Trend Micro™ TippingPoint™

Digital Vaccine Toolkit User Guide

Privacy and Personal Data Collection Disclosure

Certain features available in Trend Micro products collect and send feedback regarding product usage and detection information to Trend Micro. Some of this data is considered personal in certain jurisdictions and under certain regulations. If you do not want Trend Micro to collect personal data, you must ensure that you disable the related features.

The following link outlines the types of data that the Security Management System collects and provides detailed instructions on how to disable the specific features that feedback the information.

https://success.trendmicro.com/data-collection-disclosure

Data collected by Trend Micro is subject to the conditions stated in the Trend Micro Privacy Policy:

https://www.trendmicro.com/en_us/about/legal/privacy-policy-product.html

Legal Notice

Publication: February 2023

Introduction to DV Toolkit

The Digital Vaccine Toolkit (DV Toolkit) application is designed to be used with the Trend Micro™ TippingPoint™ IPS System software and hardware, including the Local Security Manager (LSM) and the Security Management System (SMS). You use the DV Toolkit to create custom attack filters that you can use to detect and prevent network intrusions. You can integrate and use these filters in conjunction with the TippingPoint Digital Vaccine filters that are active in your system. You can use the toolkit to address specific security concerns by implementing your own proactive filter-based security to protect your corporate environment from attack.

This section includes the following topics:

Before using this tool

The TippingPoint DVLabs organization provides the complementary DV Toolkit to Trend customers who want to take a more customized approach to attack prevention. The DVT filter customization tool requires a certain level of expertise. The DV tools are designed for experienced users of TippingPoint systems who are experienced at writing security filters. As such, Trend assumes no responsibility for the filters that you develop.

If you have highly specialized needs for customized DV filters that require the services of DVLabs, contact your TippingPoint account representative for assistance. If you need assistance using the DV Toolkit or the DV converter tool (see *DV converter*), contact support. Refer to *Read Me First* to assist you with the installation and operation of the tools.

Make sure to read the *Digital Vaccine Toolkit Release Notes* before using this tool. To ensure that you have the latest versions of product documentation, visit the Online Help Center.

System requirements

The DV Toolkit and DV Converter tool have the following requirements:

Component	Requirements
Operating System	Microsoft Windows 7 (32-bit, 64-bit,) Microsoft Windows 8
Software platform	NET Framework 4 required. If not installed, refer to the Microsoft download site.
Memory	96 megabytes (MB) of RAM recommended
Hard Disk	• DV Toolkit application requires 10 MB • The standalone DV Converter application requires 10 MB
Display	800 x 600 or higher-resolution
Input Device	Microsoft mouse or compatible pointing device

Component	Requirements
Web Browsers	The DVT Help system has the following requirements: • Internet Explorer 7 or later • Mozilla Firefox 3 or later • Chrome

Installation instructions

The DV Toolkit application is available by electronic download from the Threat Management Center (TMC) website. The DV Toolkit is a .NET application and the .NET Framework is required to use the product. Before you install this product, ensure that the required version of the .NET Framework is installed on the system. It is typically recommended that no other application is running at the time of the installation.

For additional information about installing and starting the DV Toolkit application, see the *Digital Vaccine Toolkit Release Notes* available on the TMC. For information about installing the DV Converter tool, see *DV converter*.

Procedure

1. Log in to your system as an administrator.
2. Launch the DVT installation package that you downloaded from the TMC.

   The DVT installer program starts automatically.
3. Follow the on-screen instructions.

   Select the **I Agree** button and click **Next** to accept the license agreement.
4. Select the installation type to specify who can access the program, and click **Next**.
5. Accept the default location or browse to the destination on your hard disk where you want DV Toolkit to be installed and click **Next**.
6. Click the **Finish** button to complete the installation.

What to do next

Note

Do not use the Windows Add/Remove Programs function to remove the .NET Framework 4 Client while the DV Toolkit application is installed; the .NET Framework is required for proper operation of DV Toolkit and DV Converter.

Note

To uninstall the DV Toolkit application use the Add/Remove Programs function, or run the DV Toolkit installer and select the Remove Digital Vaccine Toolkit option. Removing the DV Toolkit application does not remove the .NET Framework.

Using the DV Toolkit

The DV Toolkit (DVT) can be used to write custom security filters for your TippingPoint IPS. It is not intended to be used with any other third-party network security software or hardware.

You can create custom security filters individually and grouped as a DVT package, which you can then use in the TippingPoint Local Security Manager (LSM) or the Security Management System (SMS) for intrusion prevention. A DVT package can include more than a hundred filters per package, with a limit of 8000 filters per package.

After you create a package you can install it on the LSM for a single IPS. If your installation includes an SMS that manages multiple IPS devices, you can import it in the SMS and then distributed it to any or all of the IPS devices. You can install multiple DVT packages, but only one can be active at any one time.

You can add security filters to a DVT package at any time by creating a new filter. You can import new or revised DVT packages into the LSM or SMS as required.

Note

The limit on package size is 8000 filters. If you are creating or importing more than 8000 filters, be sure to break them into multiple packages.

DV Toolkit and Digital Vaccine considerations

Before you create custom attack filters, check the TippingPoint Threat Management Center (TMC), or TippingPoint ThreatLinQ, to see if a Digital Vaccine filter already exists to prevent that type of intrusion.

If you have snort rules that you want to migrate to Digital Vaccine custom filters, determine if DV filters that accomplish the same objective are already defined. You can convert snort rule sets using the DV Converter tool and then determine which rules you want to incorporate in your DVT filter packages. For more information about DV Converter, see *DV converter*.

Helpful resources

A variety of publications and resources are available online to help with crafting regular expressions and deciphering snort rules. These resources may be useful when constructing DVT custom filters.

- Regex Coach — an interactive, graphical tool that you can use to experiment with creating regular expressions. Regex Coach is available from the cnet.com download site and installs on Windows or Linux.
- Regular-Expressions Information — the http://regular-expressions.info site provides tutorials and serves as a general regular expressions reference site.
- Snort Community — the snort community at snort.org/community links to user group and discussion forums that can be helpful if converting snort rules to use in DVT custom filters.

DV Toolkit interface

The Digital Vaccine Toolkit (DV Toolkit) consists of a main window, in which you can view and modify filter packages, a details window, in which you create and modify individual filters, and an options window, in which you can set various filter-related options. To view the properties for a filter package, use the Filter Package Properties window. To quickly launch the Details Window for a filter package, double-click a DVT package listed in the main window.

This section includes the following topics:

- *Main window*

Main window

You will see the DV Toolkit main window when you open the application. This window includes information about each of the filters in the package. Each time a filter is created and saved, information about it is added to this view. You can have only one filter package at a time open; however, you can add filters to the package you are viewing by using the Merge feature. The function of the menus and the icons on the menu bar are further described in *Menus and commands*, and *Toolbar*.

The following fields are part of the DV Toolkit main window

- **Filter:** Number automatically assigned by the DV Toolkit. Filters created with DV Toolkit start with the letter 'C'.
- **Name:** The name that you assign to the filter. The name may be up to 100 characters in length. The name and filter number will appear in the block and alert logs on the target system.
- **Severity:** Defines the alert level that will be applied to the results of the filter.
- **Protocol:** Defines the network protocol of the attack that the filter will recognize.
- **Last Modified:** Displays the date and time of the last change to the filter.

To view additional details about a filter, double-click the name of the filter to open the Details window where you can also edit the filter definition.

Menus and commands

The following table provides a description of the DV Toolkit menus and commands.

Menu Item	Selection	Hot Key	Description
File	New Package		Enables creation of a new DVT package.
	Open Package...	Ctrl + O	Opens an existing package
	Merge Packages...		Merges an existing DVT filter package, or selected filters from a DVT filter package, into the filter package you have open in DV Toolkit.
	Import XML...		Imports a DVT conversion file (an `.xml` formatted file)
	Save Package	Ctrl + S	Saves the filter package
	Save Package As...		Saves the package with changed name or location
	Properties		View information about the package properties. See *Filter package properties*.
	Exit	Alt + F4	Exits the application
Edit	Cut	Ctrl + X	Cuts the selected filters
	Copy	Ctrl + C	Copies the selected filters
	Paste	Ctrl + V	Pastes the filters from the clipboard

Menu Item	Selection	Hot Key	Description
	Select All	Ctrl + A	Selects all items in the list view.
Filter	New...	Ctrl + N	Opens the *Details window* to allow you to create a new custom filter
	Edit...		Opens the *Details window* to allow you to edit the selected filters
	Delete		Deletes the selected filters
	Options...		Opens the *Options window* to allow you to enable trigger constraints to provide auto-trigger protection when defining payload search strings. See *Filter trigger constraints*.
Help	Users Guide		Opens an HTML version of the users guide
	About DV Toolkit		Displays the Version number and TippingPoint contact information

Toolbar

You can use the DV Toolkit toolbar to perform DVT functions and to search (see *Search*) for filter packages. The following functions are available through the toolbar:

- New Package
- Open Package
- Save Package
- Merge Packages
- Import XML
- Cut
- Copy
- Paste
- New Filter
- Edit Filter
- Delete Filter
- Renumber Shield

See the *Menus and commands* table for more information about each button's function.

Search

Use the Search field to search the **Name** fields of the filter package for a specific name or string of characters in a name. The search operation is an incremental search, which begins searching the package as soon as you type a character. Each added character extends the search to the next string that matches.

The search is case-insensitive until you enter an uppercase character in the search box. Any uppercase character makes the search case-sensitive and only results that match the entered case will be returned.

Details window

This section describes the tabs and fields located in the DV Toolkit Details window. The Details window opens when you *edit or create a new custom filter or filter package*. The following tabs in the Details window are common to all filter types:

- *General*
- *Properties*
- *Triggers*
- *Payload*

The following tabs may or may not be displayed, depending on the type of filter that you are creating, which is based on the protocols defined for the filter pattern matching.

- *IPv4*
- *IPv6*
- *ICMP*
- *TCP*
- *UDP*
- *HTTP*

General

The General tab is the default view in the Details window. The **General** tab includes the following information about the filter:

- **Filter** — A number automatically assigned by the DV Toolkit. Filters created with DV Toolkit start with the letter 'C'.
- **Name** — The name assigned to the filter. You can specify the name when you define or edit a filter. The name may be up to 72 characters in length. For a converted Snort rule, the name is the converted value of the Snort **msg** attribute. The name and filter number will appear in the block and alert logs on the target system.
- **Author** — The name of the user who created the filter. Appears in filter details screen on target system.
- **Description** — A description of the filter. The description is used by the LSM and the SMS applications. The description can be up to 1,999 characters long.

Properties

The Properties tab is where you specify the properties that you want to define for a custom filter. These are attributes that characterize the filter type. The following illustration shows the Properties tab in the Details window:

The **Properties** that you can define for a Digital Vaccine filter include the following:

- **Protocol** — The network protocol that will be matched by the filter. For example, selecting the TCP protocol enables the filter to match TCP source and destination ports that you specify on the TCP tab. The following protocols are supported:

  - IPv4 — Inspects IP version 4 packets for matches.
  - ICMPv4 — Matches ICMP protocol in IP version 4 packets only.
  - UDP — Matches UDP protocol in IP traffic for the version selected (either IPv4 or IPv6).
  - TCP — Matches TCP protocol in IP traffic for the version selected (either IPv4 or IPv6).
  - HTTP — Matches HTTP protocol in IP traffic for the version selected (either IPv4 or IPv6).
  - IPv6 — Inspects IP version 6 packets for matches.
  - ICMPv6 — Matches ICMP protocol in IP version 6 packets only.
- **Severity** — The alert level that will be applied to the results of the filter. The alert level appears in the LSM or SMS and helps you prioritize the threat for mitigation. You can select from **Critical**, **Major**, **Minor**, or **Low**.
- **Category** — The DV filter category to which this filter belongs. Categories allow multiple filters to be controlled together on the LSM or SMS through the management of category settings. Categories are used to enforce global settings for a category group of filters and are also used to locate filters in the LSM. For information about how to use category settings, refer to the LSM or the SMS documentation. Categories include:
  - **Exploits** – Filters that target specific attack programs or data.
  - **Identity Theft** – Filters that target phishing and social engineering attacks.
  - **Reconnaissance** – Detection of host and port scanning, and similar information-gathering programs.
  - **Security Policy** – Policy-based controls for benign events.
  - **Spyware** – Filters that block programs that steal information or intimidate victims.
  - **Virus** – Filters that block destructive virus programs.
  - **Vulnerabilities** – Filters that target all forms of methods to exercise vulnerabilities.
  - **Network Equipment** – Filters that protect network and infrastructure devices.
  - **Traffic Normalization** – Filters used to regulate and manage network bandwidth.
  - **IM** – Filters that target Instant Message and Social Networking programs.
  - **P2P** – Filters that target Peer-to-Peer programs and communication.
  - **Streaming Media** – Filters that are typically used to rate limit or block bandwidth-intensive streaming data.
- **Class** – Describes the kind of attack that the filter recognizes. This is an arbitrary designation that you can use if you choose to specify a classification for the threat. You can select one of the following classes:
  - Access
  - DoS (denial of service) - see also DDoS filters
  - Failure
  - Informational
  - Policy
  - Suspicious

  - Trojan
  - Virus
  - Worm
  - Exploit
  - P2P
  - Vulnerability
- **IP Version** — Indicates if the filter applies to IP version 4 or IP version 6 protocol for traffic inspection. You can specify either option for filters targeting UDP, TCP, HTTP protocols.
- **Attacker Address** — Specifies the IP address from which the attack originates. If an attack originates from the destination IP address rather than the source, select Destination IP.

Triggers

If you disable trigger-less filter functionality in the Filter Options window, you can use this tab to view and set the following properties:

- **Depth** — Specifies how deep into the packet the filter will search for a pattern match of the content string or strings. Specified as a value of -1 or a number of bytes from 1 to 65535. If used with **Offset**, this value is offset by the value specified for **Offset**.
- **Offset** —Specifies at what point in the packet the filter begins searching for a pattern match. Specified as a number of bytes from -65535 to 65535. The count starts at zero. For example, if 3 is specified for the value, the filter looks for a match beginning at the fourth byte character.
- **String Search** — Displays and allows you to set the String Search value. At least one string is required and must contain at least four characters (bytes) and must not exceed 12 characters.

You can reorder the list by clicking on any of the column headers. For example, if you click on the `Depth` column header, the list is reordered based on the values in the `Depth` column.

Note

By default, DV Toolkit displays a warning message if you enter a string in the String Search field that contains a trigger-constrained word or phrase. For more information about auto-trigger protection, see *Filter trigger constraints*.

Payload

The **Payload** tab contains string search information and regular expression syntax defined for the filter. The String Search data is required for the filter and determines which traffic packets will be inspected. You can define one or more strings for the payload String Search pattern and apply Boolean operators to specify how each string (or test block) is applied in the search pattern. Use the `Add` and `Delete` buttons to add and delete string searches or regular expressions.

The regular expression is optional to provide further pattern matching after the fixed string has identified a match. Layers 5-7 of a packet are searched for the Payload parameters. You can add regular expressions to your customized filters to add greater depth to the search.

Multiple regular expressions can be defined, each one building upon the next to finely tune the filter's pattern matching capability.

Note

By default, DV Toolkit displays a warning message if you enter a string in the String Search field that contains a trigger-constrained word or phrase. For more information about auto-trigger protection, see *Filter trigger constraints*.

The following parameters are applicable when defining the payload string search:

- **String Search**. This is a literal string that the filter uses to determine if the filter applies to a packet. The string, or multiple strings if specified, identify an attack or probable indication of an attack when it is found within the packet payload. When multiple strings are specified, a match is successful if any one of the string patterns or a combination of the string patterns, depending on the Boolean operator applied to the string, is detected. Binary characters can be encoded by enclosing the hex value in vertical bars. For example, the code string `|7F 7F|` searches for two consecutive DEL characters (ASCII code 0x7F). Use the + or – boxes to add or delete string.
  - **Test Block** — Specifies which Boolean operator (AND or OR) to apply when adding a string to the string pattern. AND is the default.
  - **Strip Newlines** — When selected, this option ignores newlines during a content string search.
  - **Case-sensitive** — When selected, this option enables case-sensitivity when searching for the fixed content string within candidate packets. However, some protocols (e.g., SMTP) have case-insensitive control messages. Disable the option to match content search string regardless of character case.
  - **Offset** —Specifies at what point in the packet the filter begins searching for a pattern match. Specified as a number of bytes from -65535 to 65535. The count starts at zero. For example, if 3 is specified for the value, the filter looks for a match beginning at the fourth byte character.
  - **Depth** — Specifies how deep into the packet the filter will search for a pattern match of the content string or strings. Specified as a value of -1 or a number of bytes from 1 to 65535. If used with **Offset**, this value is offset by the value specified for **Offset**.

The following parameters are applicable when defining a regular expression for the payload:

- **Regular Expression**. Use a regular expression to refine your search target. Regular expressions are optional. The DV Toolkit supports the Perl Compatible Regular Expressions (PCRE) syntax for the specification of the regular expression information. The DV Toolkit will verify that the regular expression is valid before packaging the filter. Binary characters can be encoded by preceding the hex character code with the `\x` escape sequence. For example, the expression `\x7F(\x00)*\x7F` matches a sequence of two DEL characters (ASCII code 0x7F) separated by zero or more NUL characters (ASCII code 0x00). Select Regular Expressions in the Payload tree view to add or delete expressions. Expand the selection to view all defined expressions.
  - **Test Block** — Specifies which Boolean operator (AND or OR) to apply when adding a regular expression. AND is the default.
  - **Strip Newlines** — Enables the stripnewline primitive for a regular expression.
  - **Case-sensitive** — Select this option to specify that case-sensitivity is enabled for the regular expression defined for the filter. When unchecked, the case is ignored for pattern matching.
  - **Offset** —Specifies at what point in the packet the filter begins applying the regular expression. Specified as a number of bytes from -65535 to 65535. The count starts at zero.
  - **Depth** — Specifies how deep into the packet the filter applies the regular expression. Specified as a value of -1 or a number of bytes from 1 to 65535. If used with **Offset**, this value is offset by the value specified for **Offset**.

IPv4

The **IPv4** tab enables you to specify the Source and Destination IPv4 addresses for the filter, when you select IP Version 4 for the filter properties.

Clicking the **Import Addresses** button allows you to import source and destination IP addresses from a CSV file. The addresses from the chosen file populate the `Source Address` and `Destination Address` text fields.

If the source and destination address fields are left blank and the protocol specified is other than IP or IP6, the filter applies to traffic coming from and going to any and all source and destination IP addresses.

Note

When the Protocol is set to IP or IP6 on the Properties tab (see *Properties*), you must enter an IP source address or destination address, or both. If you leave both fields blank or specify both as "any," an error message is displayed. For more information about filtering for IPv6 traffic, see *IPv6*.

Multiple addresses may be listed separated by commas, and addresses may be represented in CIDR notation (`201.10.20.0/24` matches only the first 24 bits of the address). You can also negate a source or destination address with the **!** symbol. For example if you enter `!201.10.20.0` in the `Source Address` field, that address is negated.

You can specify values for various elements in the IP packet that the filter will match, or accept the default option to ignore them. The range of valid values for each option are specified in parenthesis.

- **Protocol** (0-255) — Specifies a Protocol value that the filter will match. For example, the protocol value for ICMP for IPv6 is 58.
- **Header Length** (20-60) — Specifies a value for the IP header length the filter matches.
- **Type of Service** (0-255) — Specifies the type of service (or Differentiated Services) value the filter matches.
- **Identification** (0-65535) — Specifies a value for the header ID the filter matches.
- **Offset** (0-8191) — Specifies an offset value in the packet header the filter matches.
- **Time to Live** (0-255) — Specifies the TTL value in the packet header the filter matches.
- **Data Length** (0-65535) — Specifies the data length value in the packet header the filter matches.

You can also set IP flag and fragment options if you want to filter for IP fragmentation attacks.

- **IP Flags** — The following flag options can be set to **On** (checked), **Off** (unchecked), or **Ignore** (dimmed checkmark) to match flag states in the IP header. Ignore is the default setting.
  - Reserved
  - Don’t Fragment
  - More Fragments
- **Fragment** —This option specifies if the filter only matches on fragmented packages.

IPv6

If you specify IP version 6 for the filter Properties, the IPv6 tab enables you to specify the Source and Destination IPv6 addresses for the filter.

Source Address Clicking the **Import Addresses** button allows you to import source and destination IP addresses from a CSV file. The addresses from the chosen file populate the **Source Address** and **Destination Address** text fields.

When IP6 is selected for the Protocol on the Properties tab (see *Properties*), an IP source address or destination address, or both must be entered on the IPv6 tab. If both fields are blank or specified as "any," an error message is displayed.

You can specify to ignore or match the following IP parameters for filters targeting IPv6 packets:

Multiple addresses may be listed separated by commas, and addresses may be represented in CIDR notation (`201.10.20.0/24` matches only the first 24 bits of the address). You can also negate a source or destination address with the **!** symbol. For example if you enter `!201.10.20.0` in the **Source Address** field, that address is negated.

- **Protocol** — Specifies the Protocol value, from 0-255, in the IP packet header that the filter matches.
- **Traffic Class** — Specifies the Traffic Class value, from 0-255, in the IP packet header that the filter matches.
- **Payload Length** — Specifies the Payload Length value, from 0-65535, that the filter matches.
- **Hop Limit** —Specifies the Hop Limit value in an IPv6 packet that the filter matches (similar to the Time to Live value in an IPv4 packet).
- **Offset** — Specifies the offset value, from 0-8191 (if fragmentation occurs), that the filter matches. Consider that the first fragment has a value of 0.
- **Identification** — Specifies the packet ID for the datagram (used if fragmentation occurs) that the filter matches.
- **Next Header** — Specifies the next header value that the filter matches.
- **Fragment** — When checked, specifies that the filter matches only on fragmented packets.

If you want to target only IPv4 addresses, see *IPv4* for information about specifying IPv4 parameters for the filter.

ICMP

The **ICMP** tab enables you to specify the ICMP type and code information for ICMP messages that the filter will match. The options displayed on the ICMP tab differ depending on which IP version is selected (version 4 or version 6) for filtering of the IP addresses.

Note

If the source and destination addresses are not specified for the IP or the IPv6 addresses, the filter applies to traffic coming from and going to any and all source and destination IP addresses for the version specified in the filter properties.

The following topics, *ICMPv4* and the *ICMPv6*, describe the parameters that can be defined for filters.

ICMPv4

When ICMP is selected as the protocol for the filter properties, DV Toolkit automatically sets the IP version as version 4. If you want to specify the IPv4 addresses to target for filtering the ICMP messages, enter the addresses on the IP tab. See *IPv4* for more information.

You can specify the following ICMP parameters for IP version 4 packets.

- **Type** — Defines the Type field that the filter matches in an ICMP packet. The field may have an integer value from 0 to 255. For example, a ping request packet (ICMP ECHO REQUEST) has type 8, while a ping reply packet has type 0. Choose from the following options to specify a value:
  - **Ignore** — Ignore the Type field. The filter matches any ICMP type. This is the default selection.
  - **Equal** — Specifies the Type number that the filter must match exactly in the ICMP packet.
  - **Not equal** — Specifies a Type number for which any other type number is matched by the filter.
  - **Greater than** — Specifies a Type number for which any greater number is matched by the filter.
  - **Less than** — Specifies a Type number for which any smaller number is matched by the filter.
  - **Range** — Specifies a Type number range for which any value in the range is matched by the filter.
- **Code** — Specifies the Code type field that the filter matches in an ICMP packet. A valid code value is dependent on the value specified for the Type. ICMP Codes are subtypes of an ICMP Type. An ICMP Type may or may not have an associated Code type. This field may have an integer value from 0 to 255. For example, Time Exceeded has Type 11 and may also have a Code value of 0 or 1.
- **ID** — Specifies the ICMP message ID that the filter matches. Valid values are from 0-65535. Ignore, Equal, or Not Equal are valid options.
- **Sequence** — Specifies the ICMP packet sequence number that the filter matches. Valid values are from 0-65535. Ignore, Equal, or Not Equal are valid options.
- **Data Length** — Specifies the Data Length value in the ICMP packet that the filter matches. Valid values are from 0-65535. Ignore, Equal, Not Equal, Greater than, or Less than are valid options.

ICMPv6

The following table describes the ICMP parameters that you can specify for IP version 6.

- **Type** — Specifies the Type field that the filter matches in an ICMPv6 packet. The field may have an integer value from 0 to 255. For example, an ICMPv6 ping request packet (ECHO REQUEST) can have type 128, while an ECHO REPLY has type 129. Specify one of the following options to designate the type number. The format for this field is `<min>:<max>`.
- **Code** — Defines the Code type field that the filter matches in an ICMP packet. A valid code value is dependent on the value specified for the Type. For example, Time Exceeded (Type 11) and may have a code value of 0 or 1. ICMP Codes are subtypes of an ICMP Type. An ICMP Type may or may not have an associated Code type. This field may have an integer value from 0 to 255. The format for this field is `[! | > | < ] <number>`.
- **Echo ID** — Specifies the Echo ID value in the packet that the filter matches. Valid values are from 0-255.
- **Echo Sequence** — Specifies the Echo Sequence value in the packet that the filter matches.
- **Valid Checksum** — When checked, specifies that the filter matches if the checksum is valid.

TCP

The **TCP** tab enables you to specify the source and destination TCP ports that you want to monitor for attacks.

Note

You must enter a value for either Source or Destination, or both. You cannot leave both Source and Destination blank.

- **Port** — You can select a predefined port service by name, or you can select **Port List** or **Port Range** to specify port numbers explicitly. The following table lists the available predefined TCP port services.

Port Service	Port Numbers
(blank)	none (This is the default selection.)
ftp	21
http	80, 3128, 8000, 8080
imap	143
ircu	6665, 6666, 6667, 6668, 6669, 7000
ms-sql	1433
nntp	119
pop3	110
portmapper	111, 32770-32779
rlogin	513
rsh	514
smb	139, 445
smtp	25
snmp	161
ssh	22
telnet	23
dns	53

- **Port List** — Enter a comma-separated list of TCP source ports. For example, to monitor ports 37, 42, and 53, enter `37, 42, 53`. This option is enabled when the source port is set to "Port List."
- **Port Range** — Enter a range of numbers by entering the beginning number and the ending number. For example, to monitor all ports from 36 to 39, inclusive, enter `36` in the first box and `39` in the second box. This option is enabled when the source port is set to “Port Range.”
- **TCP Header Flags** — The filter will be applied to TCP traffic containing any of the header flags that you select. Each flag can be set to On, Off, or Ignore (default option is Ignore).
- **Established** — The filter will be applied to TCP traffic containing the ESTABLISHED keyword.
- **TCP Window Size** — The filter can ignore the TCP window size, or can match a TCP window size of a value equal to, not equal to, greater than, or less than the specified value. Enter a value in bytes.
- **Header Length** —The filter can ignore the header length, or match the header length value that is equal to, not equal to, greater than, or less than the specified value.
- **Data Length** —The filter can ignore data length, or match the data length value that is equal to, not equal to, greater than, or less than the specified value. You can also specify a value range.
- **Acknowledgment Number** — The filter can ignore the acknowledgment number (the next sequence number the sender is expecting to receive), or match the acknowledgment number specified.

- **Sequence Number** — The filter can ignore the sequence number in the TCP packet or match the sequence number that is equal to, not equal to, greater than, or less than the value specified.

UDP

The UDP tab is available in the Details window. The **UDP** tab enables you to specify the source and destination UDP ports that you want to monitor for attacks.

Note

You must enter a value for either Source or Destination, or both. You cannot leave both Source and Destination blank.

- **Port** — You can select a predefined port service by name, or you can select **Port List** or **Port Range** to specify port numbers explicitly. The following table lists the available predefined UDP port services.

Part Service	Port Numbers
(blank)	none (This is the default selection.)
dns	53
portmapper	111, 32770-32779
snmp	161

- **Port List** — Enter a comma-separated list of UDP source ports. For example, to monitor ports 37, 42, and 53, enter `37, 42, 53`. This option is enabled when the source port is set to "Port List".
- **Port Range** — Enter a range of numbers by entering the beginning number and the ending number. For example, to monitor all ports from 36 to 39, enter `36` in the first box and `39` in the second box. This option is enabled when the source port is set to “Port Range.”

HTTP

On the HTTP tab you can specify information that will inspect HTTP requests for potential threats.

The **HTTP** tab on *the Details window* enables you to specify Uniform Resource Information (URI) for the filter search criteria. Many attacks against web servers have a characteristic string in the URI portion of the request. The DV Toolkit can perform a string or regular expression search of the HTTP URI. You can specify which part of an HTTP request the filter inspects by selecting one of the following options: **URI**, **URI Path**, **URI target**, **Header**,or **Param**.

The DV Toolkit supports the Perl Compatible Regular Expressions (PCRE) syntax for the specification of the regular expression information. The DV Toolkit will verify that the regular expression is valid before packaging the filter. For regular expression patterns, consider that the DV Toolkit HTTP decoder sanitizes URLs into a normalized form usable by the decoder prior to HTTP inspection. This helps prevent filter evasions and saves on CPU performance when running regular expressions.

The following options are available for specifying HTTP filters:

- **Method** — Defines the HTTP request method. Most HTTP requests use either the GET or POST method. Other methods include: OPTIONS, PUT, DELETE, TRACE, CONNECT, HEAD and the extended methods for distributed authoring (WebDAV): LOCK, UNLOCK, PROPFIND, PROPPATCH, MKCOL.
- **URI Length** — Defines the length of the URI for the filter between 0 to 65535. Set the values in the associated text field. The drop-down menu allows you to define the length based on a range, a specific value, and so on; the choices available are:

- **Ignore**
- **Equal**
- **Not Equal**
- **Greater than**
- **Less than**
- **Range**

The default value is `Ignore`.

- **String Search** — Specifies a string that matches a substring of the HTTP request's URI for which the filter searches. Binary characters can be encoded in the URI. If the URI is not a regular expression pattern, then enclose the hex value in vertical bars. For example, the code string `|7F 7F|` searches for two consecutive DEL characters (ASCII code 0x7F). For the string search, you can specify the Offset value and Depth:
  - **Offset** — Specifies at which point in the HTTP request packet the filter begins searching.
  - **Depth** — Specifies how deep into the packet the filter searches for a match.
- **Regular Expression** — Specifies a regular expression in the HTTP request for which the filter will search. Binary characters can be encoded by preceding the hex character code with the `\x` escape sequence. For example, the expression `\x7F(\x00)*\x7F` matches a sequence of two DEL characters (ASCII code 0x7F) separated by zero or more NUL characters (ASCII code 0x00).

  When constructing a regular expression for HTTP inspection, consider the HTTP decoder pre-processing that occurs prior to HTTP inspection of the regular expression. For example, a script embedded in a standard URL encoding resembles the following:

  ```
  /index.html?a=foobar%3cscript%3ealert%28%27foo%27%28%3c%2fscript%3e
  ```

  A cross-site scripting attack might appear as some variation of this encoding, such as:

  ```
  /index.html?a=foobar%u003csCripT>alErt(‘foo’)<%2fscrIpT%3e
  ```

  After the HTTP decoder has sanitized the URI into a normalized form, the resulting script looks like the following:

  ```
  /index.html?a=foobar<script>alert(‘foo’)</script>
  ```

  An example of a regular expression for a filter using the URI function that would detect a match for the “script” tag embedded in a URL is as follows:

  ```
  /(index\.html)?\?[^\s\n]*<script
  ```

  Notice that the regular expression does not need to take into account the multitude of encoding variations.
- String Search or Regular Expression inspection options:
  - **URI** — Specifies a string that matches a substring of the HTTP request's URI. The URI appears after the method name in the HTTP request stream. Many attacks against web servers have a characteristic signature in the URI portion of the request, which makes it easy to identify an attack using this feature. You can encode binary characters and regex patterns. If the string is not a regex pattern, enclose the hex value in vertical bars (e.g., the string |7F 7F| matches two consecutive DEL characters). If the header is a regex, precede the hex value with a "\x" escape sequence (e.g., "\x7F(\x00)+\x7F" matches two DEL characters separated by one or more NULL characters

- **URI Path** — Specifies a string that inspects a substring in the URI path of an HTTP request, beginning at the first slash and ending at the last slash.
- **URI Target** — Specifies a string that inspects a substring of the HTTP URI, starting from the last directory slash to the first parameter or end of the URI. For example, in the HTTP request:

  ```
  GET /directory/foo.php?a=1&b=2 HTTP/1.1
  ```

  The URI Target is 'foo.php'. You can encode binary characters and regex patterns. If the string is not a regex pattern, enclose the hex value in vertical bars (e.g., the string |7F 7F| matches two consecutive DEL characters). If the header is a regex, precede the hex value with a "\x" escape sequence (e.g., "\x7F(\x00)+\x7F" matches two DEL characters separated by one or more NULL characters).
- **Header** — Specifies a string that matches a substring of the HTTP header. You can encode binary characters and regex patterns. Apply the same guidelines as described for the previous options for encoding binary characters and regex patterns.
- **Param** — Specifies a string that inspects the parameters of a HTTP request, including all parameters in both GET and POST requests. You can encode binary characters and regex patterns.
- **Payload** — Specifies the portion of the request after the header lines.

- **Strip Newlines** — When selected, this option ignores newlines during a content string search.
- **Case-sensitive** — Select this option to specify that case-sensitivity is enabled for the string or expression entered for the URI inspection.

Options window

This section describes the tabs and fields located in the DV Toolkit Options window. You can open the Options window by selecting **Filter > Options...** in the menu bar.

The following tabs are available in the Options window:

- *Payload*: Allows you to set auto-trigger enforcement levels and enable or disable trigger-less filters
- *Set Filter Numbers*: Allows you to set and renumber filters
- *Filter*: Allows you to set interfield constraints

Payload

The **Payload** tab of the Filter Options window allows you to choose between the following three **String Search Constraints**:

- **Enforce auto-trigger protection constraints**
- **Disable auto-trigger protection constraints**
- **Warn user of auto-trigger protection constraints**

Depending on which radio button you select, you receive the appropriate warning about constraints when manipulating a filter's triggers.

You can also use the **Payload** tab to enable and disable trigger constraint functionality by clicking in the **Allow trigger-less filter** checkbox. You can manipulate the variable values associated with triggers through the new **Triggers** tab in the *Filter Details* window.

Set filter numbers

Modify your filter numbering on the **Set Filter Numbers** tab by setting the following:

- **Reset Filter Numbers** radio buttons — Allows you to choose between **Re-number all filters** and **Re-number selected filters**.
- **Start filter number at** text field — Allows you to designate at which filter number your re-numbering begins. The default start point is filter number 10000. For example, if you choose 1, your filters are renumbered starting at 100001.

> **Note**
>
> If you are renumbering and an existing filter is encountered in your renumbering range, that filter number is skipped and the next available number is used. For example, if you designated filters numbered 991, 992, and 993 to be renumbered starting at 100001 and filter number 100002 already exists, the filters would be renumbered 100001, 100003, and 100004.

- **Preview** area — Displays a preview of the actions you have chosen. These changes are not applied until you click either the **OK** or **Apply** button.

Filter

The **Filter** tab allows you to choose between the following inter-field constraints:

- **Enforce interfield protection** — Enforces the existing inter-field protection and generates the appropriate warning message
- **Disable interfield protection** — Disables the inter-field protection and generates the appropriate warning message
- **Warn user of interfield protection** — Generates a warning notifying the user of the level of inter-field protection, but makes no changes to the default level set in the filter

Filter package properties

Use the **File > Properties** menu selection to open and view the Filter Package Properties window. The Properties window shows general information about the filter package and also provides the capability to enter a longer descriptive name for the package.

The Properties window shows the following information:

- **Name:** User-defined name for the filter package. This is the package name that appears in the LSM and the SMS.
- **Location:** Location of the filter package on your hard drive
- **Size:** Size of the package in bytes.
- **Modified:** Date and time the filter package was modified
- **Build:** Build number of the filter package. This number increments each time the package is saved
- **Filters:** Number of filters in the package
- **Author:** Creator of the package
- **Created In:** The DV Toolkit version number in which the package was created.

Filter trigger constraints

DV Toolkit provides automatic-trigger protection by validating payload strings defined for a filter against a set of trigger constraints. The constraints are meant to identify commonly expected network traffic substrings (such as `get` and `http`) that you might typically avoid using in a filter payload definition search string. By default, DV Toolkit warns you if you enter a restricted word or phrase when defining filters.

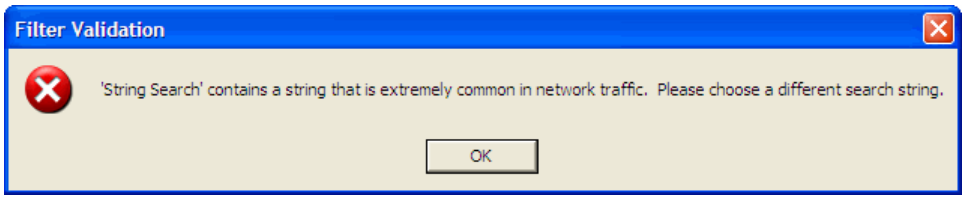
You can manage the auto-trigger protection feature by setting auto-trigger protection options (see *Setting auto-trigger protection options*):

The following options are available to configure how the trigger constraints are applied:

- *Enforce auto-trigger protection constraints*
- *Disable auto-trigger protection constraints*
- *Warn user of auto-trigger protection constraints*

Enforce auto-trigger protection constraints

When this option is selected, the trigger constraints are applied and DV Toolkit enforces the constraints by displaying the following error message if the string search entered on the payload tab contains a substring that constitutes a “bad” trigger. The error prompts you to remove the bad substring and prevents you from saving the filter with the constrained substring.

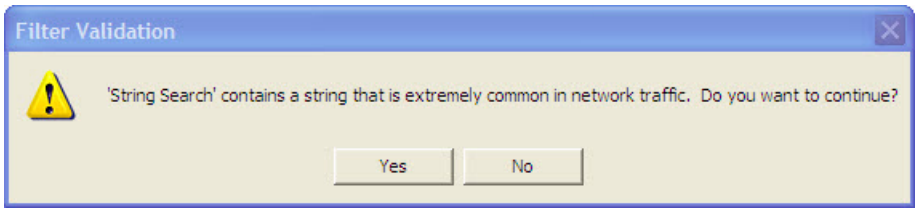


Disable auto-trigger protection constraints

This option provides no auto-trigger protection for filters defined in DV Toolkit. When this option is selected, the trigger constraints are ignored. Thus, the potential is greater that users creating filters might define payload strings that contain bad substrings.

Warn user of auto-trigger protection constraints

When this option is selected, the DV Toolkit application checks the trigger constraints and warns you with the following message if the payload search string you enter for the filter contains a potentially bad substring. You can choose to dismiss the warning and save the filter (click **Yes**), or cancel the operation (click **No**) which allows you to change or re-enter the string. This option is pre-set as the default option for auto-trigger protection.



Setting auto-trigger protection options

Describes how to set auto-trigger protection options.

To specify auto-trigger protection options, use the following steps:

Procedure

1. In the DV Toolkit main window, select **Filter > Options**.
2. Select one of the following available options to enforce, disable, or warn of the trigger constraints:
   - *Enforce auto-trigger protection constraints*
   - *Disable auto-trigger protection constraints*
   - *Warn user of auto-trigger protection constraints*
3. Click **Apply** to immediately apply the option to defined DVT filters.
4. Click **OK** to set the default option.

Custom DV filters

Use the Digital Vaccine Toolkit (DV Toolkit) to create custom filters and filter packages that you deploy on your TippingPoint systems. The custom package can be on a single TippingPoint IPS device managed with the LSM or distributed to multiple IPS devices by using the TippingPoint Security Management System (SMS).

This section discusses the following topics:

- *When to write custom filters*
- *Example scenarios*
- *Creating a custom filter*
- *Merging DVT filter packages*
- *Using snort rules in custom filters*
- *Distributing custom filters*

When to write custom filters

Before you write a custom filter, you should analyze attack reports, consider your system constraints, and decide which parts of your network you want to protect. You should also check the Threat Management Center web site and TippingPoint ThreatLinQ (http://threatlinq.tippingpoint.com) to see if there is a Digital Vaccine filter package already available that meets your needs.

If your current network security strategy relies on the use of Snort filters for threat prevention, consider if you want to convert the Snort rules to Digital Vaccine format for use in DVT filter packages. You can convert entire rule sets or selected rules that you want to incorporate into Digital Vaccine packages using the Digital Vaccine Converter tool. For more information, see *Using snort rules in custom filters*.

Assess the need to protect any in-house applications. If your organization is using custom software applications with known vulnerabilities, you can work with your development team to create custom filters to prevent damage from attacks. Use care when defining your filters that you don't include words or phrases in

the payload search string that will trigger alerts needlessly for non-malicious activity. Consider using DV Toolkit trigger constraints to enforce auto-trigger protection. See *Filter trigger constraints* for details.

The sample filters in *DVT sample filters* of this user guide provide a broader understanding of the DVT filter writing process.

Example scenarios

Often, organizations find that they need to block or rate limit access to a particular website. In the following examples, we look at two filters that address security issues such as these. These filters are part of the Digital Vaccine standard package and are explained here to help you understand how these types of filters are defined.

- *Blocking access to a web site*
- *Blocking a data stream*

Blocking access to a web site

Digital Vaccine filter 4624, available in the standard Digital Vaccine package, blocks all access to youtube.com. In the following discussion, we look at this HTTP filter.

Specifying a payload search string

The String Search expression in a filter enables the IPS to determine which filters to apply in the inspection. Because many websites rely on the "Host" header of an HTTP request to know which website to display (see RFC 2616, 14.23), this is considered a reliable detection point. For this filter, we apply this principal and use the following String Search expressions.

```
youtube.com|0a|
youtube.com|0d|
```

Explanation

- The pipe "|" characters represent that these are byte values.
- We include the |0a| and |0d| at the end of the string for the following reasons:
  - **Performance** – Helps minimize the traffic load for inspection by limiting the amount of data or web pages that the filter inspects. For example, the filter would not run on a thousand web pages with "Check out my youtube.com page!", which is not the intended inspection target; a search string of simply "youtube.com" would send up much more traffic and hinder performance.
  - **False Negative** – Most browsers will terminate an HTTP Header with a |0d 0a| (Carriage Return and Line Feed). However, only the |0a| is required. A simple evasion to a search string of "youtube.com|0d 0a|" is to simply remove the Carriage Return byte or supply multiples before the Line Feed, as in either of the following examples:
    - "Host: www.youtube.com|0a|" or
    - "Host: www.youtube.com|0d 0d 0d 0d 0d 0d 0d 0d 0d 0d 0a|"

    While it appears odd, it is a valid HTTP request and would likely be accepted by the remote server.

Note

For best results, do not include “www” in the search string because it may or may not be included in the header. Many websites simply accept “Host: youtube.com” and redirect users to the “www.youtube.com” site.

Refining the search with a regular expression

After specifying the fixed string to search for, we add a regular expression to refine the match. After specifying HTTP as the target protocol, we can then specify HTTP decoding information. For purposes of this filter, we specify a header regular expression that will be matched only in the HTTP header.

Let’s assume for this example that the data being inspected resembles the following:

```
User-Agent: Mozilla/5.0 (Windows; U; Windows
        NT 5.1; en-us; rv:1.8.0.4) Gecko/20060508Firefox/1.5.0.4|0d 0a|Host:
        www.youtube.com|0d 0a|
        Accept:text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;
        q=0.8,image/png,*/*;q=0.5|0d 0a|Accept-Language: en-us,en;q=0.5|0d 0a|
        Accept-Encoding: gzip,deflate|0d 0a|Accept-Charset: ISO-8859-1,utf-8;q=0.7,*,
        q=0.7|0d 0a|Keep-Alive: 300|0d 0a|Connection: keep-alive|0d 0a|
```

As specified in the HTTP RFC, headers may come in any order after the HTTP URI. Therefore, you want to exercise care in matching the correct data and avoid opening the filter to evasions. The following expression works well for this case:

```
(^|\x0a)Host:[\t]*(youtube.com|[A-Za-z0-9\-\.]*\.youtube\.com)\x0d*\x0a
```

The following is a breakdown of this expression:

`(^|\x0a)`	Specifies to look for the pattern match at the beginning of the packet or at the end of the prior HTTP header line. This helps to establish context and ensures that the right header is being examined.
`Host:[\x20\x09]*`	The “Host” header. The HTTP RFC allows for any number of whitespace (space and tab) after the header, so this is accounted for here.
`(youtube.com|[A-Za-z0-9\-\.]*\.youtube\.com)`	This part of the expression allows for variations of “youtube.com,” such as “account.youtube.com” or “myfakehostname.youtube.com,” that might be accepted by the Web server, while eliminating domain names such as “www.mycheapknockoffyoutube.com”, which could be considered a false positive.
`\x0d*\x0a`	This part of the expression looks for the terminating line feed. The ‘*’ after the carriage return indicates that zero or more can appear before the line feed. This ensures the filter is looking at “.com” and helps to reduce the chance of false positives. For example, “cloud computing” on content delivery networks could result in a false positive without this check.

Blocking a data stream

In this example we create a filter to block or rate limit the download of a data stream. In this case, we will use filter 4371 as our example to block a video stream from Youtube.

Note

YouTube may use different encodings than those described in this example in the future (such as MPEG 4 version 2, see filter 8057).

When a user visits Youtube, their browser downloads an Adobe Flash object that plays video, and then begins to download video content in the form of Adobe Flash Video format. Let’s say you want to allow users to visit

Youtube, but want to rate limit the video traffic to allow for more important protocols to have higher quality access over the network. To do this, you need to define a filter that matches and takes action on a particular stream.

After researching how the video stream is formed, you find a simple “FLV” tag in the header that indicates the Flash Video format. However, “FLV” could appear in just about any data stream, so you will need more context for a proper search string.

Further research indicates the following format on the header field:

dword	Magic Bytes “FLV” (all uppercase) and version number (always |01|)
byte	TypeFlags. First 5 bits are 0, next bit is if audio tags are present, next bit is 0, 8th bit is for if video tags are present.
dword	DataOffset. Usually \x00\x00\x00\x09; may be bigger in future versions, yet has not changed for several years.
dword	FLV File Body PreviousTagSize0 (must be 0)

Specifying the payload string search

FLV|01| is only 4 bytes, so you will need one more to satisfy the minimum requirement for a DVT filter. The next byte uses bits as flags, so the value could be a multiple combination of bytes. Because search strings work at the byte level, you will need multiple search strings to properly send a traffic stream up for inspection. The more bytes you add to a search string, the more reliable and better performing is the filter. For this example, we use the following search strings:

```
FLV|01 00 00 00 00 09 00 00 00 00|
FLV|01 01 00 00 00 09 00 00 00 00|
FLV|01 04 00 00 00 09 00 00 00 00|
FLV|01 05 00 00 00 09 00 00 00 00|
```

Specifying a regular expression

Now that the search strings have been defined, we need a single regular expression to ensure that the filter is inspecting the beginning of a stream and not halfway down a downloaded program that happened to match the filter. We specify the following for the Payload Regular Expression:

```
(^|\x0a\x0d?\x0a)FLV\x01[\x00\x01\x04\x05]\x00\x00\x00[\x09-\xff]
\x00\x00\x00\x00
```

Note

This expression is the search string restated with a context match added at the beginning of the expression.

Because of differences in the way HTTP servers return information, the pattern match will look for a caret (^) for the packet start or a double line feed with optional carriage returns. We use a Payload regular expression rather than an HTTP regular expression—the HTTP decoder is useful for inspecting client requests, not returned responses and stream.

Remember to set the “Source Port” as HTTP. When the IPS sees the traffic, it inspects the ports based on what is in the packets, not by what was negotiated in the TCP handshake. This establishes context for which side of the conversation is being inspected.

Creating a custom filter

Describes how to create a custom filter.

This procedure takes you step-by-step through the creation of a sample custom filter in DV Toolkit. This filter prevents attacks by the Code Red II worm and is also described in *DVT sample filters*.

Procedure

1. Open the DV Toolkit application.
2. Select **Filter > New** from the menu.

   The **Details** window opens at the **General** tab.
3. Enter the following information on the General tab:
   - **Name:** Code Red II Worm
   - **Author:** John Smith
   - **Description:** Code Red II is a worm that exploits a buffer overflow vulnerability in a Microsoft IIS Server ISAPI Extension.
4. Click the **Properties** tab.
5. Define the filter properties.

   Use the pull-down menus to match the following properties:
   - **Protocol**: HTTP
   - **Severity**: Major
   - **Category**: Virus (These categories belong to one of a subset of three main categories into which Digital Vaccine filters are organized and managed in the LSM. For more information about the filter categories see *Properties*).
   - **Class**: Access (specifies the type of threat this filter targets. This is an arbitrary classification provided for the filter-writer to classify the type of threat for analysis).
   - **IP Version**: Select the **Version 4** (IPv4) option.
   - **Attacker Address**: Select the **Source IP** option.
6. Click the **Payload** tab.

   This tab is where you enter the string and regular expression that define the payload filter criteria (layers 5-7). You can enter multiple strings for the String Search and for the Regular Expression. For this example, we enter a single string and a single regular expression.

   a. Click **ADD** and enter the following text in the **String Search** text box.

      Note that **AND** is selected by default for the Test Block.

      ```
      /default\.ida
      ```

   b. Check the **case-sensitive** check box and enter 0 for **Offset** and -1 for **Depth**.

   c. Select **Regular Expression** and click **ADD**.

   d. Enter the following text in the **Regular Expression** text box.

      ```
      /default\.ida\?X{224}%u9090%u6858%ucbd3%u7801

      %u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090
      ```

```
%u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00
```

e. Click **Enter** to validate the search criteria you entered.

Note

When trigger constraints are enabled, DV Toolkit protects you from entering a word or phrase that could constitute a "bad" substring. For more information about this feature, see *Filter trigger constraints*.

7. Click the **TCP** tab where you will enter the destination port information.

   HTTP is already selected as the Destination Port because you selected **HTTP** as the protocol in an earlier step when defining the properties. Accept this default setting and the remainder of the default settings.

8. Click the **HTTP** tab, then select **Regular Expression** in the tree view and click **ADD**.

9. Enter the following URI regular expression in the Regular Expression text box, as shown in the figure below:

```
/default.ida\?X{224}%u9090%u6858%ucbd3%u7801%u9090

%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u9090

%u8190%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00
```

   The URI regular expression string specified is intended to match the substring of the HTTP request's URI. Make sure URI is the selected option for the Regular Expression. For more information about HTTP inspection processes for regular expressions, see *HTTP* .

   Click **Enter** to validate the criteria entered.

Note

If the string entered is not Perl-Compatible Regular Expression compliant, it is treated as fixed content.

10. Click **OK** to save the custom filter.

    The filter is added to the DV Toolkit main window and assigned the number C001.

11. Use the **File > Save As** menu selection to save the filter package.

    The filter package name is limited to 34 characters or less. The package file is created with a `.csw` extension.

What to do next

After creating a filter, you can continue to create additional filters, incorporate converted Snort rules into your filters, or deploy the filters to your TippingPoint devices. To load the newly created filter package on an LSM or an SMS, see *Distributing custom filters*.

Merging DVT filter packages

The DV Toolkit provides a Merge function that enables you to select filters from DVT filter packages and merge them with another DVT filter package. You can use this feature to merge converted Snort rules into an existing DVT filter package or combine filters defined in several different DVT filter packages into a single customized filter package.

To merge DVT filter packages:

Describes how to merge DVT filter packages.

Procedure

1. Open the DVT filter package that you want to use to create the merged DVT filter in DV Toolkit.
2. In the menu, select **File > Merge**.
3. In the Merge Package dialog box that opens, locate the DVT filter package containing the filters that you want to include in the merged file and then select the file.

   All filters contained in the selected file are displayed in the Digital Vaccine Filter List.

   Note

   All DVT filter packages have the `.csw` file extension.

4. Select the filters that you want to include in the merge operation.

   By default, all filters in the list are selected for the merge operation. You can exclude filters by unchecking the checkbox for a filter or deselect all by unchecking the Filter check box in the column header.
5. After making your selections, click **Merge** to complete the operation.
6. DV Toolkit imports the selected filters into the opened filter package adding them to the filter set.
7. You can further customize the filters contained in the merged filter set or save the filter set as a DVT filter package.

Using snort rules in custom filters

For organizations that use Snort for network intrusion prevention, DV Toolkit includes a Snort conversion tool. Using the Digital Vaccine Converter tool you can convert Snort rules to Digital Vaccine format and use the rules or rule set in your Digital Vaccine custom filter packages.

After importing the converted rules to DV Toolkit and saving them to a DVT filter package file, you can modify the filters and merge them with other DVT filters to create custom filter packages. You can also use the DVT command-line interface (CLI) to create DVT filter packages with converted Snort rules. For more information, see *Digital Vaccine Toolkit Command Line Interface*.

For more information about converting and using Snort rule sets in DVT filter packages, see *DV converter*.

Distributing custom filters

After you create and save the DV Toolkit package, you can install it on a TippingPoint device using the LSM or by using the SMS. The SMS enables you to activate and distribute a DVT custom package to multiple IPS devices through SMS profiles. After filters have been distributed to the LSM or SMS, they must be enabled.

This section includes the following procedures:

- *Installing a filter package by using the LSM*
- *Importing and distributing a filter package on the SMS*

Installing a filter package by using the LSM

Describes how to install a filter package by using the LSM.

You can use the LSM to install the package on an IPS device.

Procedure

1. Log in to the LSM.
2. Select **System > Update > Install Package** from the navigation menu.
3. Click **Browse** next to the **Package File** field.
4. Browse to the location of the DVT package file, click on the package file name in the **Choose File** window, and click **Open**.

   The file path and name is displayed.
5. Click **Install Package**.

   A progress bar appears while the DVT package is being installed.

What to do next

The DVT filter package must be added to a security profile and enabled. For information about adding a filter to a security profile and enabling it, refer to "Edit Individual Filter Settings" in the *LSM User Guide*.

Importing and distributing a filter package on the SMS

Describes how to import and distribute a filter package on the SMS.

Using the SMS you can distribute DVT packages to multiple IPS devices. To enable DVT packages by using the SMS, you must import the package, distribute the package to IPS devices, and activate the package. You can activate packages before or after distributing them.

Procedure

1. Log in to the SMS Client.
2. In the **Profiles** Navigation pane, click **DV Toolkit Packages**.

   The **Profiles – DV Toolkit Packages** screen displays.
3. In the **DV Toolkit Package Inventory** area, click **Import** and then browse to the file location and select the package file you are importing.

   The file imports and displays in the **DV Toolkit Package Inventory** area. The package is also added to the **DV Toolkit Package** navigation tree.
4. Use one of the following actions to distribute the package:
   a. In the **DV Toolkit Package Inventory** section, select the package and click **Distribute**.
   b. In the **DV Toolkit Package Inventory** section, click **Details**.

      On the displayed screen, click **Distribute**.
   c. Right-click an entry and select **Distribute**.

What to do next

The SMS distributes the package. During the distribution process, status updates are displayed in the **Distribution Progress** section. If the package has not been activated, you must activate it before it is enabled for the devices.

You can also deactivate and uninstall DVT packages that have been distributed to IPS devices. For additional information about DVT package management in the SMS, refer to "Digital Vaccine Package Management" in the *SMS User Guide*.

Note

You must have Super User or Administrator access to import, distribute, and configure DV Toolkit packages on the SMS.

DV converter

Digital Vaccine Converter is part of the Digital Vaccine Toolkit tool set. The DV Converter tool converts Snort filters to Digital Vaccine syntax. The filters can then be imported to Digital Vaccine Toolkit formatted for use in DVT filter packages. The converted filters are treated as any other DVT filter package and can be managed by the Local Security Manager (LSM) on a TippingPoint IPS device or by the Security Management System (SMS).

Note

The DV Converter parser accepts well-formed Snort syntax developed according to the Snort open-source development standards and translates it to Digital Vaccine syntax. Rules that do not comply with these standards may not successfully convert. For more information about Snort rules and the Snort open-source community standards, visit the Snort website.

This section includes the following topics:

Getting started with DV converter

The Digital Vaccine Converter is a companion application of the Digital Vaccine Toolkit (DV Toolkit); although part of the DV Toolkit tool set, it installs and is launched as a separate application. The tool is designed to translate Snort rule syntax to Digital Vaccine syntax and validate the syntax against Digital Vaccine formatting rules to prepare the rules for import to DV Toolkit. Once imported, you can use the rules in DVT filter packages.

When you open a Snort filter (**.rules*) file in DV Converter, the rules contained in the file are listed in the main view with status information about the rule conversion displayed.

For more information about the user interface and to get started using the DV Converter tool, see the following topics:

Installation

Describes how to install the DV Converter application.

The Digital Vaccine Converter is available for download from the TippingPoint Threat Management Center (TMC), along with the Digital Vaccine Toolkit. DV Converter installs as a separate application.

To install the DV Converter application, access the TippingPoint Threat Management Center (TMC).

Note

Before installing the application see the system requirements in the Digital Vaccine Toolkit Release Notes available on the TMC.

Procedure

1. Log in to the TMC at https://tmc.tippingpoint.com using your customer credentials.
2. On the **Releases** tab, choose **Digital Vaccine Toolkit**.
3. When the Digital Vaccine Toolkit download page opens, select the latest version of the DV Toolkit and then click on the DVConverter installer package file (either the 32-bit or the 64-bit version, whichever is appropriate for your system).
4. Save the file in a location on the system to which you are installing the application.
5. Click the file name (`dvtoolkit_converter-vn.n.msi`) to start the installer.

   `vn.n` represents the version you downloaded (may include additional digits that indicate build number). `nn`bit represents either the 32-bit or the 64-bit version.

What to do next

To start Digital Vaccine Converter, click the Digital Vaccine Converter icon on your desktop.

Menu commands

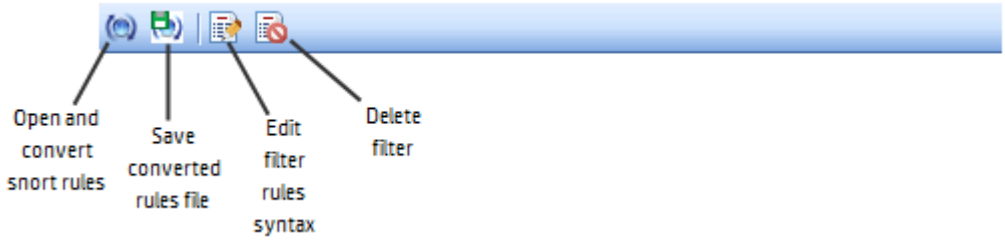
The Digital Vaccine Converter user interface consists of a main window, in which you can view the Snort rules parsed by DV Converter and use menu and toolbar commands to perform tasks, and a Filter Details window in which you can edit attribute values for converted rules. The DV Converter tool also has a command-line interface from which you can run the converter.

The following table describes the menu actions available on the Digital Vaccine Converter menu bar:

Menu Item	Command	Option	Description
File	Open	Snort Rules	Opens a Snort rules file and displays the list of rules and the conversion status for each rule.
	Save	Digital Vaccine XML	Saves the converted Snort filter package in Digital Vaccine format. **Note** A package may contain up to 8000 filters.
	Exit		Exits the application. Can also use Shortcut key Alt + F4.
Rule	Edit...		Opens the Filter Details window for the selected filter, in which you can view the converted Snort values and edit values to conform to valid syntax.
	Delete		Deletes the selected filter or filters.
	User Variables...		Opens the User Variable window for the selected filter, in which you can view and modify variable *Type*, *Name*, and *Value*.
	Options...		Opens the Options window for the selected filter, in which you can enable and disable primitive constraints.
Help	Users Guide		Opens the online help for the tool.
	About		Displays the application version number and company information .

Toolbar icons

You can use the Digital Vaccine Converter toolbar in the main window as a shortcut to menu actions.



Filter details window

The **Filter Details** window opens when you double-click a filter rule listed in the main window or when you choose the **Edit** option from the menu bar or toolbar. The **Filter Details** window presents a table of the attribute values defined for the filter rule which you can modify.

The Filter Details window comprises three main areas:

- **Filter Data Table** – Displays each attribute (in the **Tag** column) and its associated value (**Value** column) defined in the rule header or rule options and provides a status indicator (**Status** column) that represents if the value is acceptable Digital Vaccine syntax.
- **Status Memo** – Correlates with the status column and provides more information about the status or the value defined for the attribute. This information may be that the value for the associated Tag is valid or invalid, or it may be a tip that explains how changing the value of an attribute can improve the performance of the filter. The message correlates to the *Status indicators* shown for the attribute.

- **Edit Data** – The edit area in which you change the value of an attribute (Tag item), if the value is editable.

For more information about editing filter attribute values in the Filter Details window, see *Editing and validating Snort rule attributes* and the *Snort to Digital Vaccine conversion table*. You can also refer to the DV Converter online help for more details.

User-defined variables window

The User Defined Variables window allows you to view and modify the *Type*, *Name*, and *Value* for your filter variables. To open this window, select **Rule > User Variables...** in the menu bar.

The User Defined Variables window has three columns listing the variables in your filter:

- *Type* – A drop-down menu allows you to select among the choices **IP Address** and **Port** for your filter variable type.
- *Name* – The text field allows you to change the name of the selected filter variable.
- *Value* – The text field allows you to change the value of the selected filter variable.

For more information about editing filter attribute values in the Filter Details window, see *Editing and validating Snort rule attributes* and the *Snort to Digital Vaccine conversion table*. You can also refer to the DV Converter online help for more details.

Options (filter primitives) window

The Options window allows you to set primitive constraints during the conversion process. To open this window, select **Rule > Options ...** in the menu bar.

The Options window allows you to choose any of the following conversion constraints by selected their checkbox:

- **Ignore unsupported primitives**
- **Ignore unsupported modifiers**
- **Ignore unsupported values**

You can chose any combination of constraints, or none. If a checkbox is deselected, your filter warns you of unsupported primitives. The default is for all options to be deselected.

The following tables display the Rules Headers primitives that are either supported or ignored:

Snort option	Support level
Actions	I (ignored)
Protocols	Y
IP Addresses	Y
Port Numbers	Y
Direction Operator	I

The following tables display the Rules Options primitives that are either supported or ignored:

Snort option	Support level
msg	Y

Snort option	Support level
reference	Y
sid	Y
rev	Y
classtype	I
metadata	I
uricontent	Y
urilen	Y
isdataat	Y
fragoffset	Y
ttl	Y
tos	Y
id	Y
fragbits	Y
dsize	Y
flags	Y
flow	I
seq	Y
ack	Y
window	Y
itype	Y
icode	Y
icmp id	Y
icmp seq	Y
ip proto	Y
content	Y
pcre	Y

The following tables display the Content Modifiers primitives that are either supported or ignored:

Snort option	Support level
nocase	Y
rawbytes	N
depth	Y
offset	Y
distance	Y

Snort option	Support level
within	Y
http client body	Y
http cookie	N
http raw cookie	N
http header	Y
http method	Y
http uri	Y
http raw uri	Y
http stat code	N
http stat msg	N
fast pattern	N

The following tables display the PCRE Modifiers primitives that are either supported or ignored:

Snort option	Support level
i	Y
s	Y
m	Y
x	Y
A	Y
E	Y
G	Y
R	Y
U	Y
B	Y
P	Y
H	Y
M	N
C	Y
O	N
I	Y
D	Y
K	Y
Y	Y
S	N

The following rules options are *not* supported:

- gid
- priority
- protected content
- hash
- length
- pkt data
- file data
- base64 decode
- base64 data
- byte test
- byte jump
- byte extract
- ftpbounce
- asn1
- cvs
- dce iface
- dce opnum
- dce stub data
- sip method
- sip stat code
- sip header
- sip body
- gtp type
- gtp info
- gtp version
- ssl version
- ssl state
- ipopts
- flowbits
- rpc
- sameip
- stream reassemble

- stream size
- logto
- session
- resp
- react
- tag
- activates
- activated by
- count
- replace
- detection filter

Status indicators

Status icons appear in the main window and in the Filter Details window of the DV Converter interface. These icons represent the state of the converted rule syntax. The following table describes how to interpret these status indicators.

Status Icon	Description	Action Required
	In the Main window, represents a valid rule conversion. In the Filter Details window, represents a valid attribute value.	None. The rule is valid and can be modified or saved to a DVT (XML-formatted) conversion file.
	In the Main window, represents invalid DVT syntax for the rule. In the Filter Details window, represents and invalid attribute value.	The rule or rule attribute is invalid and must be modified and validated before conversion is successful.
	In the Filter Details window, represents and undefined attribute.	None. DV Converter ignores undefined attributes when compiling the conversion file.
	In the Filter Details window, this icon represents and informational message is available. The information is displayed in the Status Memo area and may provide a performance tip, recommendations, or warnings related to the attribute value.	Review the information provided in the status memo area and make adjustments to the attribute value if warranted.

Command line interface

DV Converter enables you to run a conversion from the command line. To use the DV Converter command line interface, see *DV Converter Command Line Interface*.

Converting Snort filters to Digital Vaccine

A Snort rule consists of a rule header and the rule options that are specified in the rule body. The rule header includes such items as the rule action, protocol, IP addresses, and source and destination ports. The rule

options, though not required in a Snort filter, typically define additional metadata, payload, or detection information. For more information about Snort rules, refer to the Snort open-source community website.

The DV Converter parser translates all attribute values in a Snort rule to an equivalent attribute if applicable in a Digital Vaccine filter. If a Snort attribute is not defined in Digital Vaccine syntax, the undefined attribute is ignored for the conversion. For example, common Snort attributes such as the "action" attribute, the bidirectional operators, and others, are not applicable in Digital Vaccine filters. The negation of String Search data strings are not supported in DV filters.

If all translated applicable attributes have valid values, the rule will validate successfully for Digital Vaccine format.

For more information about converting Snort filters, see the following topics:

- *Snort to Digital Vaccine conversion table*
- *Allowed Snort variables*
- *Converting Snort filters*

Snort to Digital Vaccine conversion table

The following table describes the Snort attributes that convert to Digital Vaccine syntax.

Attribute	Snort Syntax	DVT Conversion
action	Snort rule header attribute, such as "alert"	Not valid DVT syntax.
protocol	TCP, UDP, IP, ICMP	Converts to equivalent DVT value: IP, ICMP, UDP, TCP, HTTP.
IP (ip.src or ip.dest)	ANY, variable name, or numeric IP address and a CIDR[3] block	All variables convert to ANY. Specified values translate as is if successfully validated. See *Allowed Snort variables* for additional information.
Port (port.src or port.dest)	ANY, variable name, static port definitions or ranges in the format of <startingnumber>:<endingnumber>.	All variables convert to ANY. Specified values convert to the same value in DVT.
direction		Not valid DVT syntax.
msg	Text field	Converts to DVT filter name.
flow		Not valid DVT syntax.
content	String value	Converts to DVT payload search string; must be a minimum of 5 characters. **Note** If either the content, uricontent, or PCRE attribute is specified with a valid value, DV Converter will validate the rule successfully.
urilen	String value	Converts to equivalent value in DV Toolkit as applies to HTTP URI Length

Attribute	Snort Syntax	DVT Conversion
uricontent	String value	Converted as applicable to the HTTP.URI, HTTP.METHOD, or HTTP.HEADER options in DVT; must be a minimum of 5 characters.
isdataat	Integer value of -65535 to 65535 in the format: [!]<number>[, relative | rawbytes	Converts to DV Toolkit filter payload regular expression.
classtype		Not valid DVT syntax.
reference	Text field containing <id system>, <id>	Converts to DVT filter description.
sid	Text field (Snort ID number)	Added to DVT filter description.
rev	Integer	Converts to DVT revision number.
itype	A number value from 0-255 in the format: [<|>]<number.	Converts to equivalent number in DVT. Must edit the filter to specify the operators [<|>].
icode	A number value from 0-255 in the format: [<|>]<number>	Converts to equivalent number in DVT. Must edit the filter to specify the operators [<|>].
ip_proto	A number value from 0-255 in the format: [<|>]<number>	Converts to equivalent number in DVT. Must edit the filter to specify the operators [<|>].
distance	Integer value of -65535 to 65535	Converts to equivalent number value and applies to DVT Payload search string.
within	Integer value of -65535 to 65535	Converts to equivalent number value and applies to DVT Payload search string.
offset	Integer value of -65535 to 65535	Converts to equivalent number value and applies to DVT Payload search string.
depth	Integer value of 1 to 65535	Converts to equivalent number value and applies to DVT Payload search string.
nocase	Keyword that specifies to ignore case in CONTENT or URICONTENT string.	Converts to the same value (no case-sensitivity) in DVT as applies to payload search strings.
PCRE	Perl Compatible Regular Expression	Converts to DVT filter payload regular expression.
flags	Specified as format: [+*!]<MDR>	Converts to DVT filter IP header flags. (Reserved, Don't Fragment, More Fragments).
fragbits	Specified as format: [!|*|+]<FSRPAU12> [,<FSRPAU12]	Converts to DVT filter TCP Header Flags enabling the equivalent fragbits.
fragoffset	A number value from 0-8191, specified as format: [!|<|>]<number>	Converts to equivalent value in DVT.
window	Integer value from 1 to 65535	Converts to equivalent value in DVT as applies to TCP window size.
TTL	A number value from 0-255. specified in one of the following formats: ttl:[<,>,=,<=,>-]<number> ttl:[<number>-<number>]	Converts to an equivalent Time to Live value in DVT.

Attribute	Snort Syntax	DVT Conversion
TOS	A number value from 0-255, specified as format: tos:[!]<number>	Converts to an equivalent Type of Service value in DVT.
ID	A number value from 0-65535, specified as format: id:<number>	Converts to an equivalent Identification value in DVT.
dsize	A number value from 0-65535, specified as format: dsize:[<|>]<number>	Converts to an equivalent value for Data Length in DVT.
seq	A number value from 0-65535, specified as format: seq:<number>	Converts to an equivalent value for Sequence Number in DVT.
ack	A number value of 0-65535, specified as format: ack:<number>	Converts to an equivalent value for the TCP Acknowledgment Number in DVT.
ICMP_ID	A number value of 0-65535, specified as format: icmp_id:<number>	Converts to an equivalent value for the ICMP ID value in DVT.
ICMP_SEQ	A number value of 0-65535, specified as format: icmp_seq:<number>	Converts to an equivalent value for the ICMP Sequence value in DVT.

Allowed Snort variables

The DV Converter tool accepts the following Snort variables for conversion of IP and Port designations. The values for these variables are translated to “ANY”, or for port settings to the values indicated in parenthesis below, during the conversion process.

IP variables

- $HTTP_SERVERS
- $HOME_NET
- $EXTERNAL_NET
- $DNS_SERVERS
- $TELNET_SERVERS
- $SSH_SERVERS
- $AIM_SERVERS
- ANY

Port variables

- $HTTP_PORTS (80, 3128,8000,8080)
- $SSH_PORTS (22)
- $SHELLCODE_PORTS (1024-65535)
- $ORACLE_PORTS (1024-65535)
- ANY

Converting Snort filters

To convert Snort filters in Digital Vaccine Converter, open a Snort rules (`.rules`) file in DV Converter and save the file to Digital Vaccine XML format. All rules in the file that have successfully validated to Digital Vaccine syntax are saved in the conversion file. Invalid rules are not exported to the file and are reported in the Export Error Log that displays.

To convert a Snort rules file to Digital Vaccine

Describes how to convert a Snort Rules file to Digital Vaccine.

Procedure

1. In DV Converter, open the `.rules` file that you intend to convert by selecting **File > Open > Snort Rules** from the menu and browsing to the file.
2. Determine if any rules are invalid and which you wish to modify by reviewing the displayed *Status indicators* for the rules in the file.
3. If any rules are invalid, edit the rule attribute values until they validate successfully.

   You can also delete any rules that you do not want in the conversion file.
4. When all rules that you want to convert are validated, save the conversion to a file by following these steps:
   a. In the menu, choose **File > Save > Digital Vaccine XML**, or select the Save icon on the toolbar.
   b. Navigate to the location where you want to save the file and name the file with a meaningful name that properly identifies the rule set.
   c. Click **Save**.

      DV Converter saves the file in XML format (by default the `.xml` file extension is selected).

What to do next

After saving the conversion file, you can import it into Digital Vaccine Toolkit to include in DVT filter packages. A package may contain up to 8000 filters.

Editing and validating Snort rule attributes

When you open a snort filter in DV Converter, you can edit the rule attributes by double-clicking a rule listed in the main view or by selecting the rule and choosing either of the following options:

- From the menu bar, select **Rule > Edit**
- From the toolbar, select (Edit filter)

The Filter Details window (see *Filter details window*) opens enabling you to modify Tag values in the **Edit Data** area. You can only modify valid key-value pairs. If the Tag attribute is an undefined DVT attribute (as indicated by the icon), you cannot edit the value. DV Converter ignores the undefined attributes when compiling the converted file.

Note

You cannot add or delete attributes in the Digital Vaccine Converter tool. To do so, you must modify the snort rule outside of the DV Converter tool and then reload the rule file into DV Converter.

To edit an entry

Describes how to edit a snort rule attribute entry.

Procedure

1. In the Filter Details window, select a Tag item from the Filter Data list.

   The selected Tag value appears in the **Edit Data** area.

2. Note the information provided for the Tag item in the **Status Memo** area.

   If the item is a valid DV filter attribute you can edit the values for the attribute. The value specified for the attribute must be valid.

3. In the **Edit Data** area, enter a new value in the **Value** field and then click **Enter** to apply your changes.

   If the value is a string value, you can check the case-sensitivity check box to specify that case-sensitivity will be applied to the filter string. If other values are displayed for the Tag item, modify accordingly.

4. Repeat the preceding steps for each Tag value that you want to modify until all values have validated.

   All Tag items have validated when no “invalid” icons ( ) appear in the status column.

5. When you have finished your edits, click **OK** to save the rule changes and exit the **Filter Details** window.
6. In the main window, ensure that the status icon indicates the edited rule is valid DVT syntax.
7. Complete the foregoing steps for each rule in the filter file that you need to edit.

What to do next

Note

Changed values are committed when you click **Enter** to apply the changes and then click **OK** to save the changes.

Importing converted Snort rules to DV Toolkit

To use converted snort rules in a Digital Vaccine filter package, you must import the rules in DV Toolkit. After you import the rules, you save the entire rule set creating a new DVT filter package. You can then deploy the filter package to your TippingPoint devices (in a test environment) or combine the filters with other DVT custom filters.

To import converted Snort rules

To import a Snort filter that you have converted to Digital Vaccine format, follow these instructions:

Procedure

1. In the DV Toolkit main window, click **File > Import XML** from the menu bar, or click the Import toolbar icon, and then browse to the location of the `.xml` conversion file you are importing.

   Note

   The default file extension for conversion files saved in DV Converter is `.xml`. If you named the file with some other extension, DV Toolkit will import the file which is in the required XML format.

2. Select the file and click **Open**.

   All converted rules in the file appear as a filter set in the main window. You can customize the filter set by editing the filters or create a filter package to deploy to your devices. A package may contain up to 8000 filters.

3. To create the filter package, save the filter set by selecting **File > Save As** in the menu and assigning a meaningful name to the filter package that describes the rule set.

What to do next

The filter package is ready for deployment to your TippingPoint systems.

Note

Trend recommends that you deploy the converted rules to a test environment before deploying to full production systems.

Adding converted Snort rules to an existing DVT filter package

After you have created a DVT filter package containing the Snort rules, you can use the new filter package containing only the Snort rules, or you can merge the Snort rules with existing DVT filter packages to create a customized filter set to deploy to your devices.

For more information about merging DVT filter packages, see *Merging DVT filter packages*.

For more information about deploying the filter packages, see *Distributing custom filters*.

DVT sample filters

The following examples show some sample DVT filters:

- *Example 1: Stacheldraht DDoS Tool Communications (ICMP)*
- *Example 2: ServHelper (HTTP)*
- *Example 3: AESDDoS (TCP; Custom protocol)*

Example 1: Stacheldraht DDoS Tool Communications (ICMP)

Stacheldraht agents and masters communicate using ICMP echo replies. These messages carry specific strings that distinguish them from normal ICMP packets. A sample trace showing a DDoS master responding

to a status query from an agent is provided below. A machine infected with a Stacheldraht agent will send status queries at regular time intervals in an attempt to establish and maintain communication with the master.

The packet data contains the word "spoofworks" and a lot of null bytes. A filter that will detect these messages is configured as follows:

- Protocol (*Properties*): `icmp (type = 0 Echo reply)`
- String Search (*Payload*): `|00|spoofworks|00|`
- Regular expression (*Payload*): `\x00spoofworks\x00{800}`

Packet trace for Stacheldraht Communications

```
------------------------------------------------------------
CMP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:1044 DF
Type:0  Code:0  ID:9000  Seq:0  ECHO REPLY
0x0000: 00 11 11 11 11 11 63 0B 7F 9A 00 00 08 00 45 00  ......c.......E.
0x0010: 04 14 00 00 40 00 40 01 38 E7 7F 00 00 01 7F 00  ....@.@.8.......
0x0020: 00 01 00 00 B8 9A 23 28 00 00 00 00 00 00 00 00  ......#(........
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 73 70  ..............sp
0x0040: 6F 6F 66 77 6F 72 6B 73 00 00 00 00 00 00 00 00  oofworks........
0x0050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x00A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x00B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x00C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x00D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x00E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x00F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0100: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0110: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0120: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0130: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0140: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0150: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0160: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0170: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0180: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0190: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x01A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x01B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x01C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x01D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x01E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x01F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0200: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0210: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0220: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0230: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0240: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0250: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0260: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0270: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0280: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

```
0x0290: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x02A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x02B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x02C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x02D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x02E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x02F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0300: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0310: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0320: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0330: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0340: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0350: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0360: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0370: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0380: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0390: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x03A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x03B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x03C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x03D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x03E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x03F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0400: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0410: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0420: 00 00                                            ..
```

Example 2: ServHelper (HTTP)

ServHelper is a Trojan Backdoor that targets the Windows platform. This malware is used in multiple spam campaigns targeting financial institutions, retail companies, and other businesses. It contacts a remote server to identify itself and receive control commands. These commands include setting up a reverse SSH tunnel, downloading and executing a file, deleting itself, terminating a process, and executing shell commands.

To register the infection, the malware sends an HTTP POST request over port 80 to the C&C server. (Other variants communicate over different port numbers.) For example, the malware sends a request similar to the following:

```
POST /ghuae/huadh.php HTTP/1.1
Connection: Keep-Alive
Content-Type: application/x-www-form-urlencoded; charset=utf-8
User-Agent: Embarcadero URI Client/1.0
Content-Length: 137
Host: checksolutions[.]pw

key=Gsiss744%40sd&sysid=specinj%3AWindows+7+%28Version+6.1%2C+Build+7600%2C+64-bit+
Edition%29_x64_username%3A+MalwrLab_32190&resp=init+ok
```

The body of the POST request contains the parameters **key**, **sysid**, and **resp**.

- **key:** Holds a hardcoded value that changes among variants. We do not recommend that you use this value in the detection logic.
- **sysid:** Holds a hard coded ID, like **specinj**, for example, and infected system OS specific information. The hard coded ID value changes among variants, so we do not recommend that you use this value in the detection logic.

- **resp:** Holds the hard coded value **init+ok**. This value is specific to this type of request. In other requests generated by the malware, this parameter holds responses pertaining to executed commands received from the C&C server. You can use this value in the detection logic.

A generic filter that detects the ServHelper POST request is configured as follows:

- **Protocol:** `TCP`
- **Source:** `1024-65535`
- **Destination:** `Any`
- **Trigger (String Search):** `&syid=`
- **Payload (Regular Expression):** `\x0d\x0akey=[^&\x0d\x0a]{0,48}&sysid=[^&\x0d\x0a]{0,156}&resp=init\+ok$`

Example 3: AESDDoS (TCP; Custom protocol)

AESDDoS, which is also known as Spike, MrBlack, and Darkshell, is a bot agent that targets different platforms and architectures. Variants of this malware are delivered onto unsecured Docker containers that are exposed to the Internet without proper authentication. The malware's primary function is to receive control commands to perform various types of Denial-of-Service (DoS) attacks against a given target. These attack types include SYN, LSYN, UDP, UDPS and TCP floods. It is also capable of executing shell commands on the infected system and downloading and executing a file, among others.

To register the infection, the malware sends a TCP packet over port 9002 to the C&C server. (Other variants communicate over different port numbers.) For example, the malware sends a request similar to the following:

```
00000000  56 45 52 53 30 4e 45 58  3a 33 2e 31 31 2e 30 2d   VERS0NEX :3.11.0-
00000010  31 32 2d 67 65 6e 65 72  69 63 7c 31 7c 33 33 39   12-gener ic|1|3392
00000020  32 7c 4d 72 2e 42 6c 61  63 6b 00                  2|Mr.Bla ck.
```

The packet has the following string format pattern: `VERS0NEX:%s|%d|%d|%s`. Other variants have a slightly different format pattern. The packet starts with the hardcoded string `VERS0NEX`, but other variants use the string `VERSONEX`, with the digit zero replaced by the letter O.

Various system information, delimited with the hardcoded pipe "|" character, comes after the hardcoded string.

- **3.11.0-12-generic:** The infected system kernel release version.
- **1:** Represents the number of processors online.
- **3392:** A pseudo-random number generated by the malware.
- **Mr.Black:** Hardcoded in the malware, and used as a unique identifier. Because this string changes among variants, it is not recommended to take it in the detection logic.
- **\x00:** Ends the packet.

A generic filter that detects the AESDDoS registration request is configured as follows:

- **Protocol:** `TCP`
- **Source:** `Any`
- **Destination:** `Any`

- **Trigger (String Search):**
  - VERS0NEX
  - VERSONEX
- **Payload (Regular Expression):** `^VERS[0O]NEX:[^\x7c\x0d\x0a]{1,48}\|\d[^\x7c\x0d\x0a]{0,24}\|`

> **Note**
>
> The caret (^) at the beginning of the regular expression forces the expression to match only at the beginning of the TCP packet.

Digital Vaccine Toolkit Command Line Interface

DV Toolkit provides a command-line interface that you can use to create DVT filter packages. The commands are available from the following location in your DV Toolkit Installation directory:

```
C:\Program Files\Trend Micro TippingPoint\DVToolkit
```

Syntax

To run a command from the DVT command-line interface, use the following syntax at the system command prompt:

```
dvt [command] [OPTION] SOURCE DESTINATION [-l log]
```

SOURCE = the path/filename of the source file or files that are processed to create the DVT package file.

DESTINATION = the path and filename of the DVT filter package.

For a description of the commands and options available in the CLI, see *DVT commands*.

DVT commands

The DVT command line interface provides the following commands to create DVT filter packages:

- *Import*
- *Merge*
- *Help*

Import

This command imports XML (`.xml`) conversion files to DV Toolkit and creates a DVT (`.csw`) filter package. The DVT filter can be opened in DV Toolkit and edited or customized as needed.

```
dvt [-i --import] [OPTION] SOURCE DESTINATION [-l LOG]
```

Options

You can use the following options with the `dvt` command.

Command Option	Description
`-a (or --append)`	Appends the contents of the *SOURCE*.xml file to an existing filter package.
`-k (or --keep)`	If DESTINATION file already exists, keeps the original file and creates a new filter package with the same name and a sequence number.
`-o (or --overwrite)`	Overwrites an existing DVT filter package file with the contents of the *SOURCE*.xml file specified.
`-s (or --skip)`	Skips the import process if a DESTINATION file already exists for the specified filter.
`-x (or --exclude)`	Excludes filters with string searches that are extremely common in network traffic (i.e., enforces the trigger-protection constraints). For more information, see *Filter trigger constraints*.
`-r (or --recursive)`	Performs a conversion on the rule files in a directory recursively.
`-l LOG (or --log LOG)`	Writes the conversion results to a log file. *LOG* = the name of the log file
`-v (or --verbose)`	Writes the command results to the console screen.
`? (or -help)`	Displays command usage information on the console screen.

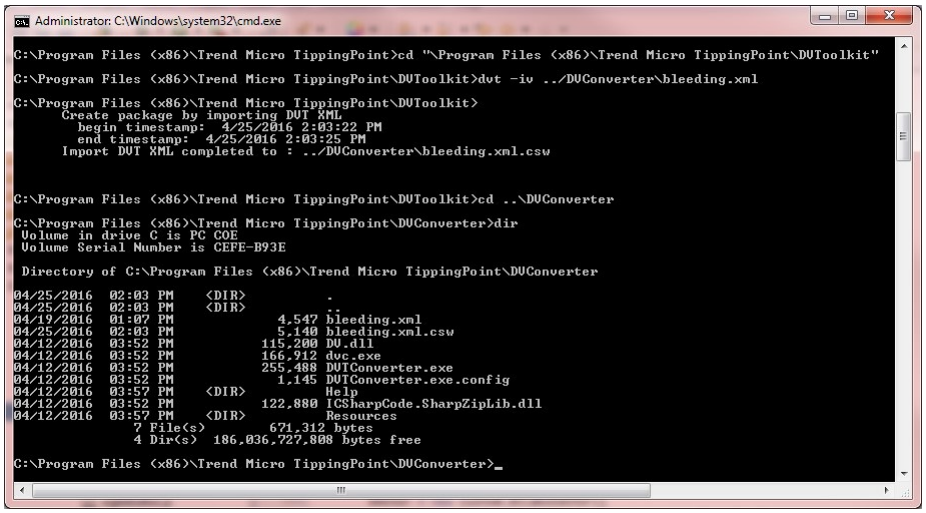
Examples

The following example imports the filters in a DVT conversion (.xml) file named `bleeding.xml` and creates a DVT filter package in the same directory:

```
C:\Program Files (x86)\Trend Micro TippingPoint\DVToolkit>dvt -iv ..
\DVConverter\bleeding.xml
```

The DV Toolkit creates the DVT package file using the same name with the .csw extension appended.

The following illustration shows the results:



Merge

This command merges two DVT filter packages into a single filter package:

```
dvt [-m --merge][OPTION]SOURCE DESTINATION[-l LOG]
```

Options

You can use the following options with the `dvt merge` command.

Command Option	Description
`-r (or --recursive)`	Processes the merge command on each file in a specified directory recursively, creating a single DVT filter package.
`-l LOG (or --log LOG)`	Writes the command results to a log file. *LOG* = the name of the log file
`-v (or --verbose)`	Writes the command results to the console screen.
`? (or -help)`	Displays command usage information on the console screen.

Examples

The following command merges the `test.csw` filter package in the `/home/filters` directory with the `dest.csw` filter package in the same directory.

```
dvt --merge -v /home/filters/test.csw/home/filters/dest.csw
```

The following command uses the recursive option to merge all DVT filter packages (*.csw files) in the `/home/filters` directory and subdirectories into a single `dest.csw` filter package.

```
dvt -mR /home/filters /home/filters/dest.csw
```

Help

The following command invokes the command-line help to provide information about command usage.

```
dvt ? [Command-name]
```

Examples

Either one of the following examples display command usage information about the import command:

```
dvt --help --import
dvt ? -i
```

DV Converter Command Line Interface

DV Converter provides a command-line interface that you can use to run Snort conversions. The commands are available from the following location in your DV Converter Installation directory:

```
C:\Program Files\Trend Micro TippingPoint\DVConverter
```

Command syntax and options

The DV Converter command-line interface accepts the following syntax and options.

Syntax

To run a command-line conversion, issue the following command at your system command line.

```
dvc [OPTION] SOURCE DESTINATION [-l log]
```

SOURCE = the source path and/or filename containing the Snort rules that you are converting

DESTINATION = the destination path and filename that will contain the converted rules formatted in DVT syntax

Options

You can use the following options with the `dvc` command.

Command Option	Description
`-a --append`	Appends the specified DESTINATION file with the current conversion.
`-k --keep`	Creates a new conversion and if the destination conversion file already exists creates a new conversion file with the same name and a sequence number appended. For example: SnortDestinationfile
`-o --overwrite`	Creates a new conversion overwriting the existing DESTINATION conversion file if one already exists.
`-s --skip`	Skips the conversion if conversion file already exists for specified DESTINATION.
`-r --recursive`	Performs a conversion on the rule files in a directory recursively.
`-l LOG --log LOG`	Outputs the conversion results to a log file. *LOG* = specifies the name of the log file
`-v --verbose`	Outputs the conversion results to the console screen.
`? (or -help)`	Displays command usage information on the console screen.

Examples

The following example syntax runs a conversion on a Snort filter (rules) file located in the `\home\filters` directory and saves the converted syntax to a DVT-formatted file named `test.rules.xml` in the `\home\filters\dvt` directory. The verbose argument is used to show the command results.

```
dvc -v \home\filters\test.rules \home\filters\dvt\test.rules.xml
```

With the `-v` option invoked with this command, the following conversion results are displayed to the console:

```
C:\Program Files\Trend Micro TippingPoint\DVConverter> dvc -v -snort
\home\filters\test.rules -dvt \home\filters\dvt\test.rules.xml
arg = -v
arg = -snort
arg = \home\filters\test.rules
arg = -dvt
arg = c:\home\filters\dvt\test.rules.xml
```

```
Snort Rule Conversion Completed
end timestamp: 4/25/2016 10:32:31 PM
```