშრუტიზატორს უნდა გააჩნდეს ტრაფიკის ნაკადების აღრიცხვის ტექნოლოგიის მხარდაჭერა. ადმინისტრატორს მოთხოვნისამებრ უნდა შეეძლოს მოქნილი შაბლონების განსაზღვრა, რომლებშიც აღწერილი იქნება ნაკადების აღრიცხვის პოლიტიკები. შაბლონის ფარგლებში ადმინისტრატორს უნდა შეეძლოს მოქნილად განსაზღვროს ტრაფიკის ის პარამეტრები, რომელთა მიხედვითაც განხორციელდება ნაკადის დადგენა და შემდგომი აღრიცხვა:
	1. მარშრუტიზატორის ადმინისტრატორს უნდა შეეძლოს მოთხოვნისამებრ მოქნილი შაბლონების განსაზღვრა. შაბლონის ფარგლებში უნდა აღიწეროს თავსართების ველები და პარამეტრები რის მიხედვითაც განხორციელდება ტრაფიკის ნაკადის დადგენა. შესაძლებელი უნდა იყოს ისეთი ველებისა და პარამეტრების აღწერა როგორებიცაა:
		1. IPv4 - IP ToS, IP Protocol, IP Source Address, IP Destination Address, Transport Source Port, Transport Destination Port, Interface Input, Interface Output
		2. IPv6 - Traffic Class, Protocol, IP Source Address, IP Destination Address, Transport Source Port, Transport Destination Port, Interface Input, Interface Output
		3. IP Autonomous System - Routing Source AS, Routing Destination AS, IPv4 Next Hop Address, IPv6 Next-hop Address
		4. შაბლონის ფარგლებში შესაძლებელი უნდა იყოს, მე-6 პუნქტში, აღწერილი ტექნოლოგიის გამოყენება, რათა მარშრუტიზატორმა განახორციელოს აპლიკაციის ამოცნობა. აპლიკაციის სახელის გამოყენება შესაძლებელი უნდა იყოს ნაკადის დადგენისათვის.
	2. მარშრუტიზატორის ადმინისტრატორს შაბლონის ფარგლებში უნდა შეეძლოს მოქნილად განსაზღვროს, თუ რა სახის ინფორმაციის აღრიცხვა სურს დადგენილი ტრაფიკის ნაკადის ფარგლებში. შესაძლებელი უნდა იყოს შემდეგი სახის პარამეტრების აღრიცხვა:
		1. IPv4 - Routing Source AS, Routing Destination AS, Next Hop Address, TCP Flags, Interface Input, Interface Output, Counter Bytes, Counter Packets, Time Stamp First, Time Stamp Last
		2. IPv6 - Routing Source AS, Routing Destination AS, Next-hop Address, TCP Flags, Interface Input, Interface Output, Counter Bytes, Counter Packets, Time Stamp First, Time Stamp Last
		3. შაბლონის ფარგლებში შესაძლებელი უნდა იყოს, მე-7 პუნქტში, აღწერილი ტექნოლოგიის გამოყენება, რათა მარშრუტიზატორმა განახორციელოს აპლიკაციის წარმადობის აღრიცხვა. აპლიკაციების ფარგლებში შესაძლებელი უნდა იყოს ისეთი პარამეტრების აღრიცხვა როგორებიცაა:
			1. Application Usage
			2. HTTP Hostname
			3. HTTP URI
			4. TCP Application – response time, transaction time, network delay, and application delay
			5. RTP – jitter, packet loss, per-stream byte count and packet count
	3. ტრაფიკის ნაკადების აღრიცხვის განხორციელება შესაძლებელი უნდა იყოს, როგორც ინტერფეისზე შემომავალი ასევე ინტერფეისიდან გამავალი ნაკადების
	4. ტრაფიკის ნაკადების შესახებ შეგროვილი სტატისტიკის გადაცემა შესაძლებელი უნდა იყოს ქსელის ნაკადების მონიტორინგის სერვერზე
14. მარშრუტიზატორს უნდა გააჩნდეს syslog, NetFlow, SNMP, RMON და IPFIX ოქმების მხარდაჭერა;
15. მარშრუტიზატორს უნდა გააჩნდეს შემდეგი მარშრუტიზაციის ოქმების მხარდაჭერა: EIGRP, EIGRP IPv6, OSFP, OSFPv3, IS-IS, IS-IS IPV6, BGP, BGP Route Reflector, BGP IPv6, PBR;
16. მარშრუტიზატორს უნდა გააჩნდეს შემდეგი Multicast ოქმების მხარდაჭერა: PIM Dense Mode, PIM Sparse Mode, PIM Source Specific Multicast (SSM), Bidirectional PIM, HSRP aware PIM, PIMv2 for IPv6 და Multicast Listener Discovery (MLD) version 1 და MLDv2, MLDP, MLDP-Based MVPN, MoFRR, MVPN**,** IGMP;
17. მომავალში მხოლოდ პროგრამული ლიცენზიის დამატებით, მარშუტიზატორს უნდა გაუჩნდეს შემდეგი ოქმების მხარდაჭერა: L2TPv3, BFD, IPSLA, MPLS, VRF, LISP, Virtual Private LAN Services (VPLS), Ethernet over MPLS.
18. მარშრუტიზატორს უნდა გააჩნდეს შემდეგი VPN ტექნოლოგიების მხარდაჭერა უსაფრთხო კავშირის გასამართად: IPSec, IKEv1, IKEv2, DMVPN;
19. კვება AC ტიპის
20. ზომა არაუმეტეს 1RU
21. მარშრუტიზატორზე უნდა ვრცელდებოდეს მწარმოებლის ერთ წლიანი საგარანტიო მომსახურეობა, ტექნიკური მხარდაჭერა და პროგრამული უზრუნველყოფის განახლება.მოწყობილობის დაზიანების მიზეზის დადგენის შემდეგ, მოწყობილობა უნდა შეკეთდეს ან შეცვალოს შემდეგ სამუშაო დღეს.
22. **მომწოდებელმა უნდა წარმოადგინოს მწარმოებლის ავტორიზაციის ფორმა (Manufacturers Authorization Form)**