



HP NV Analytics

Software Version: 12.50

User Guide

Document Release Date: August 2015
Software Release Date: August 2015

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

© Copyright 2015 Hewlett-Packard Development Company, L.P.

Trademark Notices

Adobe® is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

Red Hat® is a registered trademark of Red Hat, Inc. in the United States and other countries.

VMware® is a registered trademark of VMware, Inc. in the United States and other countries.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to:
<https://softwaresupport.hp.com>.

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to <https://softwaresupport.hp.com> and click **Register**.

Support

Visit the HP Software Support Online web site at: <https://softwaresupport.hp.com>

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to: <https://softwaresupport.hp.com> and click **Register**.

To find more information about access levels, go to:
<https://softwaresupport.hp.com/web/softwaresupport/access-levels>.

HP Software Solutions & Integrations and Best Practices

Visit **HP Software Solutions Now** at <https://h20230.www2.hp.com/sc/solutions/index.jsp> to explore how the products in the HP Software catalog work together, exchange information, and solve business needs.

Visit the **Cross Portfolio Best Practices Library** at <https://hpln.hp.com/group/best-practices-hpsw> to access a wide variety of best practice documents and materials.

About this PDF Version of Online Help

This document is a PDF version of the online help. This PDF file is provided so you can easily print multiple topics from the help information or read the online help in PDF format. Because this content was originally created to be viewed as online help in a web browser, some topics may not be formatted properly. Some interactive topics may not be present in this PDF version. Those topics can be successfully printed from within the online help.

Contents

Chapter 1: NV Analytics	7
System Requirements	7
Install NV Analytics	7
Log Files	10
Chapter 2: Configure licensing	11
Licensing methods	11
Access the NV License Manager	12
Install a seat license	12
Set up your license server	12
Use floating licenses	13
Chapter 3: Analyzing Results	15
Settings	15
Analyzing a Network Virtualization Test Result File	17
Exporting NV Analytics Results	18
Viewing Reports	19
Response Time	20
Summaries	20
Client Network Server Breakdown	21
Parameters	21
General Analysis	22
Display Options	22
Subtransaction Parameters	23
Request/Response Details	23
Throughput	23
Endpoint Latencies	25
TCP/UDP Errors & Sessions	26
HTTP Analysis	27
Subtransaction Details	28
HTTP Parameters	29
HTTP Optimization	30
HTTP Resources and Responses	31
Resources Breakdown	32
HTTP Errors	32
HLS Errors	32
Response Summary	33
Secure Communication	33
Chapter 4: NV Analytics API	35
AnalysisEngines	35

Code sample	36
Extract Packet Lists	36
Code sample	37
AnalysisRequest	38
AnalysisSummary	41
AnalysisArtifact	42
Structure of an Analysis Report	43
Structure of the HTTP Waterfall Analysis Report	44
Structure of the Best Practices Analysis Report	46
Chapter 5: NV Analytics Protocols	48
Supported Protocols	48
Conversation Definition	48
Collecting Conversation Statistics	48
Classification of TCP, UDP, IP	49
Sub-Transaction Grouping	49
Understanding Protocol Association	49
Send Us Feedback	50

Chapter 1: NV Analytics

NV Analytics assists in pinpointing factors that negatively impact an application's performance across a network. NV Analytics performs an analysis based on packet list data, then displays the resulting data in informative reports that provide insight into an application's operation.

Analysis of HTTP, HTTPS and other protocols in waterfall diagrams provide a visual look into individual resource sizes and load times, enabling rapid analysis of transaction response times and the ability to quickly identify areas for optimization.

This section describes:

- **System Requirements:** Reviews the minimum host requirements for NV Analytics. See "[System Requirements](#)" below for more information.
- **Installing Software:** Provides step-by-step instructions on how to install NV Analytics. See "[Install NV Analytics](#)" below for more information.
- **Log Files:** Gives the location of the NV Analytics log files. See "[Log Files](#)" on page 10 for more information.

System Requirements

The minimum requirements for NV Analytics are as follows:

Processor	Quad core 2.5 GHz or stronger
Memory	4 GB RAM
Hard Disk	1 GB of free disk space
Desktop Operating System	<ul style="list-style-type: none">• Windows Server 2008 R2 SP1 (64 bit)• Windows 7 SP1 (32/64 bit)• Windows Server 2012 R2• Windows 8.1 (64 bit)
Browsers	<ul style="list-style-type: none">• Internet Explorer 9.0 and higher• Firefox• Chrome• Safari
Microsoft Office (for exporting reports)	<ul style="list-style-type: none">• Office 2007• Office 2010• Office 2013

Install NV Analytics

Note: NV Analytics reports are a part of Network Virtualization. If you have installed NV on your VuGen machine, you can view the NV Analytics reports directly in VuGen, without the need to install the standalone NV Analytics version.

If you are already using the standalone version of NV Analytics, you can install the updated 12.50 version, included with the NV installation files.

Software Prerequisites

Wireshark version 1.10.8 or later must be installed prior to installation of NV Analytics.

The following will be installed during installation if not present:

- Java Runtime Environment 8.0 update 45 (32-bit) will be installed if none of the following are present:
 - JRE 6.0 update 24 or later (32-bit)
 - JRE 7.0 (32-bit)
 - JRE 8.0 (32-bit)
- Microsoft .NET Framework 4.5.2 Full
- Microsoft Silverlight 5.1.30514

Installing Software

To install NV Analytics, run the Analytics setup file **Analytics_setup.exe** (as an administrator), click **Install**, and follow the on-line directions.

Upgrading NV Analytics: When you run the installation, the installer checks if there is a previous version of NV Analytics installed. Depending on the version you have installed, the installer does one of the following:

- Uninstalls the previous version and then continues the installation.
- Prompts you to first uninstall the previous version and then run the installation again.

After installing NV Analytics, you must reboot the computer.

After installation, access NV Analytics from the Windows Start menu: **All Programs > HP Software > HP Network Virtualization > NV Analytics > NV Analytics.**

In Windows 8.x or higher, you can access NV Analytics directly from the **Start** or **Apps** screen.

Silent Installation

Note: If Wireshark is not installed prior to installation of NV Analytics, installation will abort.

To silently install NV Analytics:

1. Copy the NV Analytics setup file from the installation package to a convenient location.
2. From the Windows Start menu, click **All Programs > Accessories**, right-click **Command Prompt**, and then click **Run as administrator**.
3. In the Command window, navigate to the location of the file copied in step 1, and enter the following command together with the required command line options:

```
Analytics_setup.exe /s /v"/qn command_line_options"
```


Command line options [* indicates a mandatory command line option]:

- **PORT=<port number>**
The port used to connect to NV Analytics.
If another Network Virtualization component is already installed, NV Analytics will use the same port.
- **ENABLE_REMOTE=TRUE | FALSE**
Opens the port in the firewall.
Default is TRUE.
- **DATA_FOLDER="<path to data folder>"**
The location where internal application data and user data is saved.
Default is <Common App Data folder>\HP\NV (C:\ProgramData\HP\NV in Win 7).
- **INSTALLDIR="<path to installation folder>"**
The location where the application files will be installed.
Default is C:\Program Files (x86)\HP\NV\
- **REBOOT_IF_NEED=TRUE | FALSE**
If a reboot is needed, automatically reboots the computer after installation completes.
Default is TRUE.

Silent Uninstallation

To silently uninstall NV Analytics:

1. Copy the file Analytics_setup.exe setup file from the installation package to a convenient location.
2. From the Windows Start menu, click **All Programs > Accessories**, right-click **Command Prompt**, and then click **Run as administrator**.
3. In the Command Prompt window, navigate to the location of the file copied in step 1, and enter the following command together with the required command line options:

```
Analytics_setup.exe /s /removeonly /v"/qn command_line_options"
```

Command line options [All the command line options are optional]:

- **REBOOT_IF_NEED=TRUE | FALSE**
If a reboot is needed, automatically reboots the computer after uninstall completes.
Default is TRUE.
- **FORCE_REBOOT=TRUE | FALSE**
Automatically reboots the computer after uninstall completes, whether or not a reboot is needed.
Default is FALSE.
- **DELETE_DATA=TRUE | FALSE**
Deletes all stored NV Analytics data.
Default is FALSE.

Installation log files

Installation logs are located under C:\HP Log. Log files are named as follows:

<Product name>_<date>_<time>.log

For example:

HP NV for Load Generator_6-4-2015_15-29-27.log

HP NV for Controller_6-4-2015_15-37-38.log

Log Files

HP Network Virtualization products' log files are located in the <installation directory>\logs, by default in \Program Files\HP\NV\logs or \Program Files (x86)\HP\NV\logs.

Chapter 2: Configure licensing

After installing HP Network Virtualization components, you need to install the NV Analytics licenses. An NV Analytics license is required for the NV Analytics analysis functionality that is integrated with VuGen, and for using the standalone version of NV Analytics.

NV Analytics licenses are managed from the NV License Manager.

Note: Licensing for general NV emulation functionality and access to the NV Global Library is managed by the LoadRunner/Performance Center licensing utility, according to Network Virtualization Vuser licenses.

What do you want to do?

- Learn about: ["Licensing methods" below](#)
- ["Access the NV License Manager" on the next page](#)
- ["Install a seat license" on the next page](#)
- ["Set up your license server" on the next page](#)
- ["Use floating licenses" on page 13](#)

Licensing methods

The following licensing methods are available for the Network Virtualization products:

Seat licenses

A seat license is created for a specific Network Virtualization product on a specific computer, and cannot be transferred to another computer.

Floating licenses

When using floating licenses, licenses are held by a license server and you check them out as necessary. To use floating licenses, you must have the HP AutoPass License Server 8.3 or later installed on your network.

When you are finished using the licenses, you return the licenses to the license server so that they can be used by other NV installations.

Trial licenses

NV Analytics has a thirty (30) day trial license. The trial license gives access to all product functionality.

NV Analytics reports in VuGen: The trial period begins the first time you use the NV Analytics functionality in VuGen.

NV Analytics: The trial period begins the first time you run an analysis.

Caution: If you install any Network Virtualization product on a virtual machine, do not clone the machine after the trial license has started.

Access the NV License Manager

You can access the NV License Manager in the following ways:

- From the Windows **Start** menu, select **All Programs > HP Software > HP Network Virtualization > NV License Manager**.
In Windows 8.x or higher, you can access NV License Manager from the **Start** or **Apps** screen.
- From a web browser, navigate to the following URL:


```
http://<hostname>:<port>/shunra/license/
```

For example:

```
http://198.51.100.24:8182/shunra/license/
```

Install a seat license

When using seat licenses, you must apply a license to every computer that uses NV Analytics.

1. Open the NV License Manager on the desired computer.
2. Click **Update license**.
3. Click **Update via: > File or key**.
4. Copy the **machine code** displayed below.
5. Click **HP license portal** to connect to the HP Licensing site.
6. Enter a valid license Entitlement Order Number (EON). You will be redirected to a page where you can enter the machine code that you previously copied, and generate a license file.
7. Enter the license key or click the folder icon  that appears to the right of the **License File** box, locate and upload the license file.
8. Click **Update**. The updated license details are displayed in the NV License Manager main page.

Set up your license server

When using the floating license method, licenses are held by a license server and you check them out as necessary. Network Virtualization works with HP AutoPass License Server to manage floating licenses.

1. Select a machine to host the license server. The license server must be accessible by all machines on which you will generate NV Analytics reports.
2. Install HP AutoPass License Server. The installation folder **autopass-8.3.zip** is located together with the Network Virtualization installation files.

The installation files are located in the following folders:

- LoadRunner: <LoadRunner installation DVD>\Additional Components\HP NV\
• Performance Center: <Performance Center installation
DVD>\AdditionalComponents\HPNV\
Unzip and run the appropriate setup file. For more information, refer to the AutoPass documentation, located in the same folder.

3. Connect to the [HP license portal](http://h30580.www3.hp.com/): <http://h30580.www3.hp.com/>, and enter a valid license Entitlement Order Number (EON). Follow the instructions on the HP site to obtain the licenses and install them on your license server.

Use floating licenses

Floating licenses are held by a license server and you check them out as necessary. A license is checked out for a specified number of days, at the end of which it is automatically returned to license server. You can return a license early, if desired.

What do you want to do?

- ["Check out a license" below](#)
- ["Return a license" below](#)

Check out a license

Note: The maximum number of days that a license can be checked out can be configured on the HP AutoPass License Server. Refer to the AutoPass documentation for details.

1. Open the NV License Manager on the machine that needs to check out a license. For details, see ["Access the NV License Manager" on the previous page](#).
2. Click **Update License**.
3. Choose **Update via: License server**.
4. In the **License server address** field, select the machine where the license server is installed. If the license server does not appear in the list, enter its address.
5. In the **License duration (days)** field, choose for how long to check out the license. By default, the maximum number of days a license can be checked out is 30.
6. Under **more settings**:
 - a. Configure the port for the AutoPass License Server. By default, 5814. The machine that checks out a license must have access to the license server.
 - b. Select **Use secured communication** to use secure communication between the NV License Manager and the license server.
7. Click **Checkout license**. The license is checked out from the license server.

Return a license

1. Open the NV License Manager on the machine that needs to return a license.
2. Click the **Update License** button.

3. Choose **Update via: License server**.
4. Click **Return license**. The license is returned to the license server.

Chapter 3: Analyzing Results

This section describes how to view and analyze the results generated by NV Analytics.

NV Analytics reports consist of data captured using a network virtualization packet list which is subsequently processed and displayed in intuitive reports. These reports are then examined to decipher problems that may exist.

The Analytics reports provide detailed data about the breakdown of each transaction. Statistics of each resource being uploaded or downloaded in a transaction are displayed in both tabular and graphic format. Precise performance data includes load times, component download analysis, response time breakdown and details of errors received. Performance optimization recommendations how to improve and optimize mobile and non-mobile transaction performance are provided.

This section includes:

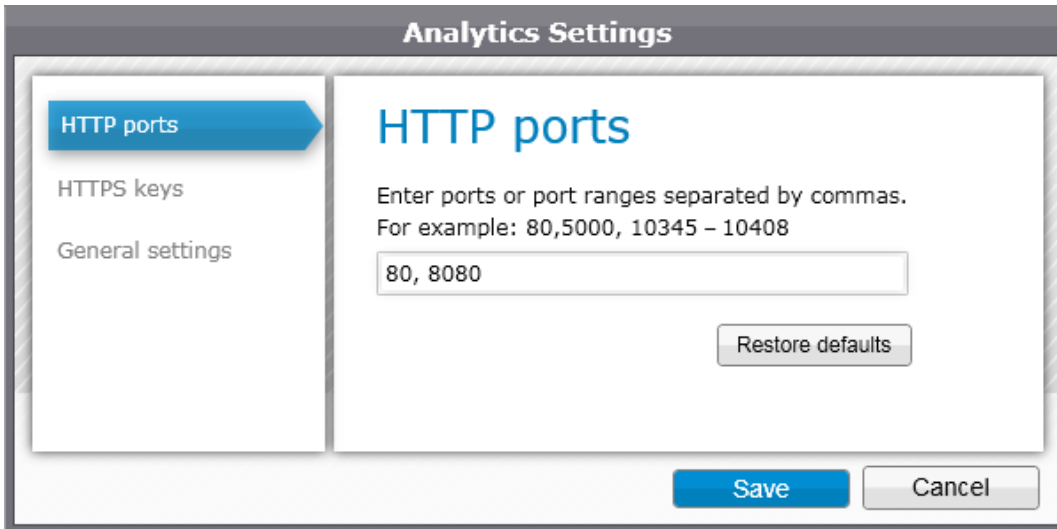
- [Settings](#) 15
- [Analyzing a Network Virtualization Test Result File](#) 17
- [Exporting NV Analytics Results](#) 18
- [Viewing Reports](#) 19
- [Secure Communication](#) 33

Settings

To configure Analytics settings, click the Settings icon (wrench) on the Welcome page. The settings are also available from the main page.

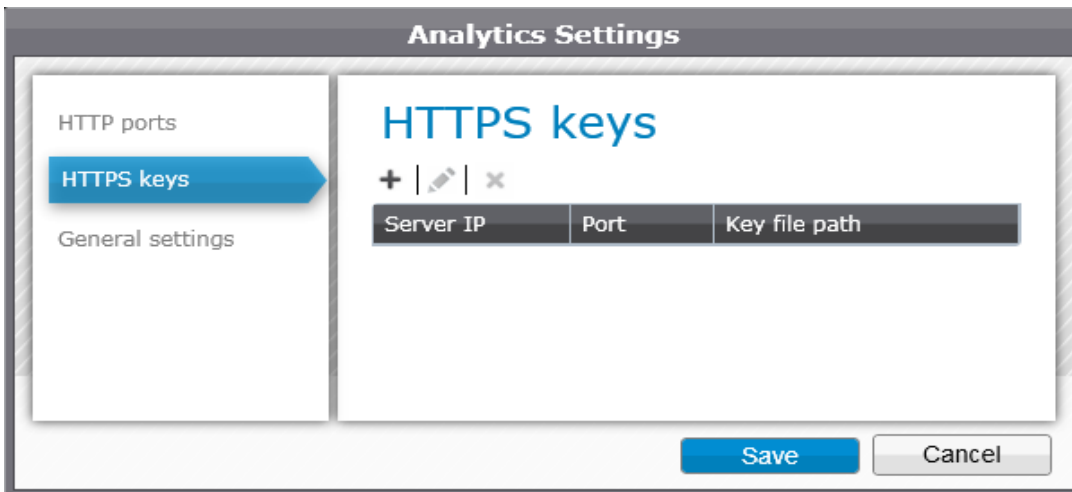
HTTP Port Settings

To modify the default ports, select the HTTP ports tab and add one or more ports or range, each separated by a comma. Add the ports that your application under test is using. HTTP/S analysis will run on the specified ports.



HTTPS Keys

To enable analysis of secure data, enter the HTTPS key.

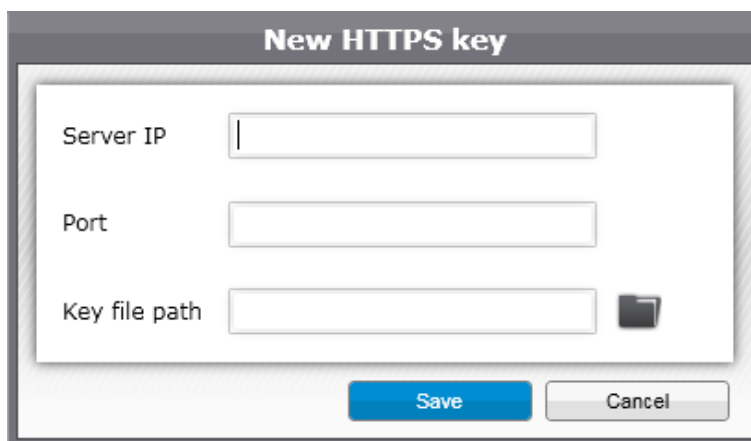


To add an HTTPS Key:

1. Click the "+" sign, and in the New HTTPS Key window, enter the required information.
2. To edit the information, click the pencil icon; to delete the Key click the "x".

The IP, port and private key of the application server, or of the debugging proxy if installed, should

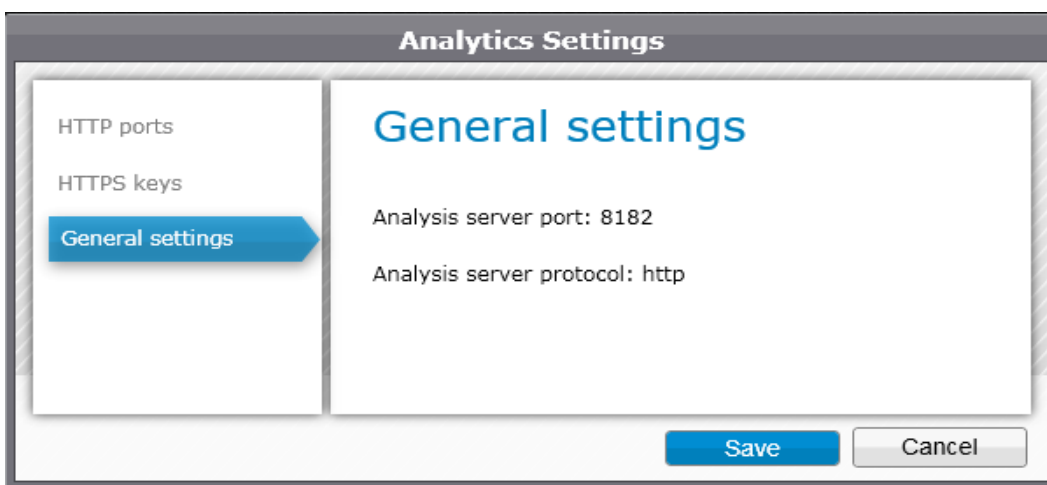
be provided.



The dialog box titled "New HTTPS key" contains three input fields: "Server IP", "Port", and "Key file path". The "Key file path" field has a file selection icon to its right. At the bottom, there are "Save" and "Cancel" buttons.

General Settings

Displays the NV Analytics server port.

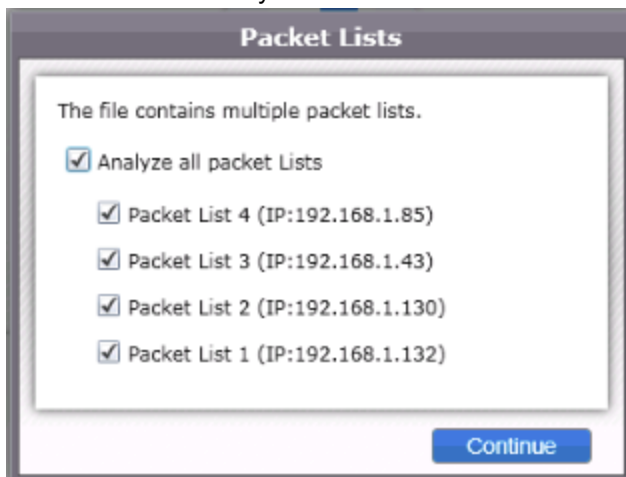


The "Analytics Settings" dialog box has a sidebar with three options: "HTTP ports", "HTTPS keys", and "General settings". The "General settings" option is selected and highlighted with a blue arrow. The main area displays "General settings" in blue text, followed by "Analysis server port: 8182" and "Analysis server protocol: http". At the bottom, there are "Save" and "Cancel" buttons.

Analyzing a Network Virtualization Test Result File

1. Start NV Analytics, and click **Open File** (to specify the port and define other settings, see "[Settings](#)" on page 15). The following file types are supported: *.shunra, *.ved, *.cap, *.pcap and *.enc.

2. When the test result file contains more than one packet list, select some or all of the packet lists to be included in the analysis.



3. After the NV Analytics window opens, in the toolbar click All Transactions to select a display of all the transactions, or select a specific transaction. The main page displays all the transactions, and each 'square' shows one transaction.

Exporting NV Analytics Results

Data of the selected view and recommendations can be exported to a .csv file and in MS Word format.

Exporting in .csv format

The export to .csv includes the throughput, the network conditions displayed in the waterfall graph, and recommendations such as the rules, grade achieved and the violations for each rule.

To export results in .csv format:

1. Click the  icon in the toolbar.

Note: When using a zoom view, the export will use the selected portion of the transaction, and not export the entire transaction.

2. In the Export settings dialog, select all or some of the reports, and any or all of the options, then click Export.

Exporting in MS Word format

When exporting in MS Word format, the HTTP Report and Recommendations can be exported.

Note: When using a zoom view, the export will use the selected portion of the transaction, and not export the entire transaction.

To export results in MS Word format:

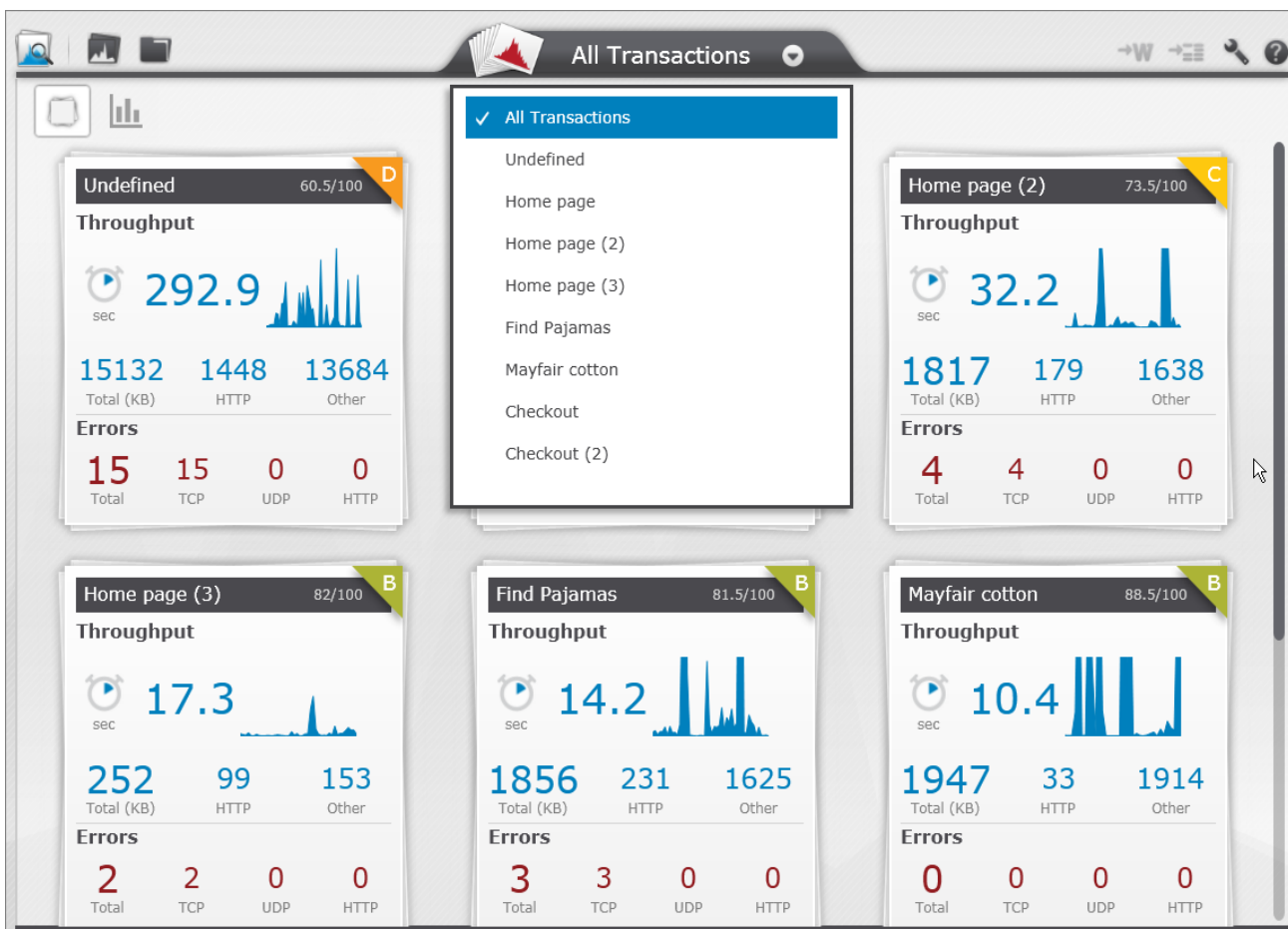
Click the  icon in the toolbar.

When using a zoom view, the export will use the selected portion of the transaction, and not export the

entire transaction. Results of searches that are highlighted are also highlighted in the exported reports.

Viewing Reports

The Overview page shows you transactions contained in the test result file you analyzed. Click on any transaction to view details of that transaction. Each Transaction section provides details of the throughput and errors, and shows a performance score with a letter and a percentage. The display is interactive, so that clicking any metric, such as "Total" displays the detailed report for that metric.




The NV Analytics Overview page displays detailed breakdown of each transaction in the following categories:

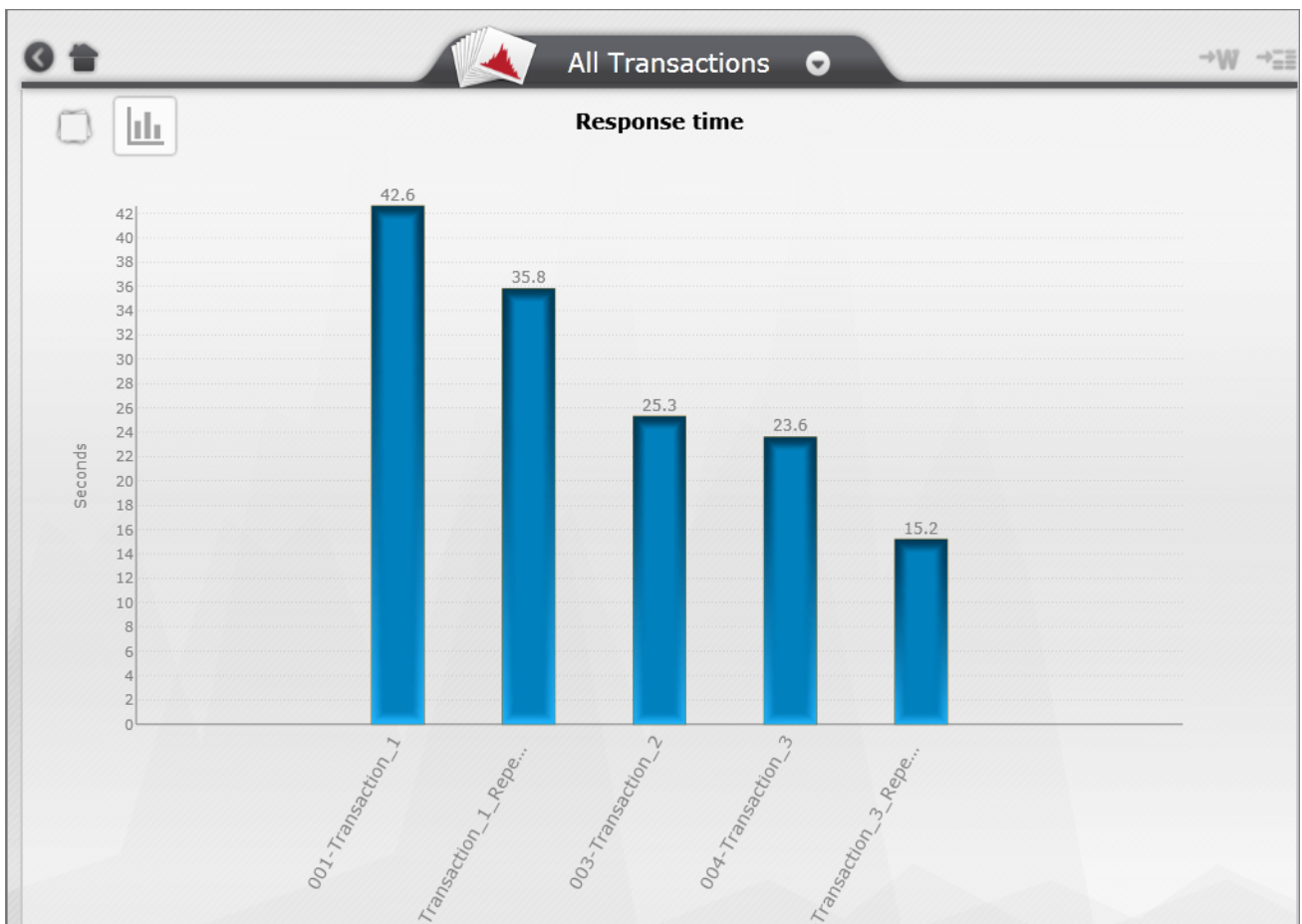
- ["Response Time" on the next page](#)
- ["Summaries" on the next page](#)
- ["General Analysis" on page 22](#)
- ["Endpoint Latencies" on page 25](#)
- ["TCP/UDP Errors & Sessions" on page 26](#)
- ["HTTP Analysis" on page 27](#)
- ["HTTP Optimization" on page 30](#)
- ["HTTP Resources and Responses" on page 31](#)

Below the list of reports (on the left side), the total Duration and Network Time for the transaction is listed. If a description was added when creating the transaction, it is also displayed.

In each of the report views, to return to the main page click the Home icon, or use the Back icon to return to the previous view, or the Forward icon to advance to the next view.

Response Time

In the All Transactions tab, click the  icon at the top left to display the Transaction Response Times of all transactions.



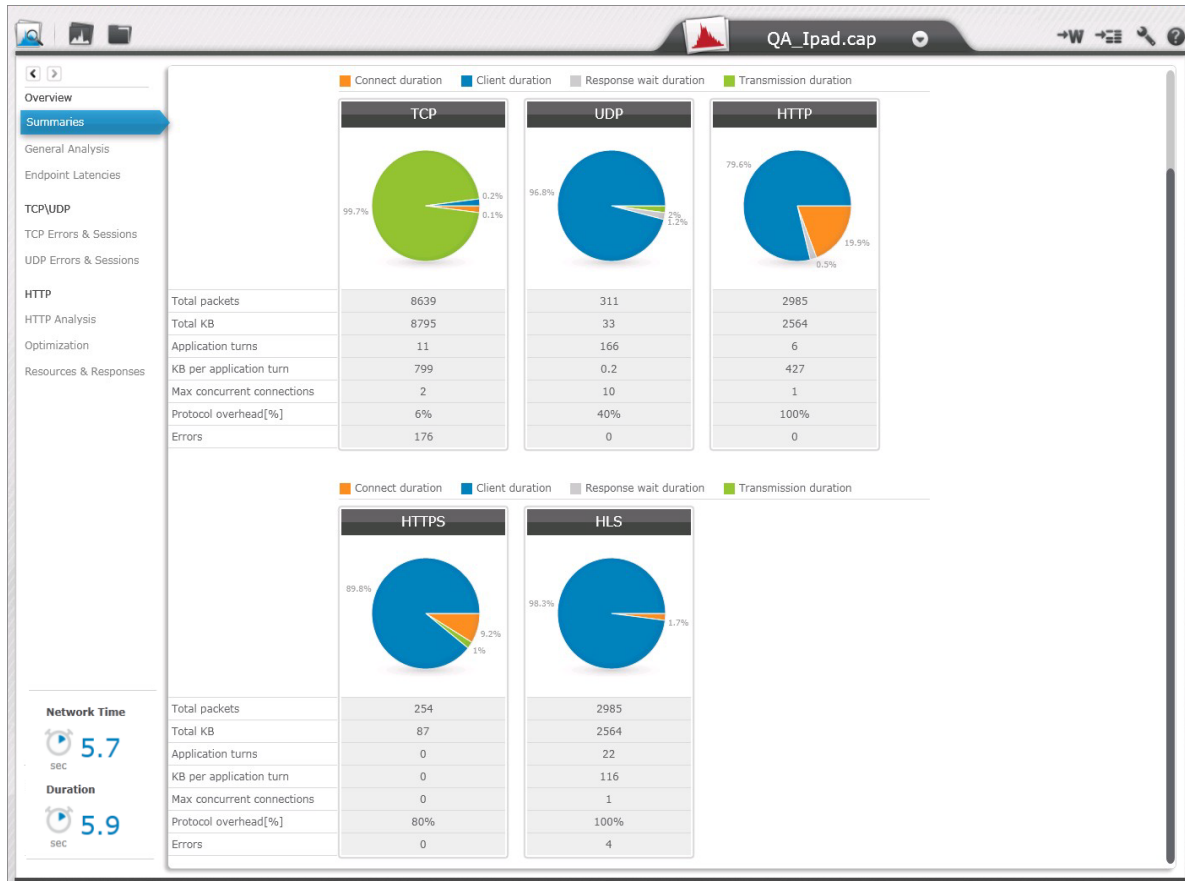
Click any bar to display the HTTP Analysis of that transaction.

Summaries

Displays the Client Network Server Breakdown of the transaction according to protocol, showing results in a pie chart and also additional details for the following protocols:

- TCP
- UCP

- HTTP
- HTTPS (secure communication)
- HLS (HTTP Live Streaming)



Client Network Server Breakdown

The legend for the chart is below the table. The values for the fields shown in each pie are:

- **Connect duration:** portion of time in which the client connected to the server, such as the request in TCP or the triple handshake in SSL (the establishment of a secure channel)
- **Client duration:** portion of time that the client processes (does not include time waiting for a server response)
- **Response wait duration:** portion of time spent waiting for the server's response
- **Transmission duration:** portion of time data was downloaded or uploaded

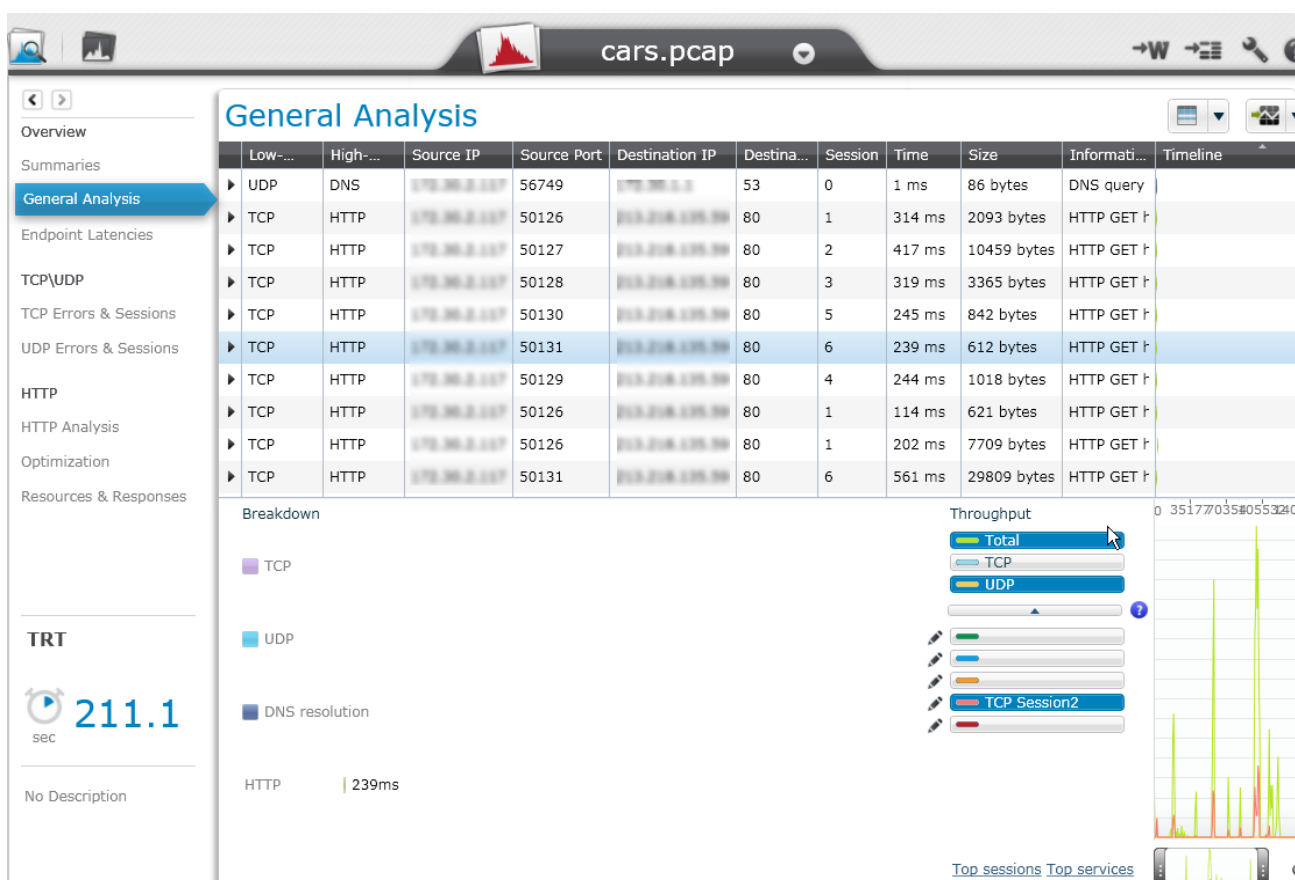
Parameters

- **Total packets:** total number of packets associated with the protocol
- **Total KB:** total throughput associated with the protocol; includes headers, such as TCP, IP, etc.

- **Application turns:** the number of times a communication flow change occurs from the request to the response, per protocol
- **KB per application turn:** the average of the throughput per Application Turn, per protocol
- **Max concurrent connections:** the highest number of concurrent connections, per protocol
- **Protocol overhead %:** the percentage of the total throughput per protocol used by non-data elements such as headers
- **Errors:** the number of errors that occurred in the transaction

General Analysis



The General Analysis displays details of all subtransactions for all protocols.



Display Options

The following options are available from the toolbar of the General Analysis and HTTP Analysis reports. The first two options are also available in the right-click menu in the table:

- **Highlight options:** click to select from the various options that highlight the resources according to the same Source IP, etc.

- Display in graph options: click  to select to display a session, service, etc. in the graph
- Filter options: click  to select parameters that limit the display of the subtransactions to the selected criteria

Subtransaction Parameters


To adjust the display of the subtransactions:

- Sort the rows according to ascending or descending order in each column (available in all tables)
- Expand or decrease the area by dragging the borders of the area (available in the General and HTTP Analysis).

The following data is provided in table format. Each row represents a subtransaction:

- **Low-level Protocol:** includes TCP and UDP
- **High-level Protocol:** includes HTTP and HTTPS, DNS, etc.
- **Source IP**
- **Source Port**
- **Destination IP**
- **Destination Port**
- **Session:** number of the TCP or UDP session in which the resource was uploaded or downloaded; color coding scheme indicates the tasks occurring in the download, such as DNS resolution, TCP setup, etc.
- **Time:** response time of the request in milliseconds
- **Size:** the number of bytes used in the request/response
- **Information:** protocol specific information (when available)
- **Timeline:** the position of the Resource within the Transaction sequence

Request/Response Details

To display the Request/Response details, expand or collapse a row with the  icon in the left column. The Client and Server portion of each Request/Response are displayed in bytes.

Breakdown

The breakdown of the share of the traffic for each protocol is displayed in the graph at the lower left of the window, including TCP, UDP, DNS Resolution and HTTP.

Throughput Graph

Note: The breakdown and throughput area can be collapsed and expanded by clicking the double arrow below the main table.

Throughput

The Throughput edit options are used to modify the display in the graph.

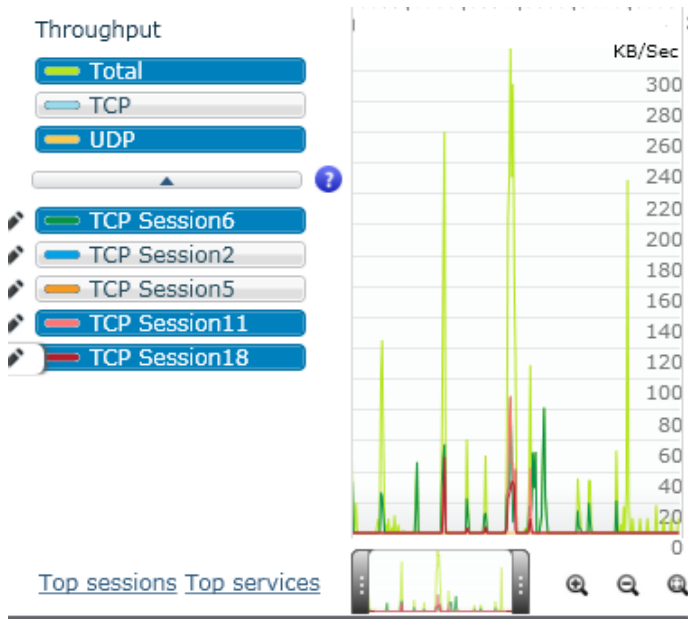
Use the zoom bar to focus on a specific period within the transaction.

- **Top Sessions:** displays up to five Sessions with the most traffic
- **Top Services:** displays up to five Services with the most traffic

In each tab of the Throughput area, click a bar to include that item, such as a specific session in the graph. The Raw data displays the throughput in KB/sec.

To add/delete components in the Throughput graph display:

1. Click one of the tabs, for example "TCP Session 18"; the graph displays the throughput for TCP Session 18 in maroon.



2. To select other sessions, services, etc., click the pencil icon for each session or hosts that is to be displayed.

	Name	Raw Data(KB)
<input checked="" type="checkbox"/>	TCP Session6	873
<input checked="" type="checkbox"/>	TCP Session2	402
<input checked="" type="checkbox"/>	TCP Session5	398
<input checked="" type="checkbox"/>	TCP Session11	385
<input type="checkbox"/>	TCP Session19	247
<input checked="" type="checkbox"/>	TCP Session18	217
<input type="checkbox"/>	TCP Session24	96

3. The following tabs are available and each displays the traffic (in KB/Sec) for each of the following values:

- TCP Sessions
 - UDP Session
 - Source IP
 - Destination IP
 - Services (IP Address and Port)
4. Select the required item, for example "TCP session 16" or a Service Name. The metric is displayed in the graph with the selected color.

Endpoint Latencies

The Endpoint Latencies report displays details of the latency observed at the client and server endpoints.

- **Source IP**
- **Destination IP**
- **Names:** The name of the server
- **Best estimate (ms):** Best estimate of the latency between the client and server, as deduced from the TCP connections between the two, taking into consideration any additional latency found in the packet capture due to bandwidth constraints (thus, it's possible for this value to be lower than the Min value)
- **Min (ms):** the minimum value observed in the packet capture
- **95th percentile (ms):** the maximum value observed in the packet capture, excluding outermost conditions
- **Max (ms):** the maximum value observed in the packet capture
- **Samples:** the number of packets used in the calculation of the latency

Endpoint Latencies

Source IP	Destination IP	Name\s	Best...	Min(ms)	5th...	95th...	Max...	Samples
172.30.2.145	217.149.21.34		43.081	25	25	67	225	1080
172.30.2.145	63.35.33.46		70.065	25	25	222	224	55
172.30.2.145	209.85.148.34		96.447	96	96	117	117	16
172.30.2.145	66.251.196.19		97	97	97	97	97	4
172.30.2.145	64.298.186.42		99.333	95	95	116	116	6
172.30.2.145	23.14.86.102		99.731	99	99	115	115	8
172.30.2.145	217.149.21.41		101.197	93	94	112	144	44
172.30.2.145	209.85.148.13		101.233	100	100	111	111	4
172.30.2.145	63.35.33.4		105.333	104	104	107	107	3
172.30.2.145	63.112.196.35		108	107	107	126	126	3
172.30.2.145	64.298.186.25		109	98	98	120	120	3
172.30.2.145	2.14.18.102		113.5	113	113	114	114	3
172.30.2.145	63.35.33.32		114.833	112	112	117	117	6
172.30.2.145	66.137.18.87		152	150	150	164	164	30
172.30.2.145	64.298.186.18		152.768	150	150	162	192	119
172.30.2.145	64.298.121.17		153.967	151	151	157	178	67
172.30.2.145	134.35.248.89		168.266	163	164	175	208	122
172.30.2.145	209.177.76.14		171	170	170	173	173	3
172.30.2.145	66.137.18.208		208.416	206	207	210	235	86
172.30.2.145	178.17.138.155	Site 22	225.75	224	224	226	226	10

TRT
292.9 sec
No Description

TCP/UDP Errors & Sessions

Select **TCP errors & sessions** to display the details of transaction that used the TCP protocol. Select **UDP errors & sessions** to display the details of transaction that used the UDP protocol.

TCP Errors & Sessions

▼ Errors
No protocol errors found

▼ Sessions

Session	Source Address	Source Port	Destination Address	Destination Port	Total Responses	Average...	Total Errors	Error Types
▶ 0	10.0.0.88	49208	10.0.0.56	80	1	1	0	No Errors
▶ 1	10.0.0.88	49209	10.0.0.56	80	1	0	0	No Errors
▶ 2	10.0.0.88	49210	10.0.0.56	80	1	1	0	No Errors
▶ 3	10.0.0.88	49211	10.0.0.56	80	1	1	0	No Errors
▶ 4	10.0.0.88	49212	10.0.0.56	80	1	0	0	No Errors
▶ 5	10.0.0.88	49213	10.0.0.56	80	1	0	0	No Errors

Session details for TCP and UDP are displayed separately, and each includes:

- **Session**
- **Source Address**
- **Source Port**
- **Destination Address**
- **Destination Port**
- **Total Responses**
- **Average Response Time**
- **Total Errors**
- **Error Types**

HTTP Analysis

The details of the transaction are displayed in both table and graph format.

The screenshot shows the HP NV Analytics interface for a file named 'cars.pcap'. The 'HTTP Analysis' section is active, displaying a table of transactions. Below the table, a detailed view of a selected transaction is shown, including a throughput graph and various timing metrics.

	Status	Type	Resource	Host	Session	Instances	Time	Size	Timeline
▶	200	HTML	/	www.ca	1	1	340 ms	2093 bytes	
▶	200	7	/styleTransverse.css	www.ca	2	1/11	419 ms	10459 bytes	
▶	200	7	/styleTransverse.css	www.ca	3	1/11	320 ms	3365 bytes	
▶	200	7	/styleLogin.css	www.ca	5	1	247 ms	842 bytes	
▶	200	7	/modalbox.css	www.ca	6	1	241 ms	612 bytes	
▶	200	7	/styleLogin.css	www.ca	4	1	246 ms	1018 bytes	
▶	200	7	/modalbox.css	www.ca	1	1	116 ms	621 bytes	
▶	200	JS	/include.js	www.ca	1	1	203 ms	7709 bytes	
▶	200	JS	/prototype.js	www.ca	6	1	562 ms	29809 bytes	
▶	200	JS	/scriptaculous.js	www.ca	5	1/11	130 ms	2136 bytes	
▶	200	JS	/effects.js	www.ca	4	1	291 ms	9951 bytes	
▶	200	JS	/modalbox.js	www.ca	3	1	179 ms	7674 bytes	
▶	200	JS	/iopopup.js	www.ca	5	1	2524 ms	1831 bytes	
▶	200	7	/logo_login.gif	www.ca	1	1	83 ms	5616 bytes	

TRT
211.1 sec
No Description

HTTP request/response


- DNS: 24 ms (wait 23 ms)
- Connect time: 97 ms (wait 1 ms)
- Request: 1 ms
- Wait: 110 ms
- Response: 108 ms

Throughput

- Total
- TCP
- HTTP

The throughput graph shows data points at 0, 35177, 70354, 105532, 140709, and 17.

The following options are available in the toolbar of the General Analysis and HTTP Analysis reports. The first two options are also available in the right-click menu in the table:

- Highlight options: click  to select from the various options that highlight the resources according to the same Source IP, etc. This example shows all the Session's participants.

HTTP Analysis

	Status	Type	Resource	Host	Session	Instances	Time	Size	Timeline
▶	200	HTML	/	www.ca	1	1	340 ms	2093 bytes	
▶	200	?	/styleTransverse.css	www.ca	2	1/11	419 ms	10459 bytes	
▶	200	?	/styleTransverse.css	www.ca	3	1/11	320 ms	3365 bytes	
▶	200	?	/styleLogin.css	www.ca	5	1	247 ms	842 bytes	
▶	200	?	/modalbox.css	www.ca	6	1	241 ms	612 bytes	
▶	200	?	/styleLogin.css	www.ca	4	1	246 ms	1018 bytes	
▶	200	?	/modalbox.css	www.ca	1	1	116 ms	621 bytes	
▶	200	JS	/include.js	www.ca	1	1	203 ms	7709 bytes	
▶	200	JS	/prototype.js	www.ca	6	1	562 ms	29809 bytes	
▶	200	JS	/scriptaculous.js	www.ca	5	1/11	130 ms	2136 bytes	

- Display in graph options: click to select to display a session, service, etc. in the graph

To search for similar resources according to URL:

1. In the toolbar, enter the required string in the Search area, in this example "in".

HTTP Analysis

	Status	Type	Resource	Host	Session	Instances	Time	Size	Search Results
▶	200	HTML	/redirect.html	10.0.0.56	0	1	13 ms	1746 bytes	
▶	304	?	/frames.htm	10.0.0.56	1	1	7 ms	797 bytes	
▶	304	?	/main.htm	10.0.0.56	2	1	10 ms	799 bytes	
▶	404	HTML	/UntitledFrame-1.htm	10.0.0.56	3	1	10 ms	2301 bytes	
▶	304	?	/menu.htm	10.0.0.56	4	1	5 ms	799 bytes	
▶	404	HTML	/loop-relax.mp3	10.0.0.56	5	1/6	5 ms	2193 bytes	
▶	304	?	/osho_menu.jpg	10.0.0.56	6	1	6 ms	696 bytes	
▶	304	?	/star.gif	10.0.0.56	7	1	6 ms	691 bytes	
▶	304	?	/oshobw.jpg	10.0.0.56	8	1	5 ms	693 bytes	
▶	304	?	/mcloud2.jpg	10.0.0.56	9	1	5 ms	694 bytes	
▶	404	HTML	/loop-relax.mp3	10.0.0.56	10	2/6	22 ms	2193 bytes	
▶	304	?	/info.htm	10.0.0.56	11	1	5 ms	802 bytes	
▶	304	?	/info.jpg	10.0.0.56	12	1	6 ms	691 bytes	

All matching resources are highlighted, in this case strings that contain "in". Use the arrows below the Search area to navigate between the results.


2. Clear the search results by closing the Search area (clicking the "X").

Subtransaction Details

To adjust the display of the request/responses:

- Sort the rows according to ascending or descending order in each column.
- Expand or decrease the area by dragging the borders of the area.

The following columns are displayed in the report:

- **Expand/Collapse icon** : Show or hide details of each Request.
- **Up/down arrows**: The up arrow indicates a POST or PUT; down arrow indicates a GET; a head indicates an icon header; N/A indicates HTTPS; for all other types a star is shown.
- **Status**: The HTTP status, such as 404 (page not found) or 200 (OK).
- **Type**: The icon indicates the type of file requested, for example a graphic file.
- **Resource**: Displays the path that was accessed by the subtransaction.
- **Host**: The host (domain, server, etc.) from which the resource is uploaded or downloaded.
- **Session**: Number of the TCP session in which the resource was uploaded or downloaded.
- **Instances**: Number of times a resource with the same name appears in the transaction.
- **Time**: Response time of the request in milliseconds.
- **Size**: The number of bytes used in the request/response.
- **Timeline**: The position of the Resource within the Transaction sequence.

The following data is displayed per Transaction (at the left):

- **Network Time**: The time it takes the transaction to complete between the first packet of the request and the last packet of the transaction.
- **Duration**: The total time between the transaction start and end.
- **Description**: Displays the transaction's description.

To customize the display of the graph:

- The graph at the bottom-right displays the throughput. The 'handles' can be used to focus on a specific period within the Request/Response.
- Select HTTP Throughput to display the HTTP data only, or Total Throughput to display all the data in the Request/Response.
- When the mouse moves over this area, a line moves with the mouse to connect the Request/Response area and the graph view, and indicates the time (in seconds) when this occurred during the capture. Use the zoom icons to zoom in, zoom out and to zoom out or to select no zoom



HTTP Parameters

When a resource is highlighted, the area in the below the table displays the following breakdown:

- **DNS Resolution**: Includes the wait time from the resolve time to the start of the connection (the time between the DNS query and the first SYN to the server whose name was queried).
- **Connect Time**: The TCP Setup time and any wait time between the initialization of the connection and sending of the first request data packet.
- **TLS Time**: SSL/TLS secure channel establishment.
- **Request**: Time required for the client to send the request to the server.
- **Wait**: The time (in ms) between the last packet of the request and the first packet of the response.
- **Response**: The time (in ms) between the first packet and the last packet of the response.
- **Encrypted Data Transmission**: The duration of an encrypted HTTPS session.

To display the details of a Request/Response:

Double-click the Request/Response name, or click the Expand/Collapse icon; the following columns are displayed:

- **Request Headers:** The syntax of the request header.
- **Response Headers:** The syntax of the response header.
- **Request Content/Response Content:** Displays the image, HTML data, etc. (non-printable characters may be replaced by a " . " Content in certain formats such as PDF, Audio, Video, Flash, and Fonts are not displayed.
- **Details:** Throughput of the request/response; for a HTTP POST the full URL is also displayed.

The screenshot shows a table of resources with columns for status, size, and time. The second resource is selected, and its details are shown in a panel below. The details panel has tabs for Request Headers, Response Headers, Response Content, and Details. The Response Content tab is active, showing a copyright notice and license terms.

Status	Size	Time	Resource Name	Host	Port	Order	Content Type
200	29809 bytes	562 ms	/prototype.js	www.ca	6	1	text/javascript
200	2136 bytes	130 ms	/prototype.js	www.ca	5	1/11	text/javascript

Request Headers
// [redacted] Tue Nov 06 15:01:40 +0300 2007

Response Headers
// Copyright (c) 2005-2007 [redacted]

Response Content
// Permission is hereby granted, free of charge, to any person obtaining
// a copy of this software and associated documentation files (the
// "Software"), to deal in the Software without restriction, including
// without limitation the rights to use, copy, modify, merge, publish,
// distribute, sublicense, and/or sell copies of the Software, and to
// permit persons to whom the Software is furnished to do so, subject to
// the following conditions:

Note: When the Resource is selected (highlighted), details of the Resource are displayed below.

HTTP Optimization

NV Analytics provides a number of Best Practice recommendations, based on data obtained from external sources, in addition to the knowledge obtained within HP from current application testing methodology.

Each transaction is given a score comparing the transaction to the best practice. The Total Score (out of 100) is the summary of the individual scores, based on the level of compliance to website programming rules.

Individual numeric scores:

Prioritization emphasizes which transactions most affect the results. Each recommendation is weighted, based on the potential performance improvement to be obtained if the recommendation is performed, and is displayed in a negative numeric point value. For example, an individual score of -8 points indicates that non-compliance with the specific recommendation more significantly affects transaction response time than non-compliance with another recommendation that received a score of -2 points.

Letter grades:

In addition, a value from A - F is provided, indicating level of compliance with the best practice, with F representing the lowest level of compliance.

Example:

In the example below, there are multiple best practices that received grades of 'F', while the individual numeric scores vary. This indicates that while the transaction shows low compliance with all of these recommendations, some are more problematic, and will more significantly affect transaction results. For the best practice recommendation to "Reduce the size of your images" - perhaps the images are quite large, but there are not many of them. Therefore, a greater benefit would be obtained by implementing the recommendation with a grade of 'F' and a score of -8, than by implementing a recommendation with a grade of 'F' and a score of -4.





Grade	Recommendation	Score
F	Make fewer HTTP requests (desktop)	- 8 Point
F	Don't download the same data twice	- 8 Point
F	Avoid image scaling in HTML	- 4 Point
F	Minify your textual components	- 4 Point
F	Try to reduce the size of the cookies	- 4 Point
F	Avoid loading javascripts in the head section	- 4 Point
F	Reduce the size of your images (desktop)	- 4 Point
E	Avoid 404 error code (Not Found) errors	2 Point

To display the list of transactions with the relevant recommendations:

1. Select **Optimization** and select the rules to be included in the report; by default all are selected. The report can be displayed more than once with different rules.
2. Select **Desktop** or **Mobile**; you can also sort the results according to the **Priority** (according to the number of points given to each result) or by **Name** (alphabetical order).
3. To display additional details, click a recommendation. For example, for the recommendation "Don't download the same data twice", the details show the number of times the file appeared. Each can be clicked and viewed in the HTTP Analysis.

Note: The score and recommendations may be different for desktop and mobile results, due to platform specific optimizations.

To filter the list of rules for both mobile and desktop:

1. Click the Edit (pencil) icon  at the top right.
2. Select or deselect a rule, then click **Save**. When only some of the rules are selected, an "information icon" appears beside the Edit icon .

HTTP Resources and Responses

The Resources Breakdown displays the Instances and Total Throughput for each type of resource that was present in the transaction.

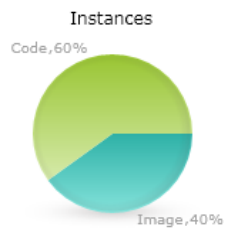
Resources Breakdown

The Instances pie chart shows the number of times that each type of resource appears in the transaction. The chart is divided according to type of resource, so that .jpeg images are one category, and css files are another category.

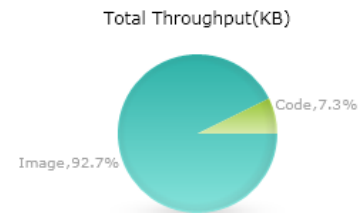
The Total Throughput chart shows the breakdown according to the same categories as the Instances chart, but the size is calculated according to the Total Throughput in KB.

HTTP Resources & Responses

▼ Resources Breakdown



Type	Name	Instances
Image	Gif	1
Code	HTML	6
Image	Jpeg	3



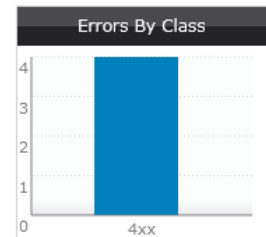
Type	Name	Total Throughput(KB)
Image	Gif	158
Code	HTML	15
Image	Jpeg	33

HTTP Errors

The table displays each error according to the subtransaction for which it occurred. The Errors by Class graph displays the totals according to the class, such as 4xx for HTTP client errors, and 5xx for server errors.

▼ Errors

	Subtransaction	Total...	Error Name
▶	GET http://10.0.0.56/UntitledFrame-1.htm	1	HTTP 404
▶	GET http://10.0.0.56/main/loop-relax.mp3	3	HTTP 404



HLS Errors

The table displays a summary of errors that occurred during video streaming.

▼ HLS Errors

Subtransaction	Total...	Error Name
▼ http://...am/2011/05/27/...	4	Video Buffering
<ul style="list-style-type: none"> • Video Buffering - Video Buffering (4 time) 		

Response Summary

Displays the number of occurrences of each HTTP response code.

▼ Response Summary

Response code	Occurrence
▣ 2xx	
▶ 200 OK	6
▣ 4xx	
▶ 404 Not found	4

Secure Communication

Secure HTTPS communication can be viewed in the NV Analytics report, including the Host Name, TCP Session Establishment and SSL Session Establishment data.

The private key must be a text file in the Privacy Enhanced Mail (PEM) format, as in this example:

```
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQC20yQBpiE1atCfIui0qa9fnPyHrCsYCoCJ/hK1S8z6IOcmBsNq
kZckz1MqcQP7i9o3SGVBXv1rAsGN3SnkqNUD4PFukUHQrjPpc0KPX9KdDCcLFu5f
Bxq+7/7hDzQA+rKWqix03WmBVcKQm+3WvpF5M+jJuIsRY806r0xb+FrpvwIDAQAB
AoGBAKWpszHPUL4vmPTqU+KZ/bDc9rMVnkL9mTXxRQIFKqroT6vUaxTQ8i1GzHNj
zyELu+NmNWD6cwixjJ/fap3HJNWMF0byZEwPyC5yKkEZQDKt3n549nTPxM
wD09geHjqwJBANZu41xS/+4WkqW5yZh8VCKmMUc0PQvXw+niXrjSucc5k5VdHw4
q0obXalp1PyrafHj6YV8PfX9XHpPRAzs/j0CQBq8iJqHttBpzc+0buCthtPR7XHT
5BTuuG3rPFoW+R/D8K2apQSoj2uEgxSFLcvpcaninPHEo0b08SfqLqCmZxkCQADB
dKA13LwQ7wktDQ2K4bIWu8Gd+d/gCJtBajVJj1UZMnqBsPOGLnaxIVC6EZXpAYVs
CdT0yDKhjqsWggkjMwkCQHZvP1E28M51k1pLsQx43nq7zbueKZwkDg/biA3y0aLb
FJ9T5JeuFAXAmG/US+zCfGLuzrSuJwHiCMnhRrB0m+Y=
-----END RSA PRIVATE KEY-----
```

If it is not in this format, use Open SSL to change the format of the private key. Determine the operating system of the Server. On Microsoft® Windows, keys are often stored in PKCS7/DER format (locally) or in .NET format (from any directory server). To convert use these commands:

```
# for PKCS7/DER keys (as held on disk)
openssl pkcs8 -nocrypt -in derfile.key -inform DER -out key.pem -outform PEM
# for NET keys (from the directory server)
openssl pkcs8 -nocrypt -in file.ick -inform NET -out key.pem -outform PEM
```

On Mac OSX, Solaris, and other systems the file format used is often PKCS#12. To convert use this command:

```
openssl pkcs12 -nodes -in file.p12 -out key.pem -nocerts -nodes
```

On Linux use these commands:

```
openssl x509 -nocrypt -in foo.der -informat DER -out key.pem -outformat PEM
openssl x509 -nocrypt -in foo.net -informat NET -out key.pem -outformat PEM
```

Note: To analyze secure communication, refer to the SSL options in ["HTTPS Keys" on page 16](#).

Chapter 4: NV Analytics API

NV Analytics API uses Representation State Transfer (REST) web services architecture. The analysis API requests all have a same URL structure, the prefix is:

[base address]/shunra/api/analysis

Note: A code sample "analyzer.py" in Python is available in the installation folder. Updates can be found in <https://gist.github.com/2773832>

It can be used to access the API. Segments of the code are also provided in this document.

The following methods are exposed:

- [AnalysisEngines](#) 35
- [Extract Packet Lists](#) 36
- [AnalysisRequest](#) 38
- [AnalysisSummary](#) 41
- [AnalysisArtifact](#) 42

AnalysisEngines

Provides a JSON contents list of the installed analysis engines.

GET

[base address]/shunra/api/analysis/engines

Example: <http://localhost:8182/shunra/api/analysis/engines>

Response

The response includes the id and names of all the analysis engines.

```
{
  "supportedAnalysisEngines": [{"name": "harExport", "id": "harExport"},
  {"name": "networkmeasurements", "id": "networkmeasurements"},
  {"name": "generalWaterfall", "id": "generalWaterfall"}, {"name": "http", "id": "http"},
  {"name": "iostats", "id": "iostats"}, {"name": "metrics", "id": "metrics"},
  {"name": "best practices", "id": "best practices"}]}
}
```

Returns

- 200 "OK"
- 404 "Not Found"
- 500 "Internal Server Error"

Code sample

```
def get_engine_id(engine_name):  
    """  
    Returns the analysis engine id, given its name.  
    This can also be used as a sort of a sanity test for the analysis api.  
  
    >>> get_engine_id('best practices')  
    u'best practices'  
    """  
    resp = get('/shunra/api/analysis/engines')  
    engines = dict([(entry['name'], entry['id']) for entry in resp  
['supportedAnalysisEngines']])  
    return engines[engine_name]
```

Extract Packet Lists

Provides a JSON contents of packet list names, IDs, endpoints and .pcap and .ved file unique IDs.

PUT

[base address]/shunra/api/analysis/packetlistmetadata

Example: <http://localhost:8182/shunra/api/analysis/packetlistmetadata>

Body

The JSON defines the analyzed emulation result (.ved or .pcap file) ID. It is a file system path for NV Analytics:

```
{  
  "id": "C:\\tmp\\Sample.ved"  
}
```

The response includes the ID of the analyzed run result and the packet list metadata (names, IDs, endpoints):

```
{  
  "packetLists": [{  
    "endpoints": [{  
      "name": "Tokyo Office",  
      "id": "6d0652db88c349de9382a54dc350349f"  
    }],  
    "name": "Packet List 3",  
    "id": "c6064d9bf25d405382e374795fef35fe"  
  },  
  {  
    "endpoints": [{
```

```
        "name": "London Office",
        "id": "de358779547c4eea8caef62bfbbb493"
    }],
    "name": "Packet List 2",
    "id": "59220e1cb4d248eba3b89a695918be91"
},
{
    "endpoints": [{
        "name": "NY Office",
        "id": "8c95498f7bb04c7598dde1d5e609082a"
    }],
    "name": "Packet List 1",
    "id": "620984c9a31b4ef694a1ac47d61b6a7e"
}],
"runResultId": "b80de7f5ffa97428b2324c8b3a9d469b"
}
```

Returns

- 200 "OK"
- 404 "Not Found"
- 500 "Internal Server Error"

Code sample

```
def get_packetlists(inputfilepath):
    """
    Returns a dictionary of the available packet lists in the given file.
    The dictionary keys are the packet lists names, and the dictionary values are the
    packet lists ids.

    >>> packetlists = get_packetlists(os.path.join(SAMPLE_FOLDER, 'Sample.ved'))
    >>> len(packetlists)
    3
    >>> 'Packet List 1' in packetlists
    True
    """
    resp = put('/shunra/api/analysis/packetlistmetadata', {'id':inputfilepath})
    return dict([(entry['name'], entry['id']) for entry in resp['packetLists']])

def get_run_result_id(inputfilepath):
    resp = put('/shunra/api/analysis/packetlistmetadata', {'id':inputfilepath})
    return resp['runResultId']
```

AnalysisRequest

Represented by JSON; provides contents of current status of analysis process per packet list, per transaction and per analysis engine.

The response is a dictionary with the following entries:

- **transactionAnalysisStatus** – a list of transactions as described below
- **reportId** – an identifier for the analysis process
- **name** – name of the analyzed packet list
- **id** – id of the analyzed packet list

The list of transactions contains an entry for each transaction associated with the packet list.

Each entry is a dictionary, containing the transaction id (id), name (name) and analysis status (analysisStatusPerEngine).

Analysis status is a dictionary whose keys are the analysis engine, and the values are their status as specified in the API documentation.

PUT

[base address]/shunra/api/analysis/request/{plid}

Example:

```
http://localhost:8182/shunra/api/analysis/request/620984c9a31b4ef694a1ac47d61b6a7e
```

Where "plid" is a packet list unique ID that has been returned by the Extract Packet List request.

Body

Contains the analysis parameters such as ports, SSL Encryption Key, and the analyzed emulation result (.ved or .pcap file) ID. and the file system path, because the file is not persisted (retained?) by the system. The body is in JSON format:

```
{  
  "ports": "80, 8080",  
  "sslEncryptionKey": "172.30.2.31,443,http,C:\\keys\\secret.key",  
  "runResultHandle": "C:\\tmp\\Sample.ved"  
}
```

The response includes a current analysis status per transaction, per installed analysis engines and the generated analysis report id identifying the analysis parameters:

```
{  
  "transactionAnalysisStatus": [{  
    "analysisStatusPerEngine": {  
      "networkmeasurements": "Started",  
      "harExport": "Started",  
      "generalWaterfall": "Started",  
      "http": "Started",  
    }  
  }  
}
```

```
        "iostats":"Started",
        "metrics":"Started",
        "best practices":"Started"
    },
    "name":"Undefined",
    "id":"ccb8713e522241c9a691c4ed1ce72d27"
}],
"reportId":"-561678026",
"name":"Packet List 1",
"id":"620984c9a31b4ef694a1ac47d61b6a7e"
}
```

Where possible analysis statuses are:

```
public enum WorkStatus {
// a job still has not been started, not proceeded, analyzed, etc
Idle(0),
// a job (for example emulation or analysis) started
Started(1),
// a job (for example emulation or analysis) finished
Finished(2),
// a job (for example analysis) failed
Failed(3);
}
```

Note: The heuristic for analysis process completeness is that all the items have either a Finished or Failed status; otherwise some items in the analysis jobs pool have not completed yet.

The client side should continue to process analysis requests until the analysis process has completed.

Returns

- 200 "OK"
- 404 "Not Found"
- 500 "Internal Server Error"

Code Sample

```
def analyze(inputfilepath, packetlist_id, settings={}):
    """
        calls analysis on a given file (use settings to pass special analysis parameters
        such as port numbers and ssl keys)

        packetlist_id should be the id return by get_packetlists for a specific packet
        list.
```

The response is a dictionary with the following entries:

- * transactionAnalysisStatus - a list of transactions as described below
- * reportId - an identifier for the analysis process
- * name - name of the analyzed packet list
- * id - id of the analyzed packet list

The list of transactions contains an entry for each transaction associated with the packet list.

Each entry is a dictionary, containing the transaction id (id), name (name) and analysis status (analysisStatusPerEngine).

Analysis status is a dictionary whose keys are the analysis engine, and the values are their status as specified in the API documentation.

```
>>> inputfilepath = os.path.join(SAMPLE_FOLDER, 'Sample.ved')
>>> packetlists = get_packetlists(inputfilepath)
>>> packetlist_id = packetlists['Packet List 1']
>>> resp = start_analysis(inputfilepath, packetlist_id)
['transactionAnalysisStatus']
>>> len(resp) # only one transaction is associated with this packet list
1
>> resp[0]['name']
u'Undefined'
>>> resp[0]['analysisStatusPerEngine']['http'] in ['Idle', 'Started', 'Finished',
'Failed']
True
"""
params = dict(settings)
params['runResultHandle'] = inputfilepath
resp = put('/shunra/api/analysis/request/'+packetlist_id, params)
return resp
def get_report_id(inputfilepath, packetlist_id, settings={}):
return analyze(inputfilepath, packetlist_id, settings)['reportId']

def get_transactions(inputfilepath, packetlist_id, settings={}):
"""
Gets all the transactions associated with a given packetlist.
The result is a list of pairs, the first element of each pair is the transaction
id, and the second is the transaction's name

>>> inputfilepath = os.path.join(SAMPLE_FOLDER, 'Sample.ved')
>>> packetlists = get_packetlists(inputfilepath)
>>> packetlist_id = packetlists['Packet List 1']
>>> result = get_transactions(inputfilepath, packetlist_id)
>>> len(result) # only one transaction is associated with this packet list
1
>>> result[0][1]
u'Undefined'
"""
return [(transaction['id'], transaction['name']) for transaction in analyze
```



```
(inputfilepath, packetlist_id, settings)['transactionAnalysisStatus']]

    def start_analysis(inputfilepath, packetlist_id, settings={}):
    """
    Starts analysis on a given file.

    The response is a list, with an entry for each transaction associated with the
    packet list.
    Each entry is a dictionary, containing the transaction id (id), name (name) and
    analysis status (analysisStatusPerEngine).
    Analysis status is a dictionary whose keys are the analysis engine, and the values
    are their status as specified in the API documentation.
    """
    return analyze(inputfilepath, packetlist_id, settings)

def is_analysis_done(inputfilepath, packetlist_id, settings={}):
    """
    Returns True if all the transactions associate with the given packet list were
    analyzed and their reports are ready to be fetched.
    """
    resp = analyze(inputfilepath, packetlist_id, settings)
    ['transactionAnalysisStatus']
    for transaction in resp:
        for engine_status in transaction['analysisStatusPerEngine'].values():
            if engine_status in ['Idle', 'Started']:
                return False
    return True
```

AnalysisSummary

Represented by JSON; provides the contents of analysis summary per packet list, per transaction, and per analysis engine.

GET

[base address]/shunra/api/analysis/summary/{runresulthandle}/{plid}/{reportId}/
{engineId}

Example:

```
http://localhost:8182/shunra/api/analysis/summary/b80de7f5ffa97428b2324c8b3a9d46
9b/620984c9a31b4ef694a1ac47d61b6a7e/-561678026/best%20practice
```

GET

[base address]/shunra/api/analysis/summary/{runresulthandle}/{plid}/{reportId}/
{trId}/{engineId}

Example:

```
http://localhost:8182/shunra/api/analysis/summary/b80de7f5ffa97428b2324c8b3a9d469b/620984c9a31b4ef694a1ac47d61b6a7e/-561678026/best%20practices
```

The first call returns the requested analysis report for all transactions in packet list. The second returns it for the specified transaction only.

The following report types are currently supported:

- **http:** HTTP Analysis
- **best practices:** Optimization report
- **iostats:** Throughput report
- **general/waterfall:** General Analysis
- **metrics:** the protocols' summary and metrics report
- **networkmeasurements:** Endpoints Latencies report
- **harExport:** a report containing the HTTP subtransaction in HAR format (experimental)

Response

GET

```
'/shunra/api/analysis/summary/%s/%s/%s/%s/%s'%(run_result_handle, packetlist_id, report_id, transaction_id, engine_id)
```

Returns

```
resp['successfulTransactionAnalysis'][0]['result'] ???
```

- 200 "OK"
- 404 "Not Found"
- 500 "Internal Server Error"

Code Sample

```
def get_analysis_report(run_result_handle, packetlist_id, report_id, transaction_id, engine_id):  
    """  
    Get the result of running one of the analysis engines on a given packet list  
    """  
  
    resp = get('/shunra/api/analysis/summary/%s/%s/%s/%s/%s'%(run_result_handle, packetlist_id, report_id, transaction_id, engine_id))  
    return resp['successfulTransactionAnalysis'][0]['result']
```

AnalysisArtifact

A file found within a transaction, such as a picture, movie, doc, text, etc.

This section contains:

- ["Structure of an Analysis Report" below](#)
- ["Structure of the HTTP Waterfall Analysis Report" on the next page](#)
- ["Structure of the Best Practices Analysis Report" on page 46](#)

GET

[base address]/shunra/api/analysis/artifact/{filehandle}

Where the artifact handle is taken from the analysis report (See ["Structure of an Analysis Report" below](#)).

Example:

```
http://localhost:8182/shunra/api/analysis/artifact/620984c9a31b4ef694a1ac47d61b6a7e%2F-561678026%2Fccb8713e522241c9a691c4ed1ce72d27%2F94660f9c01724f63bedfefb370dc4575%2Fabf8bde63762421dbe29cab1cecae661
```

Returns

- 200 "OK"
- 404 "Not Found"
- 500 "Internal Server Error"

Structure of an Analysis Report

As a result of executing step (4) "Get Analysis Result" for a specific transaction, the API returns a JSON document of the following format:

```
{
  "name": "Packet List 1",
  "successfullTransactionAnalysis": [{
    "status": "Finished",
    "result": {
      "type": "Best Practices Report",
      "subtype": "Web Applications Best Practices Report",
      "version": "0.5",
      -- Other, analysis engine dependent fields --
    },
    "name": "Undefined",
    "id": "fe15bdf3eafe4ec8bb1b055c49ca622b"
  }],
  "reportType": "best practices",
  "reportId": "129778102",
  "id": "1ae2d2ee02144e69801e0f3d1cb39d89",
  "failedTransactionAnalysis": []
}
```

Note: The text in blue will list the actual report. Notice that since the request is for a specific

engine for a specific transaction, the results for "successfulTransactionAnalysis" contains only one entry, which lists the report for that transaction, wrapped with the transaction's name and ID. The reports all share a common structure, with "type", "subtype" and "version" fields.

Structure of the HTTP Waterfall Analysis Report

A typical HTTP Waterfall report contains many entries in the "subTransactions" list, each is one of two possible types:

- **HTTP request/response:** contains a singles HTTP request coming either from a HTTP session or a decrypted HTTPS session
- **HTTPS session:** contains details about an non-decrypted HTTPS session (highlighted in red)

Not all fields are mandatory. Below, only the fields marked in red are guaranteed to be available in each entry. For example, if the response to a given request was not captured in the packet list, all the fields associated with a response do not appear in the entry. Thus, only the component that details the request's timestamps is guaranteed to be available. (Currently, the report contains only HTTP request/response pairs where the request was captured).

Timestamps are marked in blue and represent the number of seconds since January 1st, 1970. The handle to the response data is marked in orange. The response data itself can be retrieved by using the "Get Analysis Artifact" (5) API call.

```
{
  "type": "Waterfall report",
  "subtype": "Http Waterfall report",
  "version": "0.80",
  "subTransactions": [{
    "type": "HTTP request/response",
    "start": 1333054863953,
    "end": 1333054864640,
    "recomendations": "",
    "attributes": {
      "RequestContentSize": 0,
      "ResponseContentType": "application/json; charset\u003dUTF-8",
      "StatusCode": 401,
      "TcpReset": false,
      "Method": "POST",
      "Scheme": "https",
      "ResponseContentSize": 104,
      "RequestHeaders": "POST /setup/ws/1/validate HTTP/1.1\r\nHost:
setup.example.com\r\n nAccept-Encoding: gzip, deflate\r\nConnection: keep-
alive\r\nProxy-Connection: keep-alive\r\n",
      "TcpSession": 4,
      "RequestData": "",
      "RequestContentType": "text/plain",
      "URI": "/setup/ws/1/validate",
      "ResponseData":
      "\\74b85bbff75340a9b744bf8b4d1f5f6b\\-
```

```
1019702096\5d35c39d1db84f3fa16786dc78eff622\0703708ccbda491ba6d59944c1ef1114/7882  
0e83d8634265900999172f134389",
```

```
    "ResponseHeaders": "HTTP/1.1 401 Unauthorized\r\nDate: Thu, 29 Mar 2012 21:01  
GMT \r\nConnection: Keep-alive\r\n",
```

```
    "host": "setup.example.com",  
    "Referer": "https://www.example.com/"
```

```
  },
```

```
  "components": [{  
    "type": "DNSResolution",  
    "start": 1333054863853,  
    "end": 1333054863900
```

```
  },
```

```
  {
```

```
    "type": "TCPSetup",  
    "start": 1333054863900,  
    "end": 1333054863953
```

```
  },
```

```
  {
```

```
    "type": "ClientWaitAfterTCPSetup",  
    "start": 1333054863953,  
    "end": 1333054864418
```

```
  },
```

```
  {
```

```
    "type": "TLShandshake",  
    "start": 1333054864418,  
    "end": 1333054864632
```

```
  },
```

```
  {
```

```
    "type": "request",  
    "start": 1333054864632,  
    "end": 1333054864632
```

```
  },
```

```
  {
```

```
    "type": "wait",  
    "start": 1333054864632,  
    "end": 1333054864640
```

```
  },
```

```
  {
```

```
    "type": "response",  
    "start": 1333054864640,  
    "end": 1333054864640
```

```
  }
```

```
  ]},
```

```
{
```

```
  "type": "HTTPS session",  
  "start": 1333054861902,  
  "end": 1333054863281,  
  "recommendations": "",
```

```
    "attributes": {
      "SentBytes": 384,
      "ReceivedBytes": 5792,
      "host": "www.example.com",
      "TcpReset": false,
      "TcpSession": 0
    },
    "components": [{
      "type": "TCPSetup",
      "start": 1333054861902,
      "end": 1333054861902
    },
    {
      "type": "ClientWaitAfterTCPSetup",
      "start": 1333054861902,
      "end": 1333054862731
    },
    {
      "type": "TLSHandshake",
      "start": 1333054862731,
      "end": 1333054863230
    },
    {
      "type": "EncryptedDataTransmission",
      "start": 1333054863230,
      "end": 1333054863281
    }
  ]
}
--- OTHER HTTP and HTTPS ENTRIES ---
}]}
```

Structure of the Best Practices Analysis Report

The best practices report is quite simple. report is a list of entries, each representing a best practice; In the example below two best practices are highlighted in blue. Each best practice contains the following fields:

- **Name**
- **Description**
- **List of applicable scenarios** (currently DesktopWeb or MobileSafari)
- **Score**: which measures the how much the transaction follows the given best practice (a number in [0,1]),
- **Weight**: measures the impact of the best practice on the transaction (a number in [0,1]).
- **A dictionary of violations**: each entry in this dictionary is a specific type of violation on the best practice and a list of resources (or TCP sessions) that are committing that violation. Notice that a

transaction may not have any violation for a given best practice, as is the case with "Compress Components" below.

```
{
  "type": "Best Practices Report",
  "subtype": "Web Applications Best Practices Report",
  "version": "0.5",
  "report": [{
    "violations": {},
    "name": "Compress Components",
    "scenarios": [
      "DesktopWeb",
      "MobileSafari"
    ],
    "description": "Checks that textual elements are transferred in a compressed
format. Compression usually reduces the response size by about 70%. Approximately
90% of current Internet traffic travels through browsers that claim to support
gzip. ",
    "score": 100.0,
    "weight": 1.0
  },
  {
    "violations": {
      "An expiration header was not found": [
        "http://platform.example.com/widgets.js"
      ],
      "Expiration date is within the next two days": [
        "HTTP://media.example.com/media-proxy/picture1.jpg",
        "http://media.example.com/media-proxy/picture2.jpg",
        "http://media.example.com/media-proxy/picture3.jpg"
      ]
    },
    "name": "Add long term headers expiration dates",
    "scenarios": [
      "DesktopWeb",
      "MobileSafari"
    ],
    "description": "Near future headers expiration dates prevent effective caching.
This results in a repeat visit to your site to be slower than necessary.",
    "score": 65.0,
    "weight": 0.8
  }
]
```

Chapter 5: NV Analytics Protocols

This section provides details regarding how NV Analytics identifies and works with various protocols in the legacy reports, including:

- [Supported Protocols](#) 48
- [Conversation Definition](#) 48
- [Collecting Conversation Statistics](#) 48
- [Classification of TCP, UDP, IP](#) 49
- [Sub-Transaction Grouping](#) 49
- [Understanding Protocol Association](#) 49

Supported Protocols

The following protocols are supported and analyzed by NV Analytics.

Layer 2 - 3	Web
IP	HTTP
TCP	HTTPS
UDP	

Conversation Definition

The definition and identification of a conversation depends on the type of analysis being performed.

The definitions (identifications) are based on:

- IP - IP address pair (e.g. 10.0.0.1 - 10.0.0.2)
- UDP - IP address & port number pair (e.g. 10.0.0.1:6789 - 10.0.0.2:3456)
- TCP - IP address & port number pair (e.g. 10.0.0.1:6789 - 10.0.0.2:3456)
- HTTP - URL (e.g. www.google.com/images)

Collecting Conversation Statistics

NV Analytics collects statistics per conversation instance (e.g. a single Get of www.google.com URL).

Metrics shown will be for the following groupings:

- All applications
- Per application
- Per application conversation (including sub-conversation)

Classification of TCP, UDP, IP

A conversation is classified as TCP only if no higher level protocol is present. In this case it is identified as TCP Other.

A conversation is classified as UDP only if no higher level protocol is present. In this case it is identified as UDP Other.

A conversation is classified as IP only if there is no higher level protocol e.g. not TCP or UDP). In this case it is identified as IP Other.

Sub-Transaction Grouping

The NV Analytics groups sub-conversations into a single flow so that you can get data of the whole conversation or on each of the sub-conversations that it contains. How this grouping takes places is determined by how you configure NV Analytics for grouping.

Understanding Protocol Association

Conversations are associated based on the relevant application protocol. This means that if we're looking at HTTP then the underlying TCP / IP communication and communication metrics will be associated with the HTTP conversation they are part of. If we found TCP Retransmissions during a HTTP Get Request-Response conversation, these TCP Retransmissions will be associated with the HTTP conversation.

When a sequence number is received that is lower than expected (i.e. either a retransmission, a fast retransmission, or an out of order segment), NV Analytics assumes that it is a fast retransmission if:

- It has seen ≥ 2 duplicate ACKs for this segment (i.e. ≥ 3 ACKs).
- If this segment is the next un-ACKed segment.
- If this segment came within 20ms of the last duplicate ACK (20ms is arbitrary; it should be small enough to not be confused with a retransmission timeout).

Send Us Feedback



Can we make this User Guide better?

Tell us how: SW-Doc@hp.com