CyPerf Test Drive:
Quantifying Cloud Excellence
Testing That Replicates Your Network in Action

The focus of these labs is to provide users with hands-on experience on how to use CyPerf for a set of test scenarios that quantify the performance and security efficacy of a cloud-deployed network.

KEYSIGHT

Table of Contents

Overview — Quantifying Cloud Excellence Lab

Cloud excellence involves proficiently using, managing, and optimizing cloud resources to deliver value to businesses and end-users. As reliance on cloud services grows, measuring and quantifying excellence becomes crucial for optimal outcomes.

Keysight CyPerf provides organizations quantifiable insights to demonstrate their cloud deployments' effectiveness, security, and resilience.

The Test drive gives you:
Hands-on Experience: Engage with Keysight CyPerf's intuitive interface and features in a live environment
Expert Guidance: Our team of cloud professionals has put together step-by-step guidance on how to use the test drive
Strategic Insights: Post-test drive, receive a detailed report highlighting how CyPerf can enhance your cloud operations

Take advantage of this unique opportunity to harness the power of Keysight CyPerf and steer your cloud operations toward unbridled excellence. Join our test drive and pave the way for an adequate and exemplary cloud infrastructure.

Keysight CyPerf is the industry's first cloud-native software test solution that recreates every aspect of a realistic workload across a variety of physical and cloud environments to deliver unprecedented insights into end user experience, security posture, and performance bottlenecks of distributed, hybrid networks.

CyPerf delivers new heights in realism that comes from simultaneously generating both legitimate traffic mixes and malicious activities across a complex network of proxies,

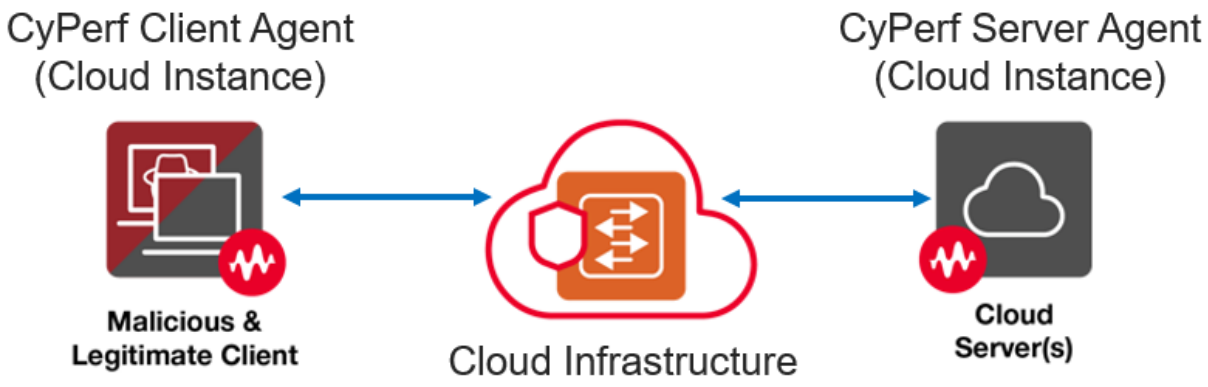
software-defined wide area networking (SD-WAN), Secure Access Service Edge (SASE), VPN tunnels, Transport Layer Security (TLS) inspection, elastic load balancers, and web applications firewalls (WAF). Combined with the unique ability to interleave applications and attacks to model user behavior and security breaches, CyPerf enables a holistic approach in replicating distributed customer deployment environments faster and with more fidelity than other solutions.

Lab Environment

A cloud-based setup with distributed, lightweight traffic agents that generate realistic application and malicious traffic to assess the performance and security efficacy of cloud-deployed network. The following elements are used in all of the labs described in this test drive.

The main two components of the Lab environment are as follows:

1. The **test tool**: Keysight's CyPerf emulating the malicious and legitimate traffic clients as well as traffic servers (all deployed as cloud instances)



2. The **device under test (DUT)**: the test traffic will run over the cloud infrastructure between the emulate clients cloud instances and emulated server cloud instances. There is no particular cloud-based network device in these labs, however other test topologies can be easily built by using a plethora of such cloud-based network devices (e.g., Next Generation Firewalls, Application/Elastic Load Balancers, Web Application Firewalls, Secure Web Gateways etc.) to quantified the performance and security of such devices under realistic traffic conditions.

The main components of Keysight's CyPerf (test tool) are as follows:

1. **Test Controller**: web-based UI for configuring and running tests, viewing real-time statistics, and reviewing results.

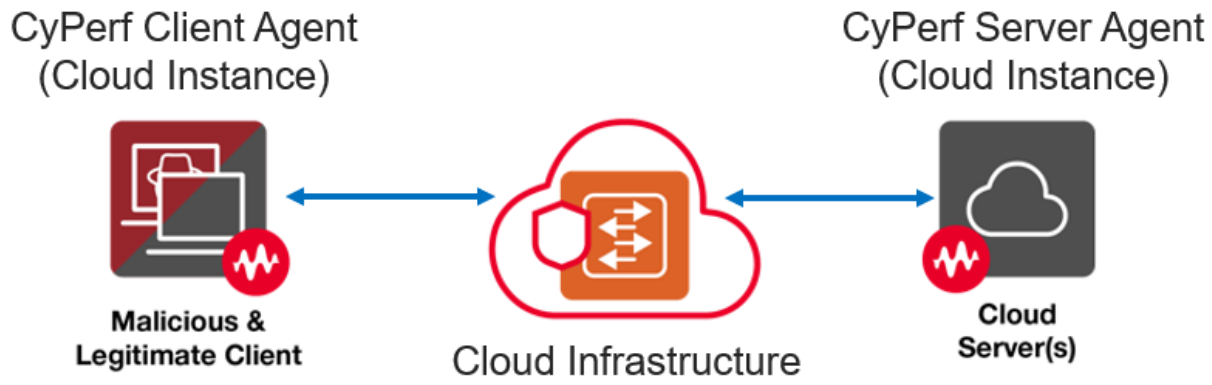
- The CyPerf Controller is deployed in the cloud and publicly available to users executing this lab as per the instructions in the video at (we recommend downloading it first for a smooth viewing experience):

https://github.com/Keysight/cyperf/blob/main/CyPerfTestDrive/Quantifying_Cloud_Excellence_Lab/CyPerf%20Cloud%20Excellence%20Test%20Drive%20Intro.mp4

2. **Traffic Agents**: software agents generating test traffic.
   - The CyPerf traffic agents (clients and servers) are deployed in the cloud to generate legitimate and malicious traffic going through the cloud infrastructure.

Setup

The following is the high-level diagram of the setup that is used in this lab:



The setup for all the labs consists of the following:

1. Traffic Agents:
   - **Client**: One CyPerf traffic agent acting as a client deployed as a public cloud instance.
   - **Server**: One CyPerf traffic agent acting as server behind also deployed as a public cloud instance.

Important

CyPerf traffic agents (clients and servers) can be virtually deployed in any Region/Zone, across a variety of public clouds (for example, Microsoft Azure, Amazon Web Services, Google Cloud Platform) as well as on on-prem machines to emulate a large-scale distributed network to test the performance and security efficacy of such infrastructures. For more details, see the product datasheet.

Running tests and labs Setup

The labs in this test drive range from a very basic concurrent connections test to more complex tests such as realistic application traffic and attacks. With every lab we increase the complexity and observe the most relevant key performance indicators (i.e., KPIs) that are required when quantifying a network device or even end-to-end network.

Caution

CyPerf delivers elastically scaling traffic agents that can spawn and tear down dynamically during a test to validate auto-scale policies and enables customers to fine-tune the balance between user experience and security. For more examples and templates for deploying environments with multiple server agents in autoscaling groups, see the CyPerf's public GitHub repo at: https://github.com/Keysight/cyperf

Resources and Prerequisites

To run this lab, users only need access to a common web browser. Everything will be run from a web interface.

All the resources for these labs can be found at the following location: https://github.com/Keysight/cyperf/tree/main/CyPerfTestDrive/Quantifying_Cloud_Excellence_Lab

This includes the following:

- Configuration files: each lab will start from a configuration file that is already pre-loaded into the controller (but also available for later reference at above GitHub location).
- Lab's document (this document).
- Intro video (CyPerf Cloud Excellence Test Drive Intro.mp4): a quick video that guides users on how to spin up and manage the test drive environment.
- Cloud Formation templates
  - Using the Cloud Formation templates found at the preceding location, users can deploy a similar setup with the one from this lab in their own cloud account.
- Terraform script deploys the same environment as the preceding Cloud Formation templates, through a single, aggregated Terraform script.

Looking for more resources? We offer a broad range of additional resources like deployment templates (for major public clouds), associated instructions and REST API wrappers at the following GitHub repository: https://github.com/Keysight/cyperf

Lab 1: Concurrent Connections with HTTP Traffic

Description

In this lab, we will observe the effect of having several concurrent connections open and transmitting data in our setup.

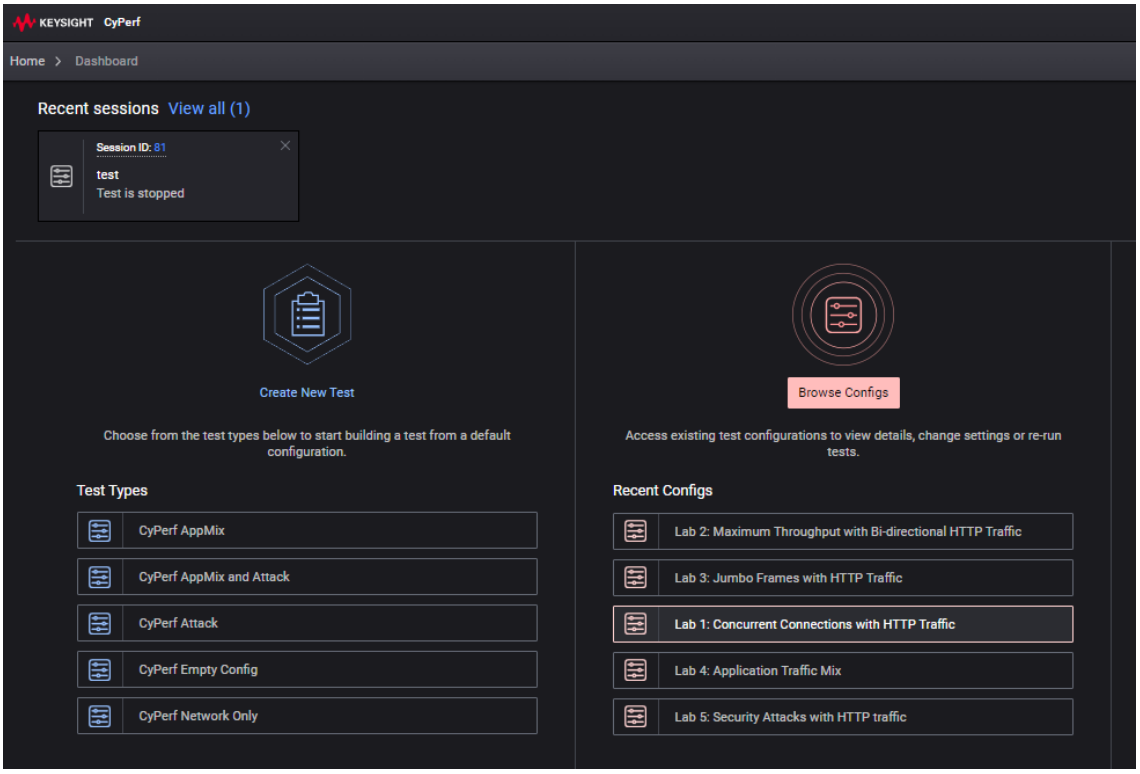
This test will try to reach and maintain 48,000 concurrent connections. Every group of connections will transmit the configured data commands and then close. The test will rotate through many connections such that the concurrent connections will be as close as possible to 48,000 at any given point in time during the test.

This lab targets the memory and CPU resources of the system since we will try to reach and maintain 48,000 concurrent connections (once a connection is closed a new one is immediately opened to maintain the configured concurrency goal). It takes time and resources to open such a large number of connections. If there were any DUTs or other devices in the network (e.g. any Layer 4 and above - NGFW, ALB, ELB, WAF etc), this lab will test resources for such devices as well. This lab uses HTTP POST and GET methods to transmit and receive data between the client and server agents.

Config

Load the config “Lab 1: Concurrent Connections with HTTP Traffic”:

- On the controller UI landing page (after login), in the Recent Configs area of the Browse Configs section click on the config “Lab 1: Concurrent Connections with HTTP Traffic”.



As soon as the config loads, the page will automatically show the test overview.

Observe the following for this config:

- The **Objectives & Timeline** section has been configured to try to achieve and maintain 48,000 connections over a duration of 300 seconds.
- The **Application Profile** tab displays an already configured AppMix with an HTTP application. It has been configured with HTTP GET and POST commands (click on the edit pencil button  to see the definition of the HTTP application).

Click on ‘START TEST’ to run the test.

Result KPIs

While running the test and after the test has completed, please observe the following important KPIs:

- **Summary** view - Application Simulated Users – Concurrent Connections

  This graph will highlight how many concurrent connections are achieved during the test, as well as how many concurrent simulated users.

- **Summary** view - Application connection rate

  This metric shows the number of connections initiated/succeeded/failed per second. Here we observe that our setup is opening and closing a fairly high number of connections continuously trying to achieve and sustain 48,000 connections at any point in time. Opening and keeping all these connections alive while transmitting data through them will stress the memory and CPU resources of our system.

- **Summary** view - Application successes/failures

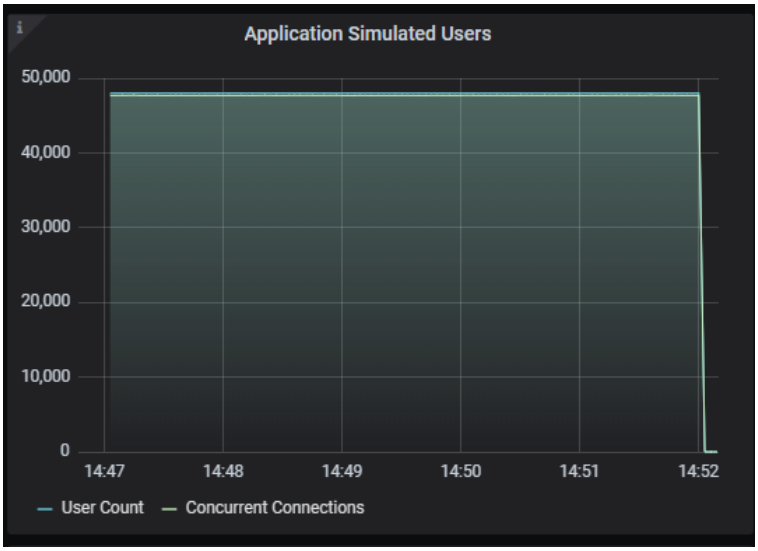
  This metric shows the number of application transactions initiated/succeeded/failed.

- **Agent Traffic Statistics** view - TX/RX Throughput for L23 and Apps

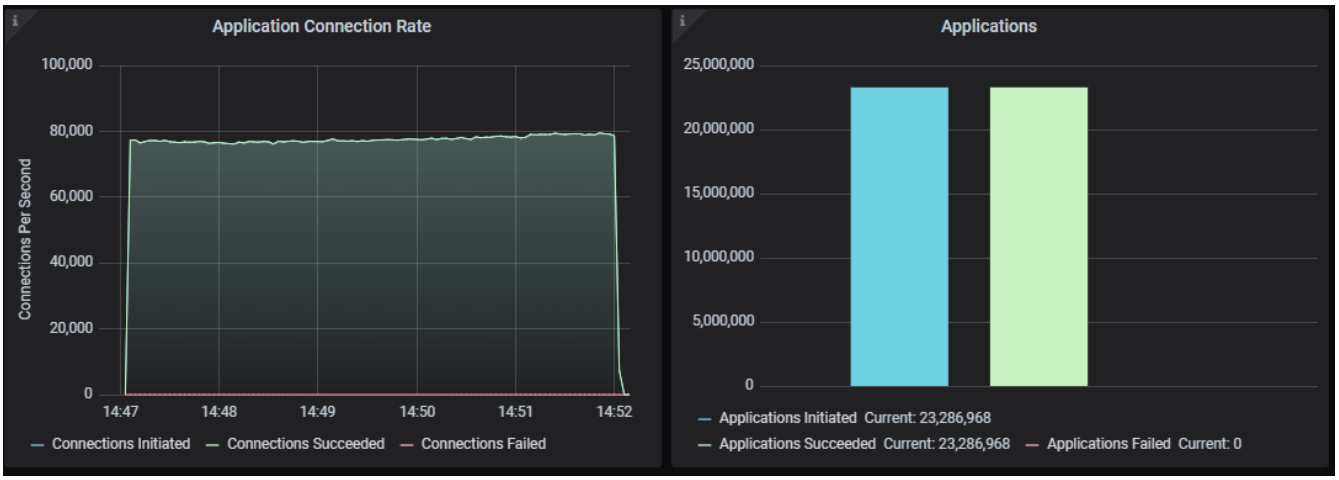
  We compare the TX vs RX throughput and observe if there are any drops while the connections per second ramped up and also while they were kept alive.

- **Agent Resource Metrics** view

  This metric shows us if the test agents are operating at full capacity and if so, one needs to increase the limits of the underlying testing infrastructure.

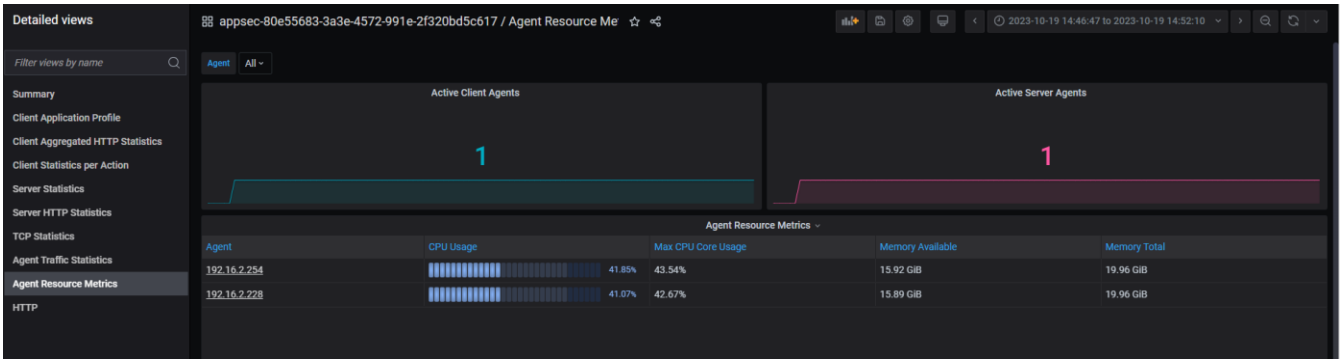


Summary **view -** Simulated Users & Concurrent Connections



Summary **view -** Application connection rate & application successes/failures

Agent Traffic Statistics view - Throughput TX/RX



Agent Resource Metrics view test agent resources

Conclusion

We observe the effect of sustaining 48,000 connections on the CPU and memory resources of our system. This helps us analyze if we need to increase the capacity of our system to be able to handle the expected load of the network. This test also helps us understand if any of the middle devices in our network need a capacity upgrade (in case we add an network device in the topolgy).

Lab 2: Maximum Throughput with Bi-directional HTTP Traffic

Description

In this lab, we will test the maximum throughput capacity of the network.

Config

Load the config “Lab 2: Maximum Throughput with Bi-directional HTTP Traffic”.

- On the controller UI landing page (click on the **Home** button in the upper left hand side of the screen), in the Recent Configs area of the Browse Configs section click on the config “Lab 2: Maximum Throughput with Bi-directional HTTP Traffic”

As soon as the config loads, the page will automatically show the test overview.

Observe the following for this new config:

- The **Objectives & Timeline** section has been configured to try and achieve 10 Gbps of throughput over a duration of 300 seconds.
- The **Application Profile** tab displays an already configured AppMix with an HTTP application. It has been configured with HTTP GET and POST commands (click on the edit pencil button  to see the definition of the HTTP application).

Click on ‘START TEST’ to run the test.

Result KPIs

While running the test and after the test has completed, please observe the following important KPIs:

- **Summary** view - Throughput

  This metric shows the overall bi-directional application throughput (transmit + receive) for both client and server

- **Summary** view - Application connection rate

  This metric shows the number of connections initiated/succeeded/failed per second. Here we observe if our setup is able to achieve and sustain connections. Every connection that is opened and kept alive will stress the memory and CPU resources of our system.

- **Summary** view - Application successes/failures

  This metric shows the number of application iterations initiated/succeeded/failed.

- **Summary** view - Instantaneous latency

  This graph has several subgraphs showing the average connection latency and min/avg/max time to first byte and time to last byte.

- **Agent Traffic Statistics** view - TX/RX Throughput for L23 and Apps

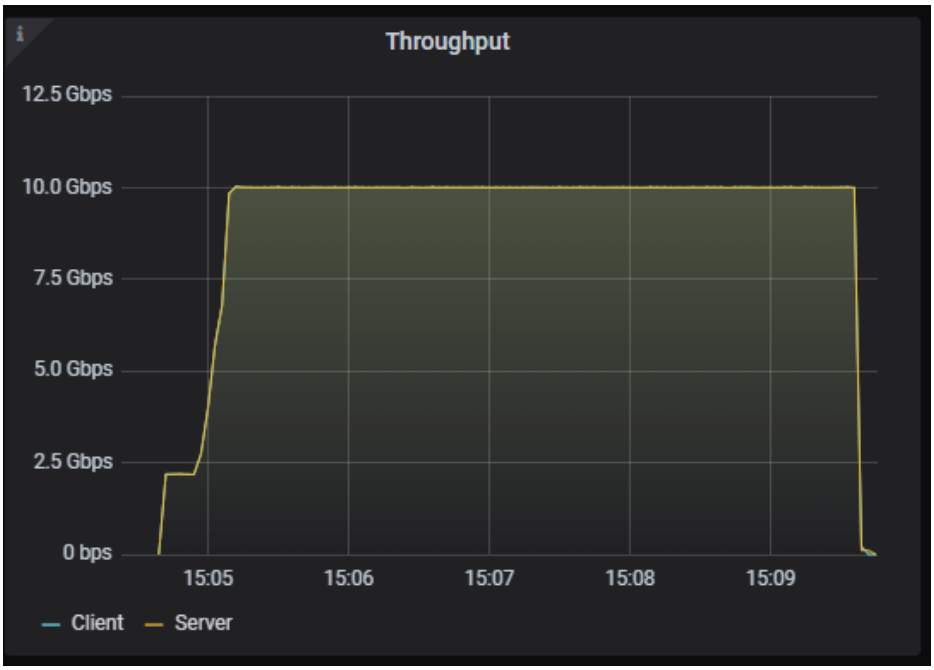
  We compare the TX vs RX throughput and observe if there are any drops while the test is running. This metric helps us characterize the performance of our network.

- **TCP Statistics** view - TCP client and server stats

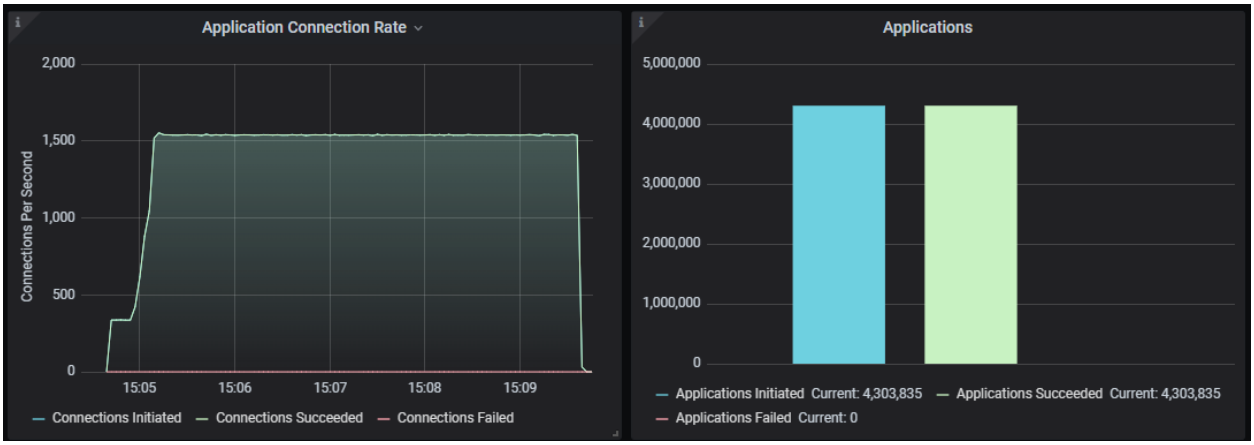
  These views include TCP handshake and data stats. A few important metrics to monitor are the retransmissions (both for data and SYN) which could indicate packet drops or other network issues.

- **Client/Server HTTP** view stats

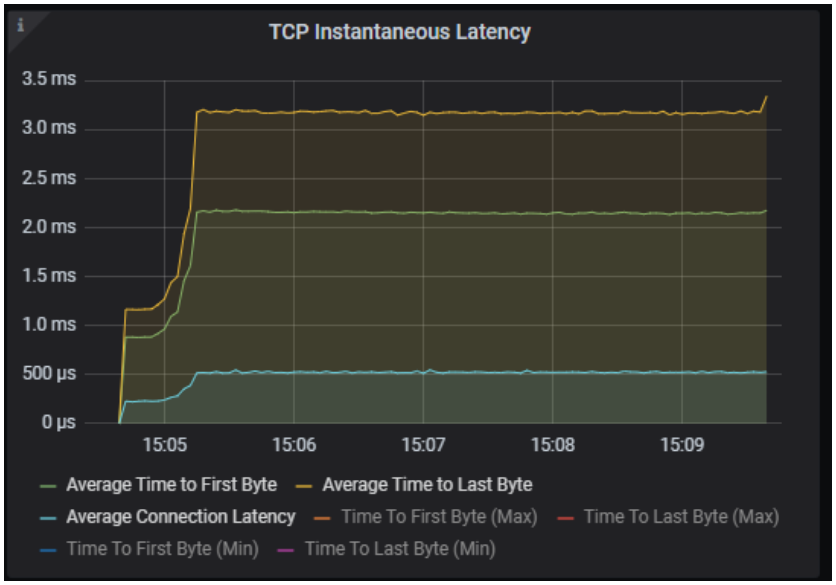
  In this metric, we observe HTTP bytes sent/received and other HTTP actions and stats.



Summary view - Throughput

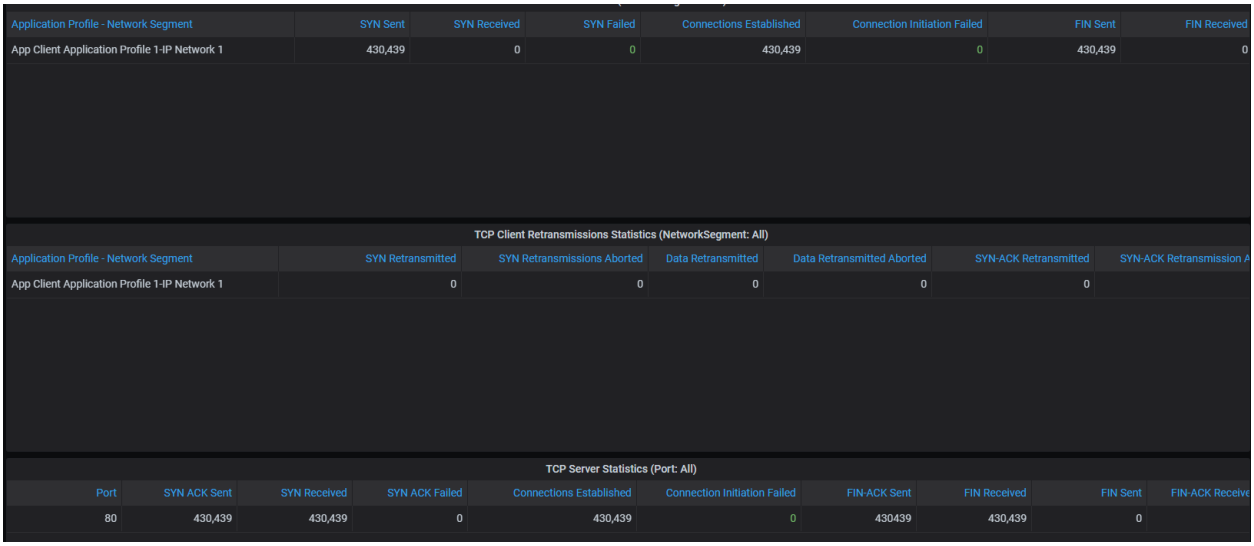


Summary view - Application connection rate & application successes/failures



Summary view – TCP Instantenious Latencies

Agent Traffic Statistics view - Throughput TX/RX



Application Profile - Network Segment	SYN Sent	SYN Received	SYN Failed	Connections Established	Connection Initiation Failed	FIN Sent	FIN Received
App Client Application Profile 1-IP Network 1	430,439	0	0	430,439	0	430,439	0

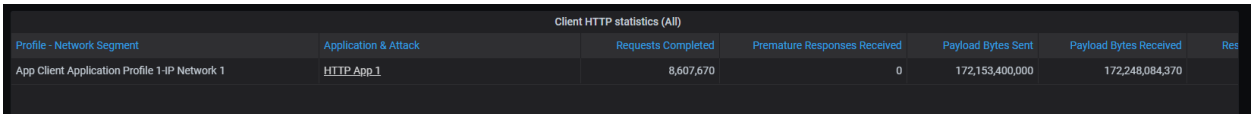
TCP Client Retransmissions Statistics (NetworkSegment: All)

Application Profile - Network Segment	SYN Retransmitted	SYN Retransmissions Aborted	Data Retransmitted	Data Retransmitted Aborted	SYN-ACK Retransmitted	SYN-ACK Retransmission A
App Client Application Profile 1-IP Network 1	0	0	0	0	0	

TCP Server Statistics (Port: All)

Port	SYN ACK Sent	SYN Received	SYN ACK Failed	Connections Established	Connection Initiation Failed	FIN-ACK Sent	FIN Received	FIN Sent	FIN-ACK Receive
80	430,439	430,439	0	430,439	0	430439	430,439	0	

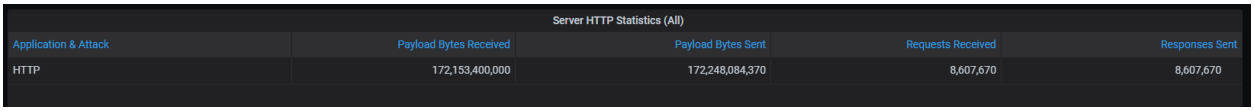
TCP Statistics view -TCP client/server stats



Client HTTP statistics (All)

Profile - Network Segment	Application & Attack	Requests Completed	Premature Responses Received	Payload Bytes Sent	Payload Bytes Received	Res
App Client Application Profile 1-IP Network 1	HTTP App 1	8,607,670	0	172,153,400,000	172,248,084,370	

Client Aggregated HTTP Statistics view - HTTP client stats



Server HTTP Statistics (All)

Application & Attack	Payload Bytes Received	Payload Bytes Sent	Requests Received	Responses Sent
HTTP	172,153,400,000	172,248,084,370	8,607,670	8,607,670

Client HTTP Statistics view HTTP server stats

Conclusion

In this test we observed the TX vs RX throughput and looked for drops or any performance degradation. This helps us characterize the true performance of our network. If there are other devices in the network, observe the drops, if any, at each of these hops.

Lab 3: Jumbo Frames Max Performance

Description

In this lab, we will observe the effect of transmitting jumbo frames in our setup.

Data will be sent using jumbo frames of 3058 bytes through the network.

If you have a DUT or other devices in the setup, set the MTU size to allow packet sizes of at least 3058 bytes on every device/hop through the network.

This lab will test the resiliency of network infrastructure to successfully allow and transmit larger than usual packet sizes.

Configuration

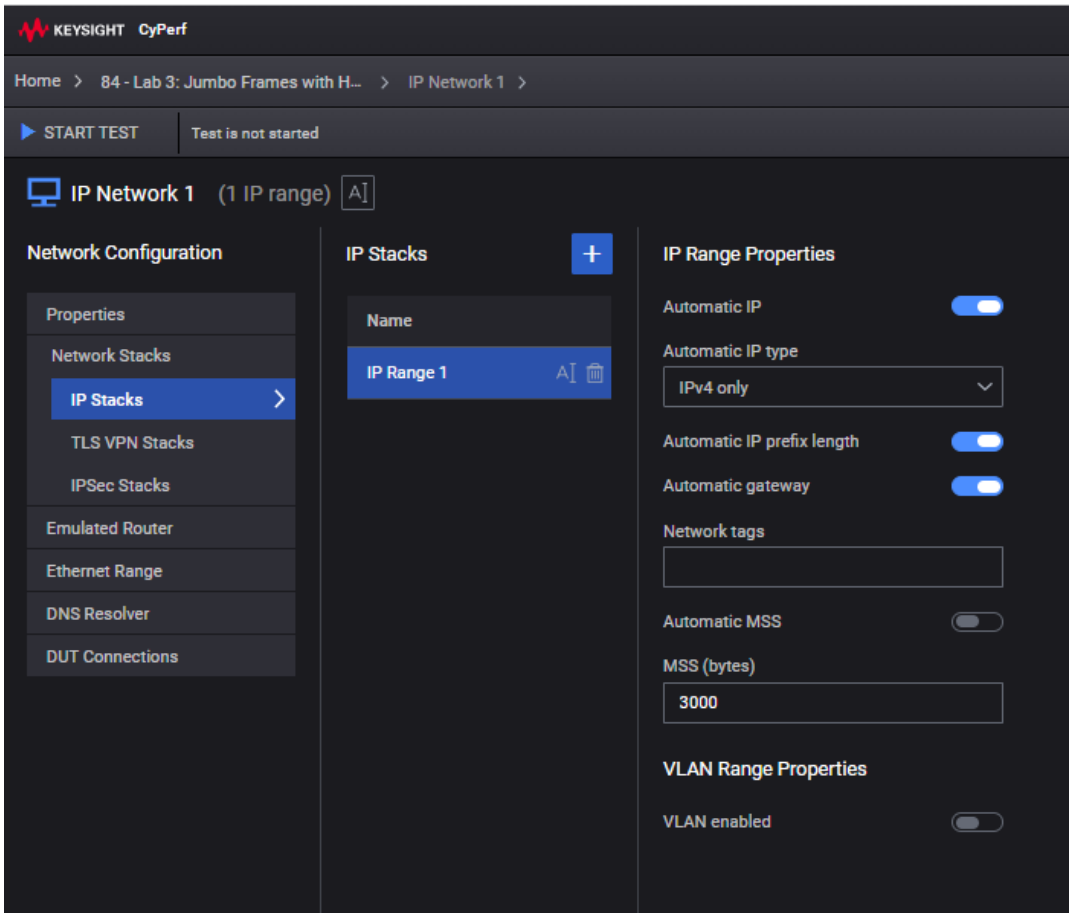
Load the config “Lab 3: Jumbo Frames Max Performance”.

- On the controller UI landing page (click on the **Home** button in the upper left hand side of the screen), in the Recent Configs area of the Browse Configs section click on the config “Lab 3: Jumbo Frames Max Performance”

As soon as the config loads, the page will automatically show the test overview.

Observe the following for this config:

- The **Objectives & Timeline** tab has been configured to try and achieve 1 Gbps throughput over duration of 300 seconds.
- The **Application Profile** tab displays an already configured AppMix with an HTTP application. It has been configured with an HTTP POST command.
    - Furthermore, if you click on “IP Network 1” block and then navigate to IP Stacks -> IP Range 1, you can notice that the MSS (maximum segment size) has been configured to 3000 Bytes. This would mean that the L2 frame size would be 3058B (including all the additional headers):



Same MSS value is configured on the server side, under “IP Network 1” block.

Click on ‘START TEST’ to run the test.

Result KPIs

While running the test and after the test has completed, please observe the following important KPIs:

- **Application Traffic Statistics** view - L23 TX/RX Throughput

  We compare the TX vs RX throughput and observe if there are any drops.

- **Application Traffic Statistics** view - L23 packets sent/received per second

  We compare the number of packets sent vs received by the client and server agents.

- **Application Traffic Statistics** view - L23 average packet sent/received size

  This metric shows that jumbo frames were transmitted by the client test agent, that these jumbo frames traversed the network, and were received by the server test agent. The actual values will be lower than the 3058 Bytes value mentioned

above since the TCP handshake packets (which are small packets) will be accounted for as well

- **Summary** view - Application connection rate

  This metric shows the number of connections initiated/succeeded/failed per second.

- **Summary** view - Application successes/failures

  This metric shows the number of applications initiated/succeeded/failed.

Application Traffic Statistics view - Throughput TX/RX

Application Traffic Statistics view **-** Packets per second & packet sizes

Summary view

Conclusion

We observe if our test setup can allow and transmit jumbo frames through every hop in the network. We can check if there are any packet drops, retransmissions or any other failures. It is important to mention that for the same target throughput, using jumbo frames would result in less packets per second which would typically save previous CPU cycles and the obvious benefit is the posibility of increased overal performance. However not all hops in an end-to-end path might support jumbo frames therefore these should be used with caution and tested before.

Lab 4: Application Traffic Mix

Description

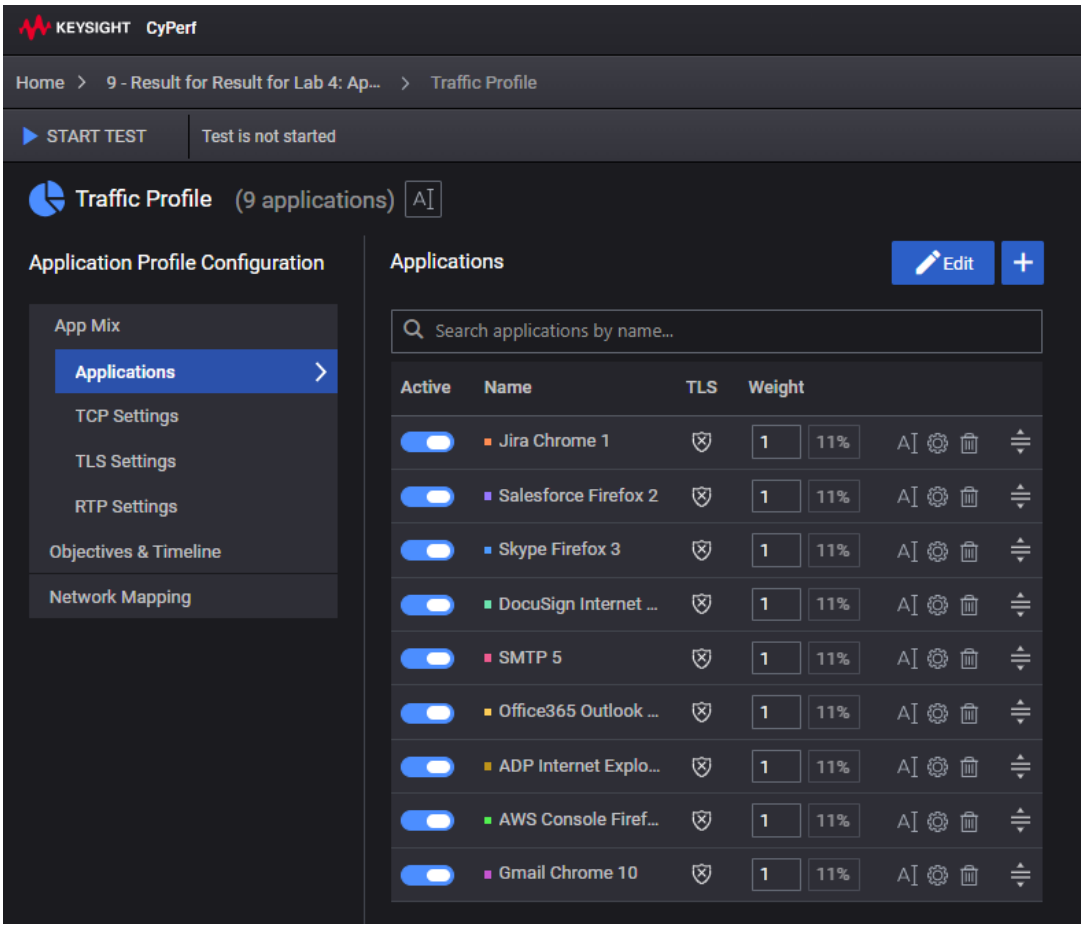
For this lab, we will use a realistic traffic profile resembling the typical applications traffic types in an enterprise network.

Many Network Equipment Manufacturers publish the performance figures of their devices in ideal scenarios with either generic, large packet HTTP traffic, or with application mixes that are geared, so that those devices render better performance.

Datasheets are a good starting point, but because each environment is unique, it is paramount to test with an application traffic profile resembling the closest production environment. This ensures that the test results are as relevant as possible to make informed decisions.

For this Lab, the emulated application traffic includes applications like Salesforce, SMTP, Jira, Skype, DocuSign, Office365 Outlook, ADP Internet Explorer 8, AWS Console Firefox, Gmail.

The complete list of the application traffic and the associated weights is emphasized next:



Config

Load the config “Lab 4: Application Traffic Mix”

- On the controller UI landing page (click on the Home button in the upper left hand side of the screen), in the Recent Configs area of the Browse Configs section click on the config “Lab 4: Application Traffic Mix”

As soon as the config loads, the page will automatically show the test overview.

Observe the following for this config:

- The **Objectives & Timeline** section has been configured to try to achieve 1 Gbps throughput over duration of 300 seconds
- The **Application Profile** tab displays an already configured AppMix with multiple realistic applications traffic.

Click on ‘START TEST’ to run the test.

Result KPIs

While running the test and after the test has completed, please observe the following important KPIs:

- **Summary** view - Throughput

  This metric shows the bidirectional throughout achieved for the client and server agents.

- **Summary** view - Application connection rate

  This metric shows the number of connections initiated/succeeded/failed per second.

- **Summary** view - Application successes/failures

  This metric shows the number of applications initiated/succeeded/failed.

- **Summary** view - Instantaneous latency

  This graph has several subgraphs showing the average connection latency and min/avg/max of the following parameters: time to first byte, time to last byte.

- **Agent Traffic Statistics** view - L23 TX/RX Throughput

  We compare the TX vs RX throughput and observe if there are any drops.

- **Client Application Profile** view - Detailed application stats per application

  These results show the number of connections and applications initiated/succeeded/failed, bytes sent/received.

- **Client Statistics per Action** view - Detailed application stats per application per action for client agents

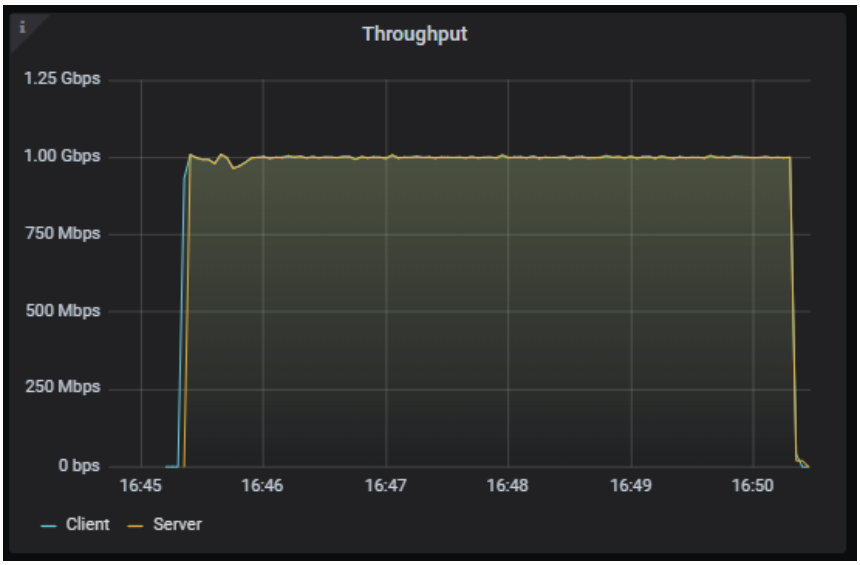
  These detailed results show the stats for every action performed by a particular application that was run during the test.

- **Server Statistics view- Detailed application stats per application server agents**

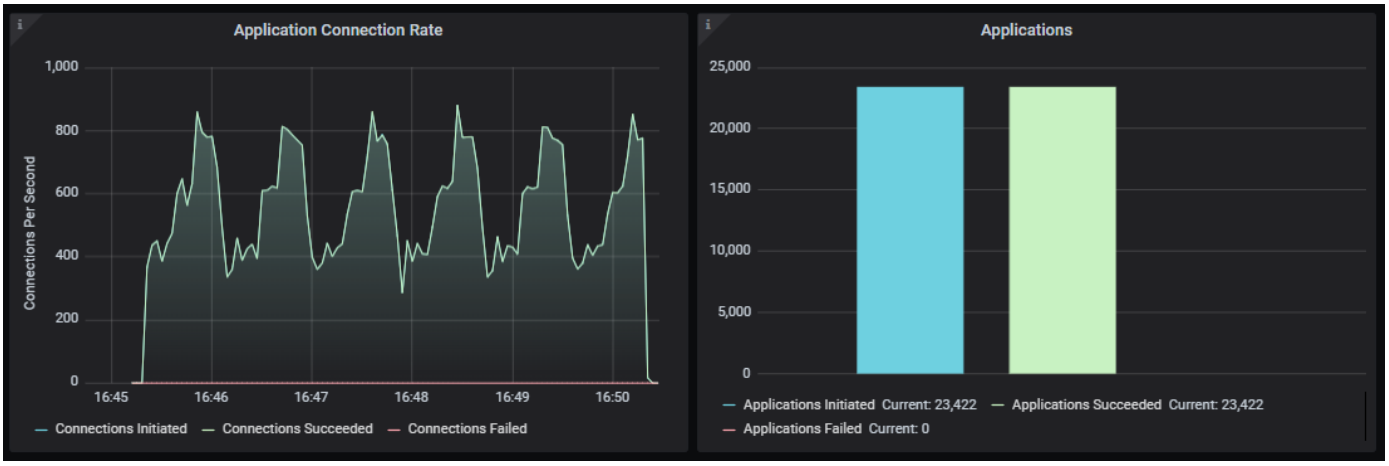
  These detailed results show server stats for every application that was run during the test

- Application stats drill down

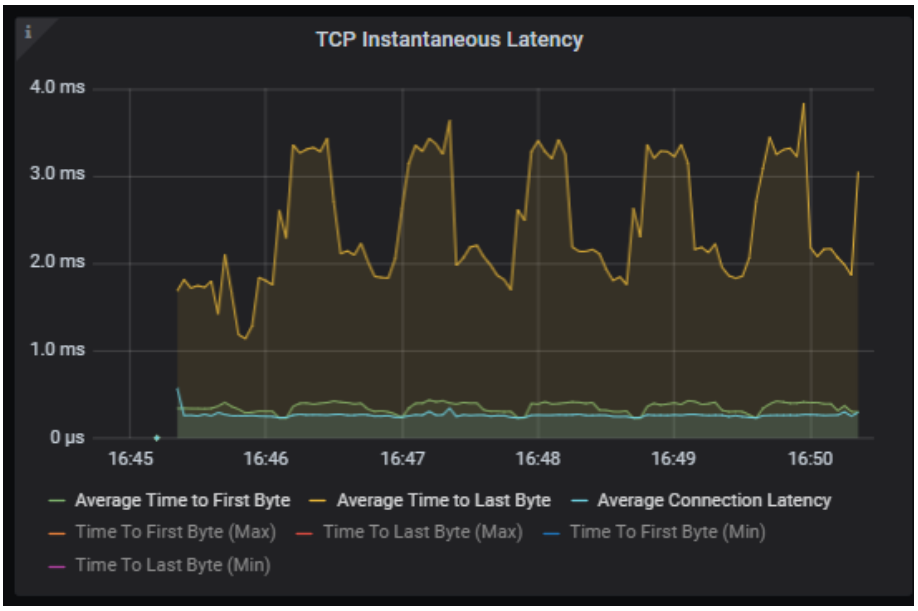
  These detailed results show the drilled-down stats for every application that was run as part of the test.



Summary view – Throughput



Summary view **-** Application connection rate & application successes/failures



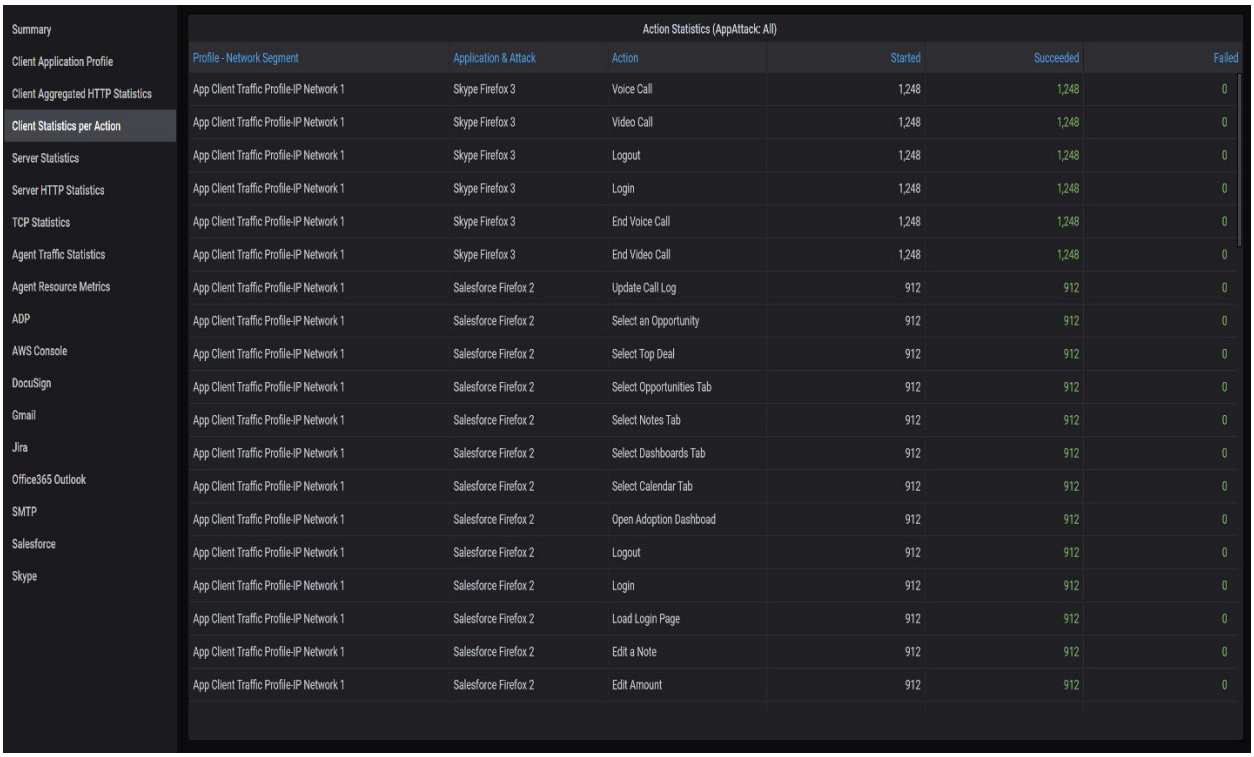
Summary view **–** TCP Instantanious Latency

Agent Traffic Statistics view - Throughput TX/RX

Application Statistics (App: All, NetworkSegment: All)

Application Profile - Network Segment	Application	Bytes Sent	Bytes Received	Connections Initiated	Connections Succeeded	Connections Failed	Connections Aborted	Applications
App Client Traffic Profile-IP Network 1	Skype Firefox 3	757,337,118	3,535,920,960	34,944	34,944	0	0	
App Client Traffic Profile-IP Network 1	Salesforce Firefox 2	1,206,674,929	3,174,951,984	3,648	3,648	0	0	
App Client Traffic Profile-IP Network 1	SMTP 5	3,768,730,608	6,593,776	8,336	8,336	0	0	
App Client Traffic Profile-IP Network 1	Office365 Outlook Microsoft Edge 7	1,468,956,840	2,880,698,688	21,312	21,312	0	0	
App Client Traffic Profile-IP Network 1	Jira Chrome 1	1,811,669,498	2,479,246,992	29,328	29,328	0	0	
App Client Traffic Profile-IP Network 1	Gmail Chrome 10	149,256,434	4,031,847,552	1,152	1,152	0	0	
App Client Traffic Profile-IP Network 1	DocuSign Internet Explorer 4	106,173,520	3,844,187,488	16,064	16,064	0	0	
App Client Traffic Profile-IP Network 1	AWS Console Firefox 9	1,188,425,109	3,033,463,288	37,560	37,560	0	0	
App Client Traffic Profile-IP Network 1	ADP Internet Explorer 8	117,049,360	3,944,399,576	19,344	19,344	0	0	

Client Application Profile view - Detailed application stats per application



Action Statistics (AppAttack: All)

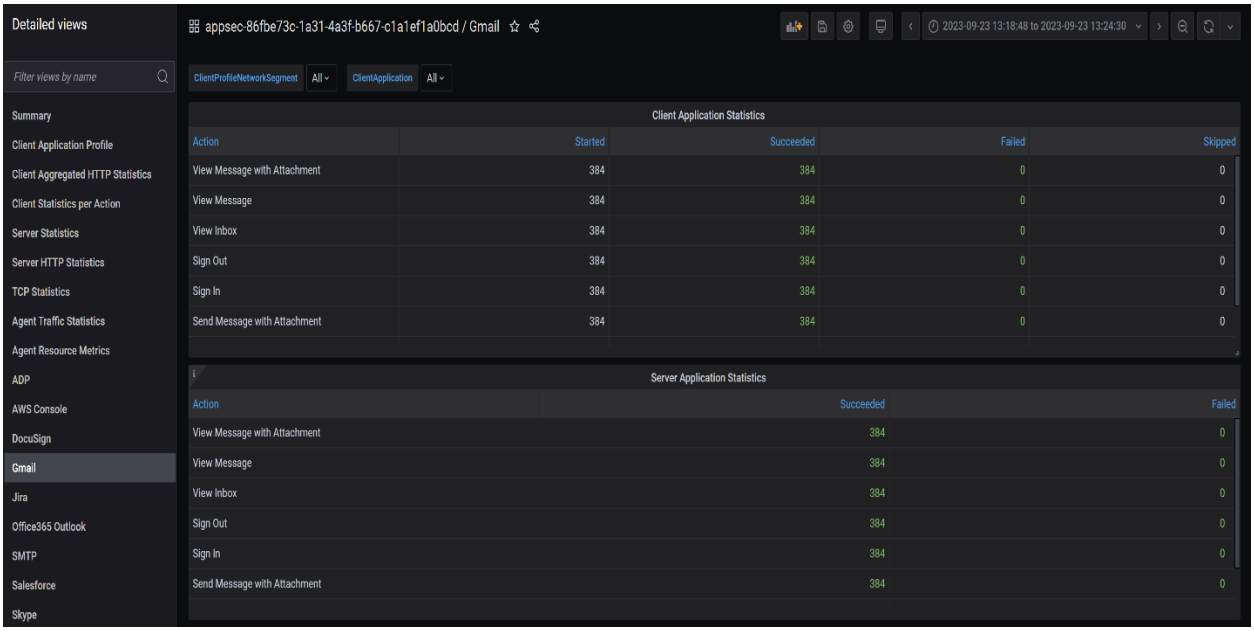
Profile - Network Segment	Application & Attack	Action	Started	Succeeded	Failed
App Client Traffic Profile-IP Network 1	Skype Firefox 3	Voice Call	1,248	1,248	0
App Client Traffic Profile-IP Network 1	Skype Firefox 3	Video Call	1,248	1,248	0
App Client Traffic Profile-IP Network 1	Skype Firefox 3	Logout	1,248	1,248	0
App Client Traffic Profile-IP Network 1	Skype Firefox 3	Login	1,248	1,248	0
App Client Traffic Profile-IP Network 1	Skype Firefox 3	End Voice Call	1,248	1,248	0
App Client Traffic Profile-IP Network 1	Skype Firefox 3	End Video Call	1,248	1,248	0
App Client Traffic Profile-IP Network 1	Salesforce Firefox 2	Update Call Log	912	912	0
App Client Traffic Profile-IP Network 1	Salesforce Firefox 2	Select an Opportunity	912	912	0
App Client Traffic Profile-IP Network 1	Salesforce Firefox 2	Select Top Deal	912	912	0
App Client Traffic Profile-IP Network 1	Salesforce Firefox 2	Select Opportunities Tab	912	912	0
App Client Traffic Profile-IP Network 1	Salesforce Firefox 2	Select Notes Tab	912	912	0
App Client Traffic Profile-IP Network 1	Salesforce Firefox 2	Select Dashboards Tab	912	912	0
App Client Traffic Profile-IP Network 1	Salesforce Firefox 2	Select Calendar Tab	912	912	0
App Client Traffic Profile-IP Network 1	Salesforce Firefox 2	Open Adoption Dashboad	912	912	0
App Client Traffic Profile-IP Network 1	Salesforce Firefox 2	Logout	912	912	0
App Client Traffic Profile-IP Network 1	Salesforce Firefox 2	Login	912	912	0
App Client Traffic Profile-IP Network 1	Salesforce Firefox 2	Load Login Page	912	912	0
App Client Traffic Profile-IP Network 1	Salesforce Firefox 2	Edit a Note	912	912	0
App Client Traffic Profile-IP Network 1	Salesforce Firefox 2	Edit Amount	912	912	0

Client Statistics per Action view - Detailed application stats per application per action for client agent

Application & Attack Statistics (All)

Application & Attack	Bytes Sent	Bytes Received	Connections Received	Strikes Allowed	Strikes Initiated
Skype	3,535,920,960	757,337,118	34,944	0	0
Salesforce	3,174,951,984	1,206,674,929	3,648	0	0
SMTP	6,593,776	3,768,730,608	8,336	0	0
Office365 Outlook	2,880,698,688	1,468,956,840	21,312	0	0
Jira	2,479,246,992	1,811,669,498	29,328	0	0
Gmail	4,031,847,552	149,256,434	1,152	0	0

Server Statistics view - Detailed application stats per application per action for server agent



Application stats drill down

Conclusion

In this test, the true performance of the infrastructure is characterized by using a more realistic application traffic mix. Similarly, if we were to use a cloud-based network device we would have been able (using the same test approach) to properly characterize the performance of the device under realistic traffic conditions which typically isn't provided in the vendors' datasheets.

Lab 5: Security Attacks with HTTP traffic

Description

In this lab, we will observe the effect of generating both legitimate traffic and security attacks over the tested infrastructure. These attacks are directed from both client to server and from server to client. The severity of these attacks varies from critical, high, medium, and low.

For this lab, the emulated attacks list includes encrypted attacks against applications that run by default over SSL, critical strikes which have a CVSS v3.0 score between 9 and 10, and DoS attacks.

Configuration

Load the config “Lab 5: Security Attacks with HTTP traffic”

- On the controller UI landing page (click on the **Home** button in the upper left hand side of the screen), in the Recent Configs area of the Browse Configs section click on the config “Lab 5: Security Attacks with HTTP traffic”

Observe:

- The **Application Profile** tab displays an already configured AppMix with an HTTP application. It has been configured with HTTP GET command.
- The **Attacks Profile** tab has been configured with ‘All Encrypted Attacks’ from client to server and ‘Firefox Browser Attacks’ from server to client.
- The **Objectives & Timeline** for the application profile tab has been configured to try and achieve 1 Gbps throughput over duration of 300 seconds.
- The **Objectives & Timeline** for the attacks profile tab has been configured to achieve 1 attack per second, 1 max concurrent attack, and 1 iteration.

Click on ‘START TEST’ to run the test.

Result KPIs

While running the test and after the test has completed, please observe the following important KPIs:

- **Summary** view - Client to server (C2S) and server to client (S2C) attacks initiated/allowed/blocked

This metric shows us the number of attacks that were initiated vs allowed to pass through the network vs blocked, thus displaying the vulnerability of the end-to-end network. If the system has a firewall as a DUT, this test can be used to test the effectiveness of the firewall as well.

- **Summary** view - Application successes/failures

  This metric shows the number of legitimate applications initiated/succeeded/failed.

- **All Encrypted Attacks** and **Firefox Browser Attacks** views - C2S and S2C attack statistics

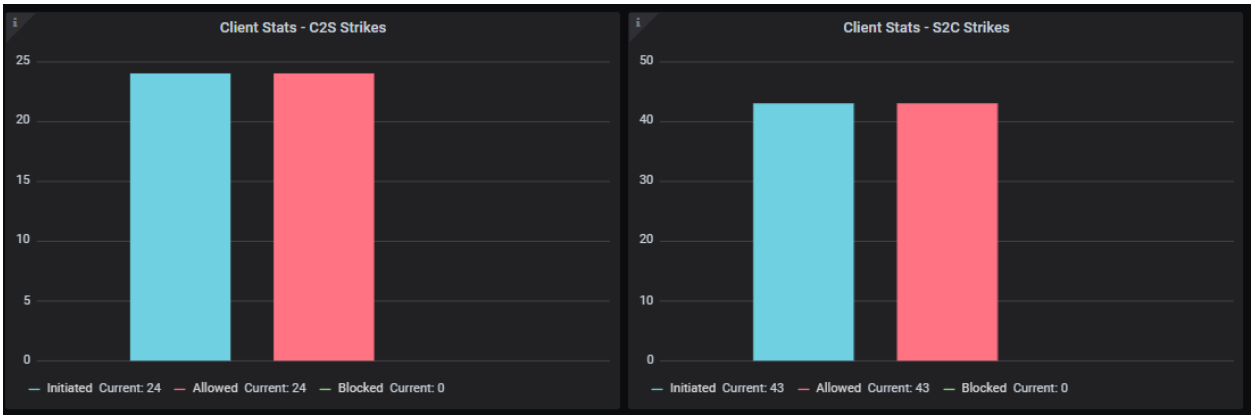
  These detailed result pages display additional info on every strike in every attack we have added in the attack profile in the test. This detailed drill-down helps you take a step further in analyzing what strikes were initiated/allowed/blocked, thus helping with troubleshooting the network for vulnerabilities in a detailed manner.

- **Summary** view - Instantaneous latency for application traffic and attacks

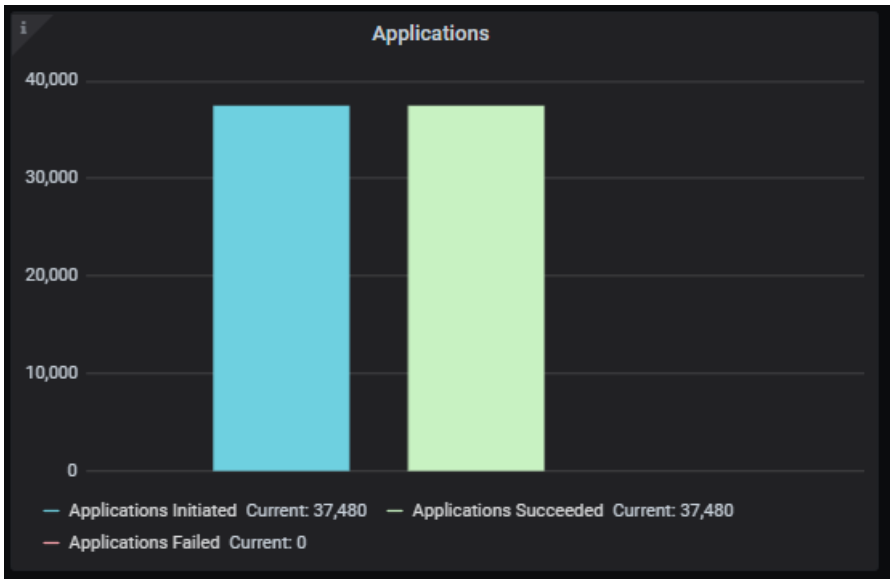
  This graph has several sub graphs showing the average connection latency and min/avg/max of the following parameters: time to first byte, time to last byte. This info is displayed via separate graphs for application traffic and attacks.

- **Agent Traffic Statistics** view - L23 TX/RX Throughput

  We compare the TX vs RX throughput and observe if there are any drops. We also compare the throughput performance of 3000-byte frame traffic vs the previous 1500-byte frames traffic.

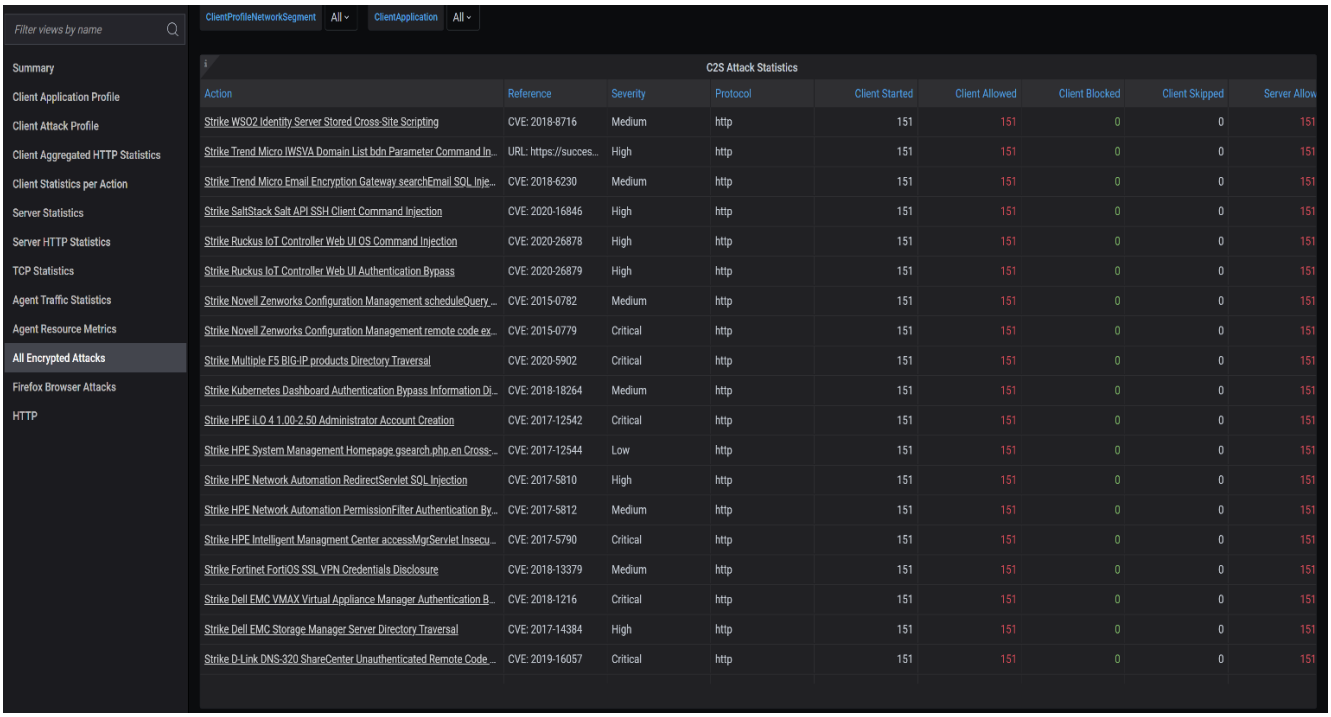


Summary view - Client to server (C2S) and server to client (S2C) attacks initiated/allowed/blocked



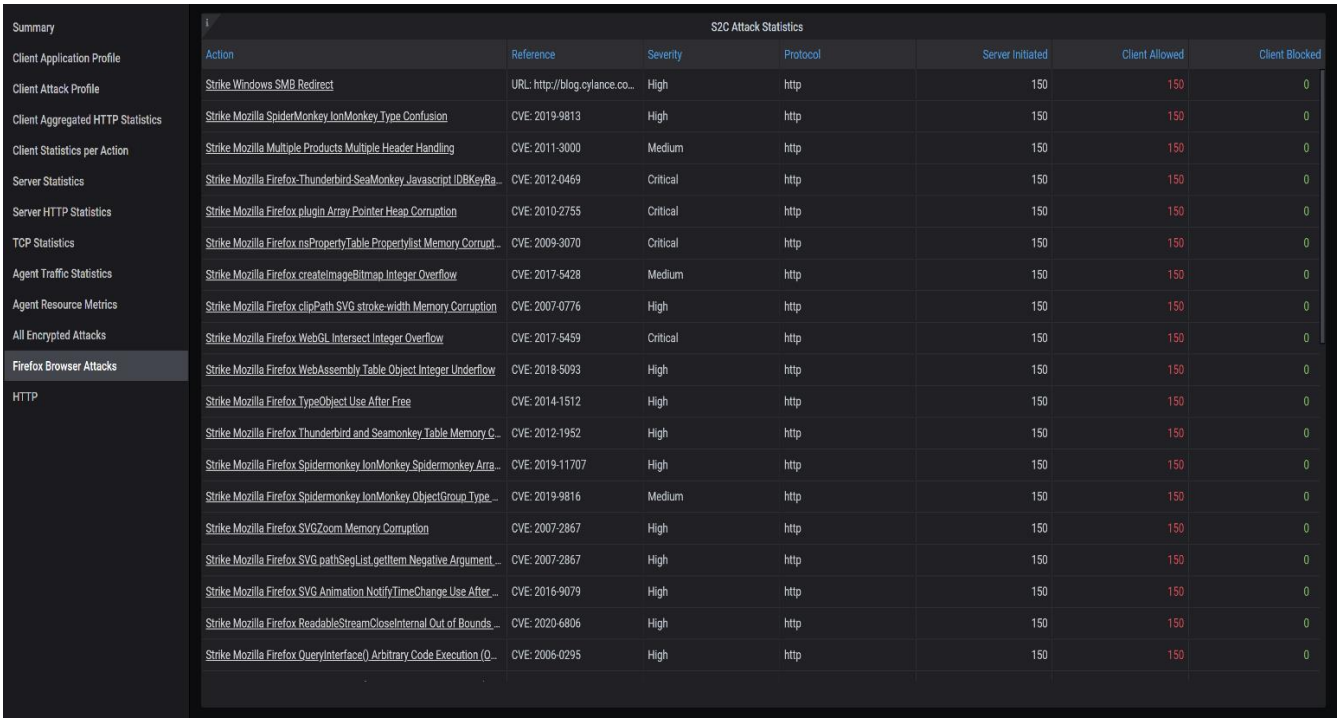
Summary view - Application successes/failures

Agent Traffic Statistics view - Throughput TX/RX



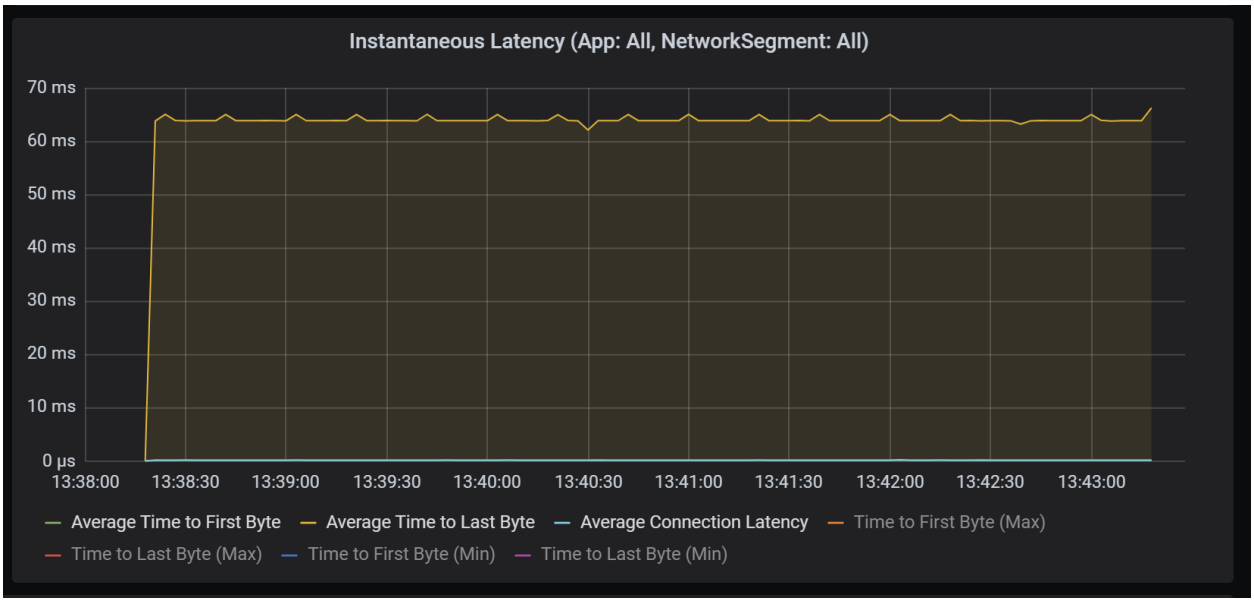
Action	Reference	Severity	Protocol	Client Started	Client Allowed	Client Blocked	Client Skipped	Server Allow
Strike WSO2 Identity Server Stored Cross-Site Scripting	CVE: 2018-8716	Medium	http	151	151	0	0	151
Strike Trend Micro IWSVA Domain List bdn Parameter Command In...	URL: https://succes...	High	http	151	151	0	0	151
Strike Trend Micro Email Encryption Gateway searchEmail SQL Inje...	CVE: 2018-6230	Medium	http	151	151	0	0	151
Strike SaltStack Salt API SSH Client Command Injection	CVE: 2020-16846	High	http	151	151	0	0	151
Strike Ruckus IoT Controller Web UI OS Command Injection	CVE: 2020-26878	High	http	151	151	0	0	151
Strike Ruckus IoT Controller Web UI Authentication Bypass	CVE: 2020-26879	High	http	151	151	0	0	151
Strike Novell Zenworks Configuration Management scheduleQuery ...	CVE: 2015-0782	Medium	http	151	151	0	0	151
Strike Novell Zenworks Configuration Management remote code ex...	CVE: 2015-0779	Critical	http	151	151	0	0	151
Strike Multiple F5 BIG-IP products Directory Traversal	CVE: 2020-5902	Critical	http	151	151	0	0	151
Strike Kubernetes Dashboard Authentication Bypass Information Di...	CVE: 2018-18264	Medium	http	151	151	0	0	151
Strike HPE iLO 4 1.00-2.50 Administrator Account Creation	CVE: 2017-12542	Critical	http	151	151	0	0	151
Strike HPE System Management Homepage gsearch.php.en Cross-...	CVE: 2017-12544	Low	http	151	151	0	0	151
Strike HPE Network Automation RedirectServlet SQL Injection	CVE: 2017-5810	High	http	151	151	0	0	151
Strike HPE Network Automation PermissionFilter Authentication By...	CVE: 2017-5812	Medium	http	151	151	0	0	151
Strike HPE Intelligent Managment Center accessMgrServlet Insecu...	CVE: 2017-5790	Critical	http	151	151	0	0	151
Strike Fortinet FortiOS SSL VPN Credentials Disclosure	CVE: 2018-13379	Medium	http	151	151	0	0	151
Strike Dell EMC VMAX Virtual Appliance Manager Authentication B...	CVE: 2018-1216	Critical	http	151	151	0	0	151
Strike Dell EMC Storage Manager Server Directory Traversal	CVE: 2017-14384	High	http	151	151	0	0	151
Strike D-Link DNS-320 ShareCenter Unauthenticated Remote Code ...	CVE: 2019-16057	Critical	http	151	151	0	0	151

All Encrypted Attacks view - C2S attack statistics

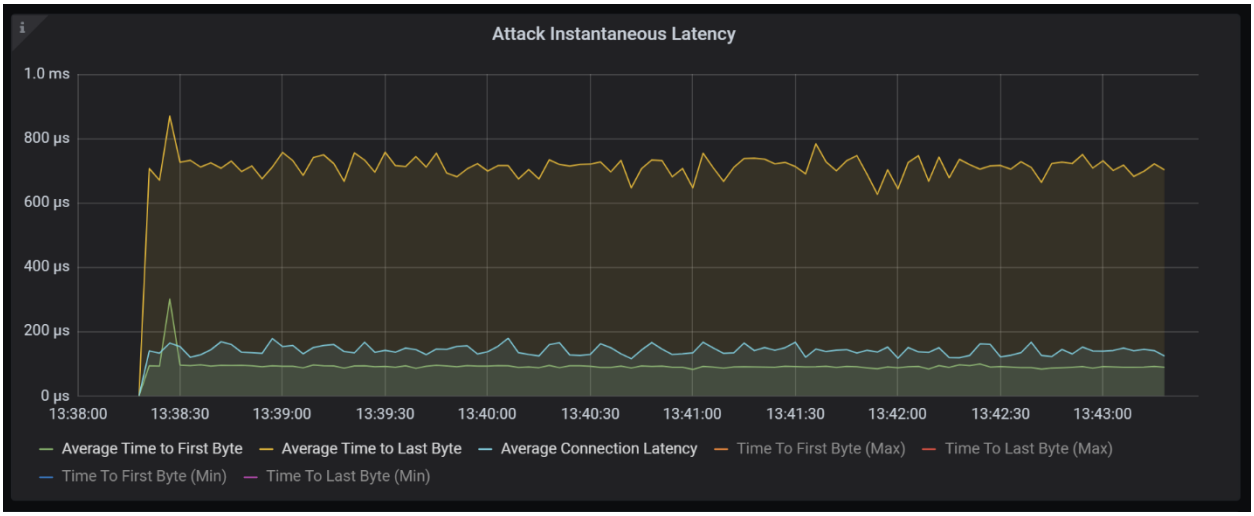


Action	Reference	Severity	Protocol	Server Initiated	Client Allowed	Client Blocked
Strike Windows SMB Redirect	URL: http://blog.cylance.co...	High	http	150	150	0
Strike Mozilla SpiderMonkey IonMonkey Type Confusion	CVE: 2019-9813	High	http	150	150	0
Strike Mozilla Multiple Products Multiple Header Handling	CVE: 2011-3000	Medium	http	150	150	0
Strike Mozilla Firefox-Thunderbird-SeaMonkey Javascript IDBKeyRa...	CVE: 2012-0469	Critical	http	150	150	0
Strike Mozilla Firefox plugin Array Pointer Heap Corruption	CVE: 2010-2755	Critical	http	150	150	0
Strike Mozilla Firefox nsPropertyTable Propertylist Memory Corrupt...	CVE: 2009-3070	Critical	http	150	150	0
Strike Mozilla Firefox createImageBitmap Integer Overflow	CVE: 2017-5428	Medium	http	150	150	0
Strike Mozilla Firefox clipPath SVG stroke-width Memory Corruption	CVE: 2007-0776	High	http	150	150	0
Strike Mozilla Firefox WebGL Intersect Integer Overflow	CVE: 2017-5459	Critical	http	150	150	0
Strike Mozilla Firefox WebAssembly Table Object Integer Underflow	CVE: 2018-5093	High	http	150	150	0
Strike Mozilla Firefox TypeObject Use After Free	CVE: 2014-1512	High	http	150	150	0
Strike Mozilla Firefox Thunderbird and Seamonkey Table Memory C...	CVE: 2012-1952	High	http	150	150	0
Strike Mozilla Firefox Spidermonkey IonMonkey Spidermonkey Arra...	CVE: 2019-11707	High	http	150	150	0
Strike Mozilla Firefox Spidermonkey IonMonkey ObjectGroup Type ...	CVE: 2019-9816	Medium	http	150	150	0
Strike Mozilla Firefox SVGZoom Memory Corruption	CVE: 2007-2867	High	http	150	150	0
Strike Mozilla Firefox SVG pathSegList.getItem Negative Argument ...	CVE: 2007-2867	High	http	150	150	0
Strike Mozilla Firefox SVG Animation NotifyTimeChange Use After ...	CVE: 2016-9079	High	http	150	150	0
Strike Mozilla Firefox ReadableStreamCloseInternal Out of Bounds ...	CVE: 2020-6806	High	http	150	150	0
Strike Mozilla Firefox QueryInterface() Arbitrary Code Execution (O...	CVE: 2006-0295	High	http	150	150	0

Firefox Browser Attacks view - S2C attack statistics



Client Application Profile view - Instantaneous latency for application traffic



Client Attack Profile view - Instantaneous latency for attacks

Conclusion

In this test, we observe the true performance of the infrastructure when running both legitimate traffic and security attacks.

This would be an important test for a network security device as it characterizes the security efficacy (e.g., block rate) of the device as well as the performance limits.