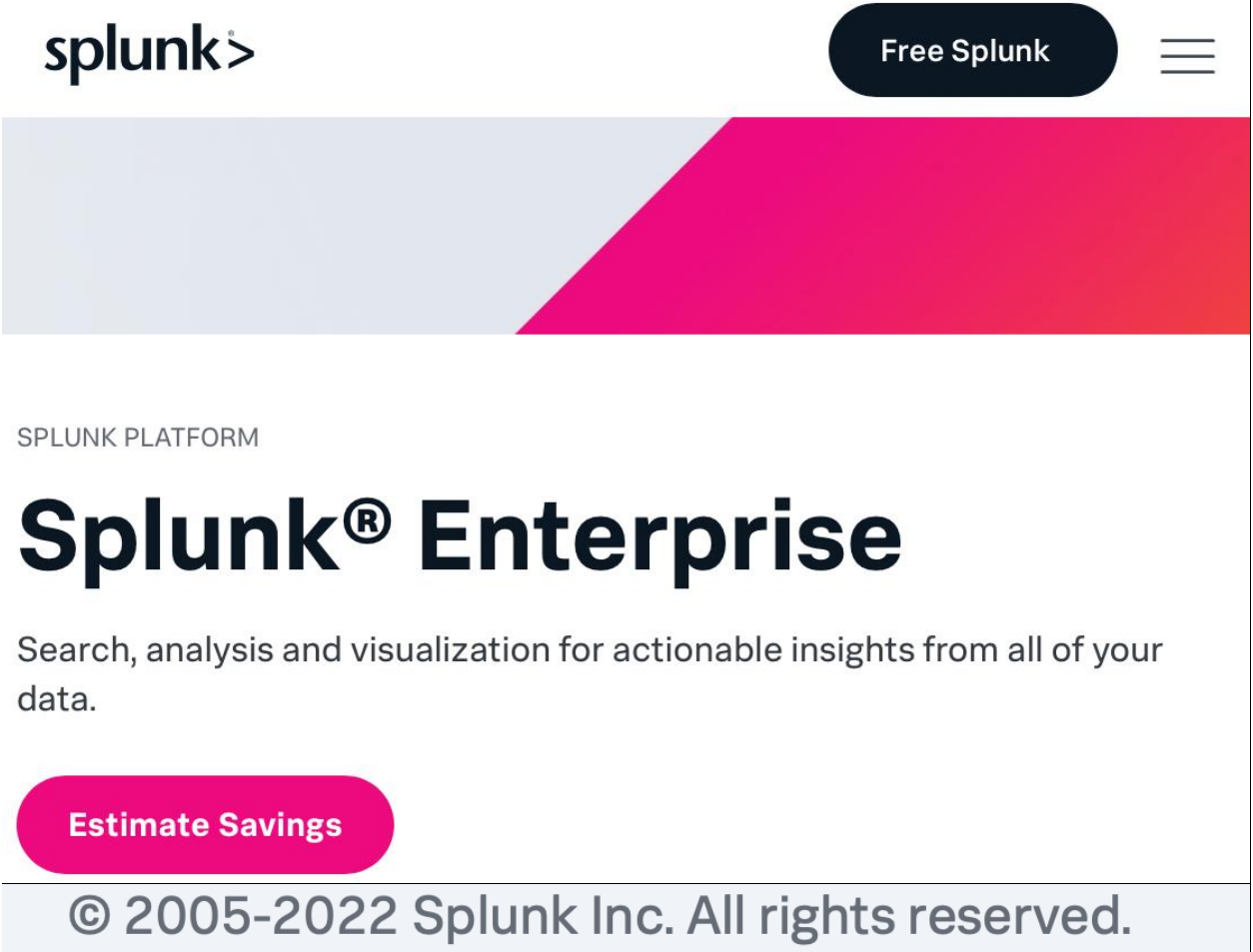
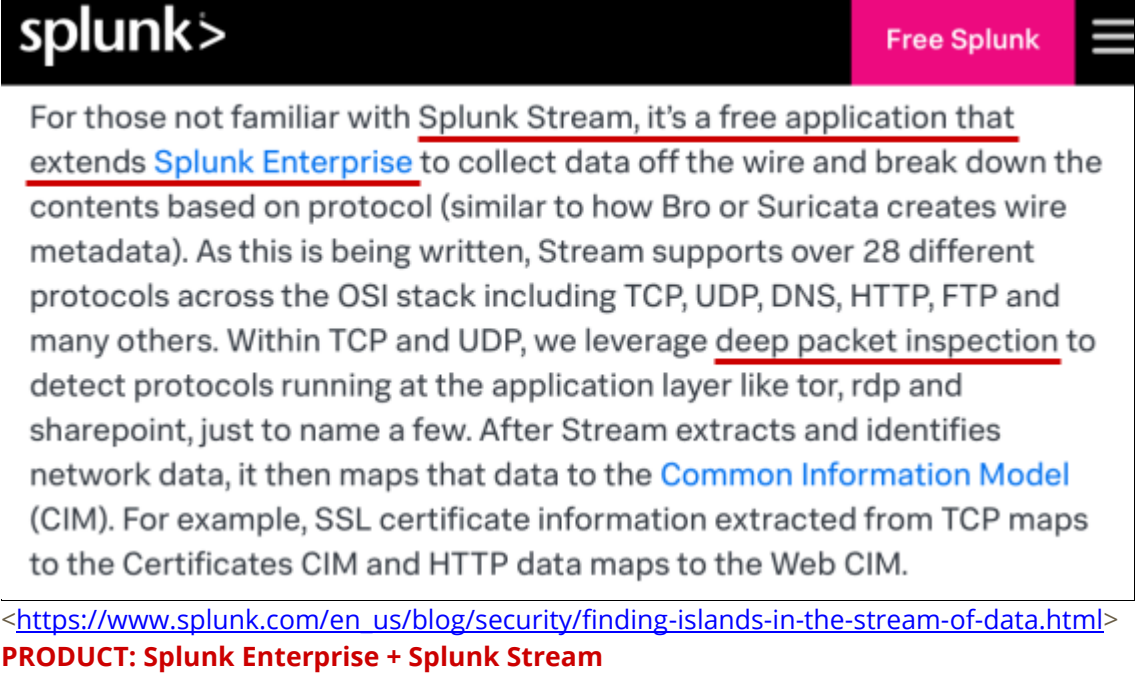


CLAIM CHARTS

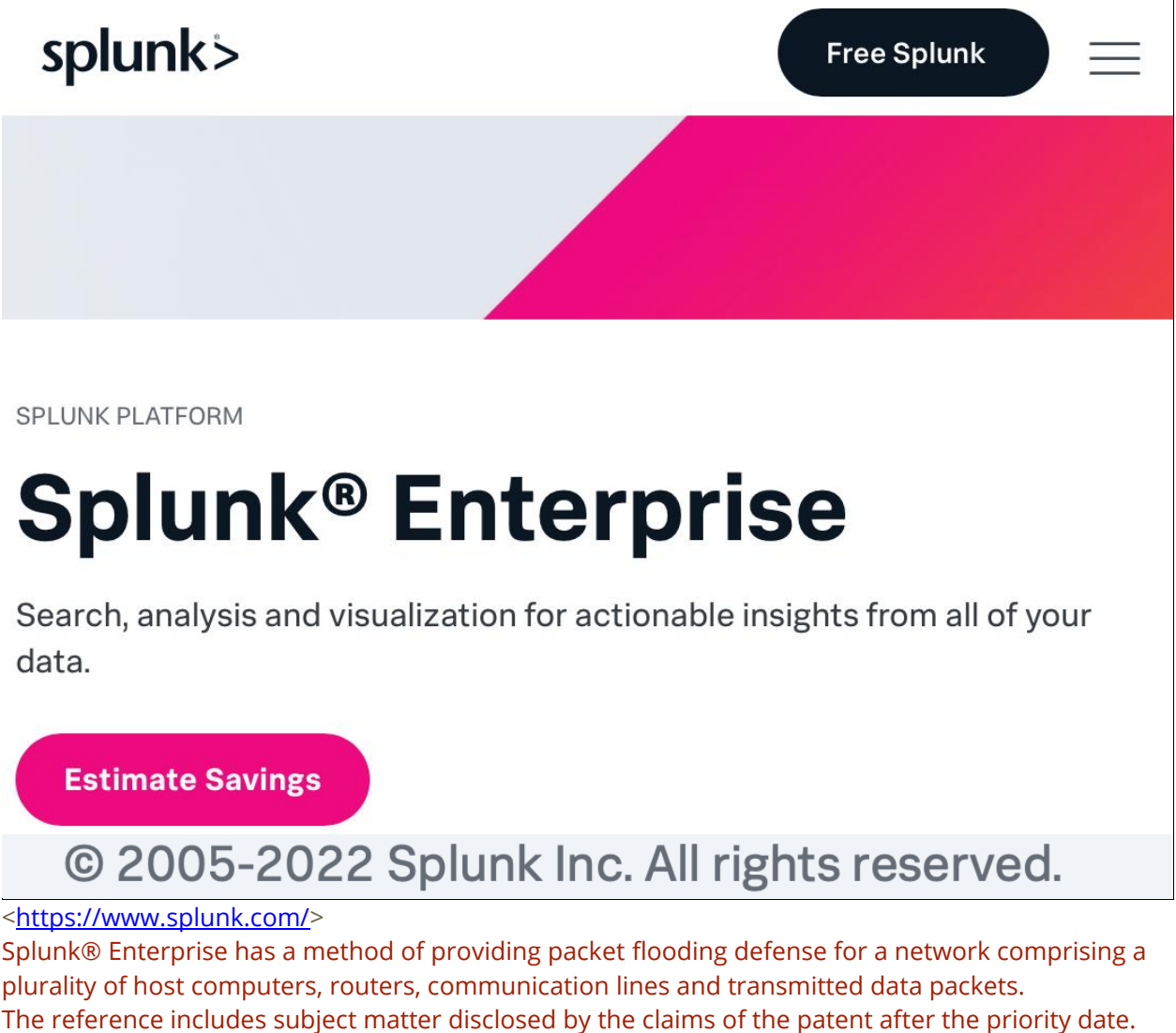
US7523497 B2 Claim 7	Splunk® Enterprise
7. A method of providing packet flooding defense for a network comprising a plurality of host computers, routers, communication lines and transmitted data packets, said method comprising the steps of:	 <https://www.splunk.com/> Splunk® Enterprise has a method of providing packet flooding defense for a network comprising a plurality of host computers, routers, communication lines and transmitted data packets. The reference includes subject matter disclosed by the claims of the patent after the priority date.

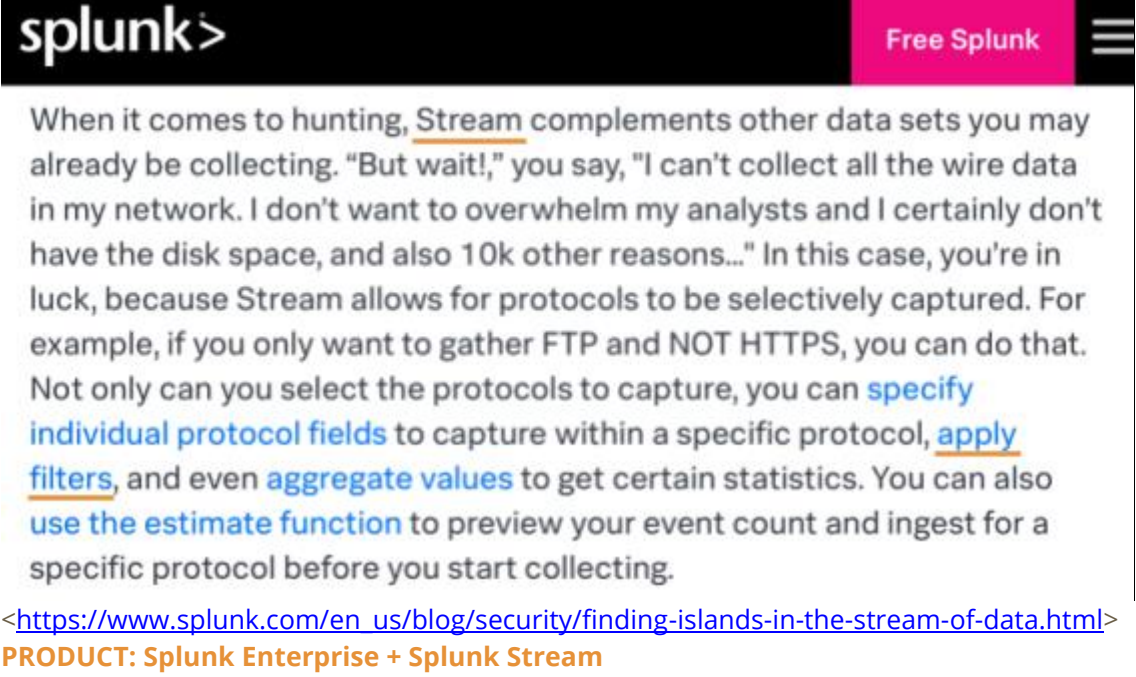
US7523497 B2 Claim 7	Splunk® Enterprise
determining a path by which data packets arrive at a host computer via packet marks provided by routers leading to said host computer; said path comprising all routers in said network via which said packets are routed to said computer;	 The screenshot shows the Splunk website header with the 'splunk>' logo and a pink 'Free Splunk' button. Below the header is a blog post snippet titled 'For those not familiar with Splunk Stream, it's a free application that extends Splunk Enterprise to collect data off the wire and break down the contents based on protocol (similar to how Bro or Suricata creates wire metadata). As this is being written, Stream supports over 28 different protocols across the OSI stack including TCP, UDP, DNS, HTTP, FTP and many others. Within TCP and UDP, we leverage deep packet inspection to detect protocols running at the application layer like tor, rdp and sharepoint, just to name a few. After Stream extracts and identifies network data, it then maps that data to the Common Information Model (CIM). For example, SSL certificate information extracted from TCP maps to the Certificates CIM and HTTP data maps to the Web CIM.' The snippet includes a URL: <https://www.splunk.com/en_us/blog/security/finding-islands-in-the-stream-of-data.html> and a product label: PRODUCT: Splunk Enterprise + Splunk Stream.
US7523497 B2 Claim 7	Splunk® Enterprise

	Deep Packet Inspection (DPI) is a fundamental technique used by firewalls to inspect headers and the payload of network packets before passing them down the network subject to security rules. <u>DPI provides information about the source and destination of the packet, the protocol, other IP and TCP/UDP header information, and the actual data.</u> In the Common Information Model, deep packet inspection data is typically mapped to the Network Traffic Data model. Visibility DPI provides raw information of everything transmitted over a network, including things that aren't necessarily part of or difficult to extract from a log, such as database query results. PCAP data can also be used to provide and identify: • DNS session analysis for malicious domain communications from each endpoint • Abnormal amounts of traffic or sessions • Abnormal amounts of domain and host communications • Known malicious traffic from a host • Expired SSL certification analysis • Abnormal host communications (internal and external) <https://lantern.splunk.com/Data_Descriptors/Data_Types/Network/Deep_packet_inspection_data> The reference describes determining a path by which data packets arrive at a host computer via packet marks provided by routers leading to said host computer; said path comprising all routers in said network via which said packets are routed to said computer..
US7523497 B2 Claim 7	Splunk® Enterprise
classifying data packets received at said host computer into wanted data packets and unwanted data packets by path;	splunk> Free Splunk When it comes to hunting, <u>Stream</u> complements other data sets you may already be collecting. "But wait!," you say, "I can't collect all the wire data in my network. I don't want to overwhelm my analysts and I certainly don't have the disk space, and also 10k other reasons..." In this case, you're in luck, because Stream allows for protocols to be selectively captured. For example, if you only want to gather FTP and NOT HTTPS, you can do that. Not only can you select the protocols to capture, you can <u>specify individual protocol fields</u> to capture within a specific protocol, <u>apply filters</u>, and even <u>aggregate values</u> to get certain statistics. You can also <u>use the estimate function</u> to preview your event count and ingest for a specific protocol before you start collecting.

	<https://www.splunk.com/en_us/blog/security/finding-islands-in-the-stream-of-data.html> PRODUCT: Splunk Enterprise + Splunk Stream
US7523497 B2 Claim 7	Splunk® Enterprise
	Deep Packet Inspection (DPI) is a fundamental technique used by firewalls to inspect headers and the payload of network packets before passing them down the network subject to security rules. DPI provides information about the source and destination of the packet, the protocol, other IP and TCP/UDP header information, and the actual data. In the Common Information Model, deep packet inspection data is typically mapped to the Network Traffic Data model. Visibility <u>DPI provides raw information of everything transmitted over a network</u>, including things that aren't necessarily part of or difficult to extract from a log, such as database query results. PCAP data can also be used to provide and identify: • DNS session analysis for malicious domain communications from each endpoint • <u>Abnormal amounts of traffic or sessions</u> • Abnormal amounts of domain and host communications • Known malicious traffic from a host • Expired SSL certification analysis • Abnormal host communications (internal and external) <https://lantern.splunk.com/Data_Descriptors/Data_Types/Network/Deep_packet_inspection_data> The reference describes classifying data packets received at said host computer into wanted data packets and unwanted data packets by path.

US7523497 B2 Claim 7	Splunk® Enterprise
associating a maximum acceptable processing rate with each class of data packet received at said host computer; and	<pre> maxSockets = <integer> * The number of HTTP connections that the HTTP event collector input accepts simultaneously. * Set this setting to constrain resource usage. * If you set this setting to 0, the input automatically sets it to one third of the maximum allowable open files on the host. * If this value is less than 50, the input sets it to 50. If this value is greater than 400000, the input sets it to 400000. * If set to a negative value, the input does not enforce a limit on connections. * Default: 0 maxThreads = <integer> * The number of threads that can be used by active HTTP transactions. * Set this to constrain resource usage. * If you set this setting to 0, the input automatically sets the limit to one third of the maximum allowable threads on the host. * If this value is less than 20, the input sets it to 20. If this value is greater than 150000, the input sets it to 150000. * If the 'maxSockets' setting has a positive value and 'maxThreads' is greater than 'maxSockets', then the input sets 'maxThreads' to be equal to 'maxSockets'. * If set to a negative value, the input does not enforce a limit on threads. * Default: 0 </pre> <https://docs.splunk.com/Documentation/Splunk/8.2.6/Admin/Inputsconf> The reference describes associating a maximum acceptable processing rate with each class of data packet received at said host computer. PRODUCT: Splunk Enterprise + Splunk Stream
US7523497 B2 Claim 7	Splunk® Enterprise
allocating a processing rate less than or equal to said maximum acceptable processing rate for unwanted data packets.	<pre> sslCommonNameToCheck = <commonName1>, <commonName2>, ... * A list of SSL Common Names to match against certificates that incoming HTTPS connections present to this instance. * If you configure this setting and also set 'requireClientCert' to "true", splunkd limits most inbound HTTPS connections to hosts that use a cert with one of the listed common names. * The most important scenario to use this setting is distributed search. * This feature does not work with the deployment server and client communication over SSL. * This setting is optional. * Default: empty string (no common name checking) </pre> <https://docs.splunk.com/Documentation/Splunk/8.2.6/Admin/Inputsconf> The reference describes allocating a processing rate less than or equal to said maximum acceptable processing rate for unwanted data packets. PRODUCT: Splunk Enterprise + Splunk Stream

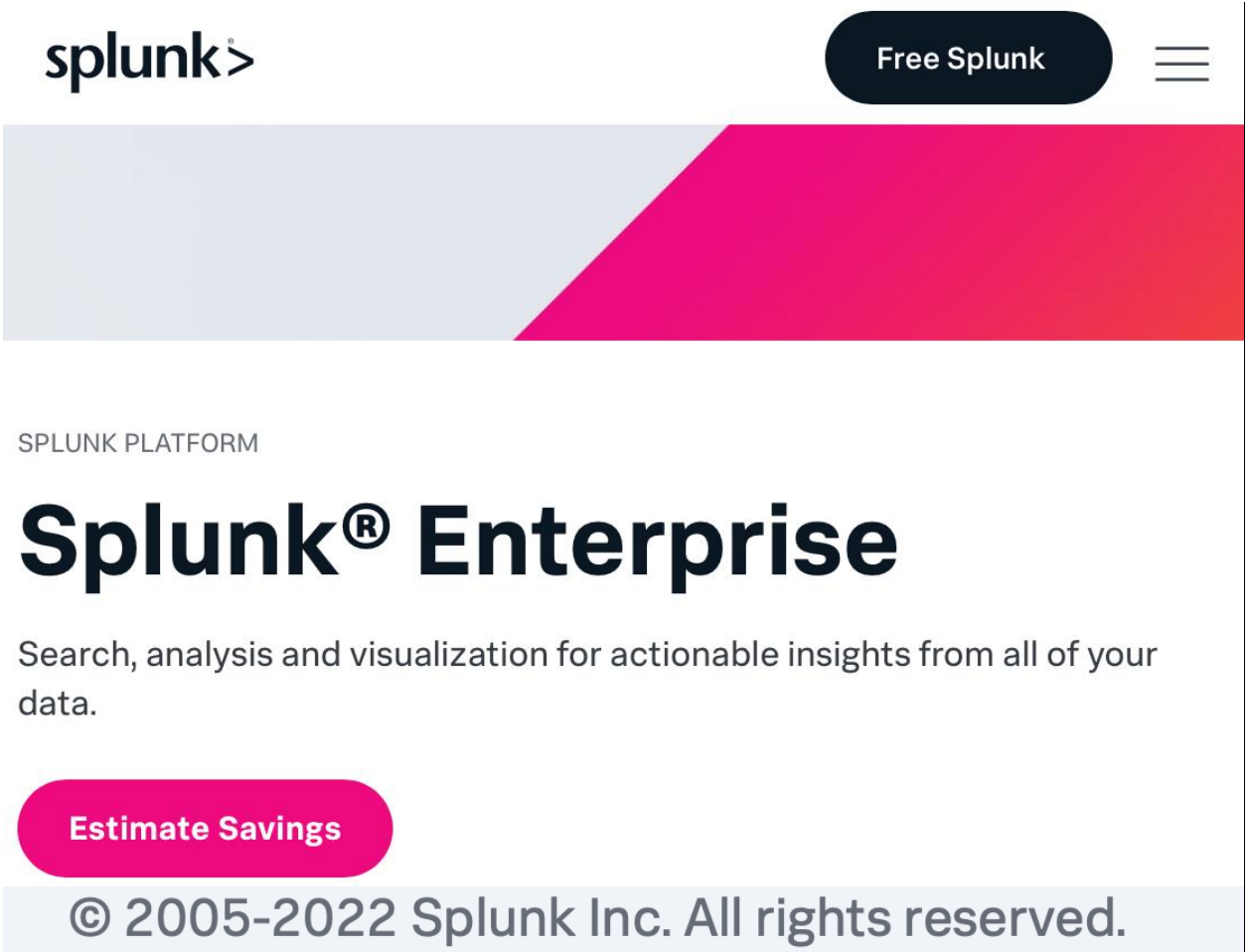
US7523497 B2 Claim 8	Splunk® Enterprise
8. A method of providing packet flooding defense for a network comprising a plurality of host computers, routers, communication lines and transmitted data packets, said method comprising the steps of:	 <https://www.splunk.com/> Splunk® Enterprise has a method of providing packet flooding defense for a network comprising a plurality of host computers, routers, communication lines and transmitted data packets. The reference includes subject matter disclosed by the claims of the patent after the priority date.

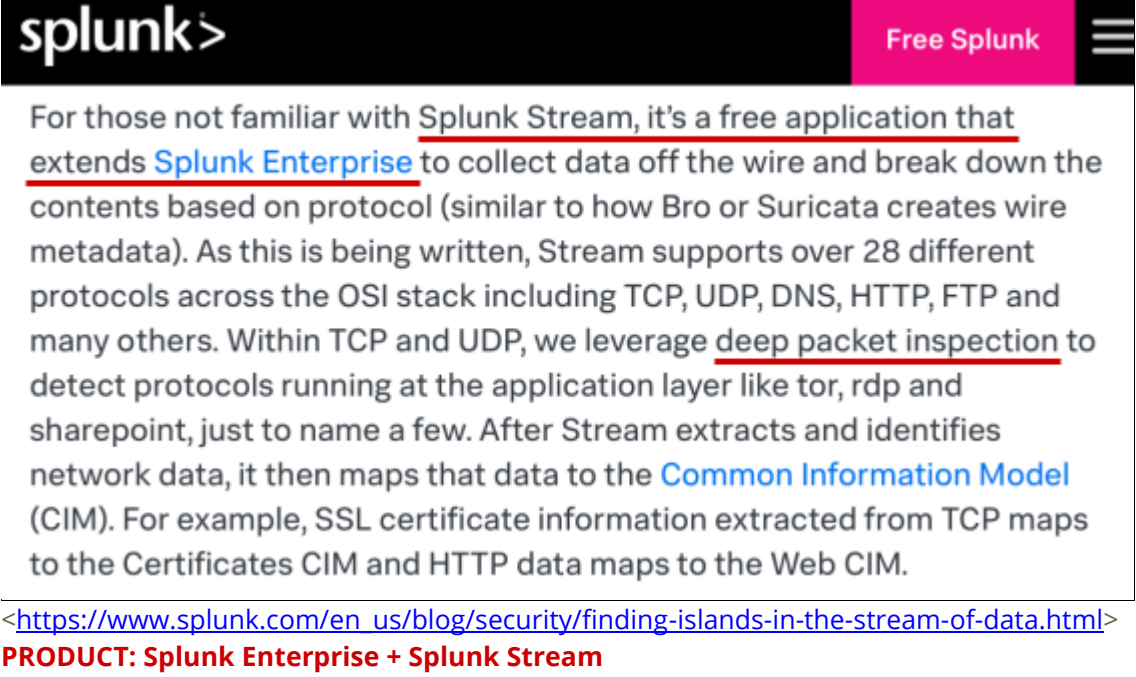
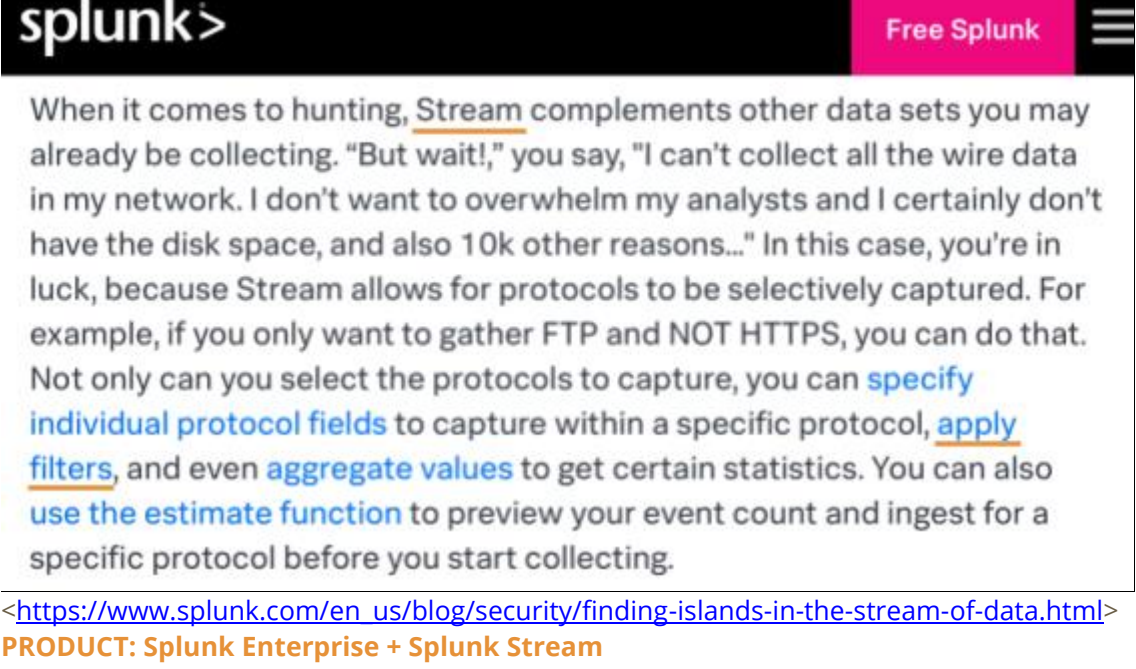
US7523497 B2 Claim 8	Splunk® Enterprise
classifying data packets received at a host computer into wanted data packets and unwanted data packets; said data packets comprising data packets from all routers in said network via which said data packets are routed to said computer;	 The screenshot shows the Splunk website header with the 'splunk>' logo and a 'Free Splunk' button. Below the header, a blog post snippet is visible, discussing the 'Stream' feature and its ability to selectively capture data based on protocols and fields. The text includes phrases like 'When it comes to hunting, <u>Stream</u> complements other data sets you may already be collecting.' and 'Not only can you select the protocols to capture, you can specify individual protocol fields to capture within a specific protocol, apply filters, and even aggregate values to get certain statistics.' The snippet ends with a URL: '<https://www.splunk.com/en_us/blog/security/finding-islands-in-the-stream-of-data.html>'. Below the snippet, the text 'PRODUCT: Splunk Enterprise + Splunk Stream' is displayed.
US7523497 B2 Claim 8	Splunk® Enterprise

associating a maximum acceptable processing rate with each class of data packet received at said computer;	<pre> maxSockets = <integer> * The number of HTTP connections that the HTTP event collector input accepts simultaneously. * Set this setting to constrain resource usage. * If you set this setting to 0, the input automatically sets it to one third of the maximum allowable open files on the host. * If this value is less than 50, the input sets it to 50. If this value is greater than 400000, the input sets it to 400000. * If set to a negative value, the input does not enforce a limit on connections. * Default: 0 maxThreads = <integer> * The number of threads that can be used by active HTTP transactions. * Set this to constrain resource usage. * If you set this setting to 0, the input automatically sets the limit to one third of the maximum allowable threads on the host. * If this value is less than 20, the input sets it to 20. If this value is greater than 150000, the input sets it to 150000. * If the 'maxSockets' setting has a positive value and 'maxThreads' is greater than 'maxSockets', then the input sets 'maxThreads' to be equal to 'maxSockets'. * If set to a negative value, the input does not enforce a limit on threads. * Default: 0 </pre> <https://docs.splunk.com/Documentation/Splunk/8.2.6/Admin/Inputsconf> The reference describes associating a maximum acceptable processing rate with each class of data packet received at said host computer. PRODUCT: Splunk Enterprise + Splunk Stream
US7523497 B2 Claim 8	Splunk® Enterprise
determining the rate at which data packets of each class are transmitted from a router to said computer;	<pre> sslCommonNameToCheck = <commonName1>, <commonName2>, ... * A list of SSL Common Names to match against certificates that incoming HTTPS connections present to this instance. * If you configure this setting and also set 'requireClientCert' to "true", splunkd limits most inbound HTTPS connections to hosts that use a cert with one of the listed common names. * The most important scenario to use this setting is distributed search. * This feature does not work with the deployment server and client communication over SSL. * This setting is optional. * Default: empty string (no common name checking) </pre> <https://docs.splunk.com/Documentation/Splunk/8.2.6/Admin/Inputsconf> The reference describes allocating a processing rate less than or equal to said maximum acceptable processing rate for unwanted data packets. PRODUCT: Splunk Enterprise + Splunk Stream

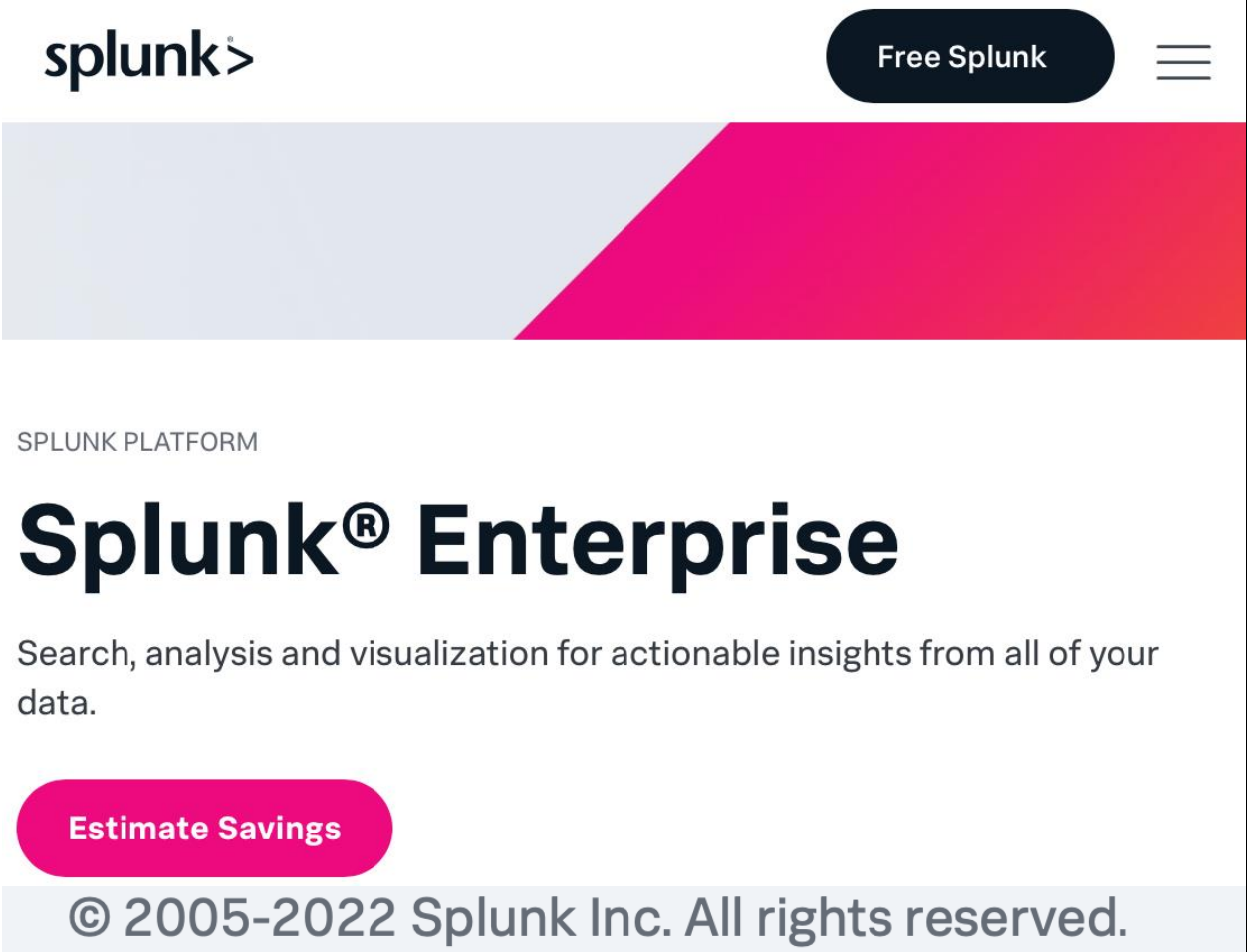
US7523497 B2 Claim 8	Splunk® Enterprise
receiving a maximum acceptable transmission rate for data packets being transmitted to said computer in said router; and	<pre> sslCommonNameToCheck = <commonName1>, <commonName2>, ... * A list of SSL Common Names to match against certificates that incoming HTTPS connections present to this instance. * If you configure this setting and also set 'requireClientCert' to "true", <u>splunkd limits most inbound HTTPS connections to hosts that use a cert with one of the listed common names.</u> * The most important scenario to use this setting is distributed search. * This feature does not work with the deployment server and client communication over SSL. * This setting is optional. * Default: empty string (no common name checking) </pre> <https://docs.splunk.com/Documentation/Splunk/8.2.6/Admin/Inputsconf> The reference describes allocating a processing rate less than or equal to said maximum acceptable processing rate for unwanted data packets. PRODUCT: Splunk Enterprise + Splunk Stream
US7523497 B2 Claim 8	Splunk® Enterprise
controlling the rate of transmission of data packets from said router to said computer by said router so that data packet transmissions received at said computer are kept below the maximum acceptable processing rate for each data packet class; and	<pre> sslCommonNameToCheck = <commonName1>, <commonName2>, ... * A list of SSL Common Names to match against certificates that incoming HTTPS connections present to this instance. * If you configure this setting and also set 'requireClientCert' to "true", <u>splunkd limits most inbound HTTPS connections to hosts that use a cert with one of the listed common names.</u> * The most important scenario to use this setting is distributed search. * This feature does not work with the deployment server and client communication over SSL. * This setting is optional. * Default: empty string (no common name checking) </pre> <https://docs.splunk.com/Documentation/Splunk/8.2.6/Admin/Inputsconf> The reference describes allocating a processing rate less than or equal to said maximum acceptable processing rate for unwanted data packets. PRODUCT: Splunk Enterprise + Splunk Stream

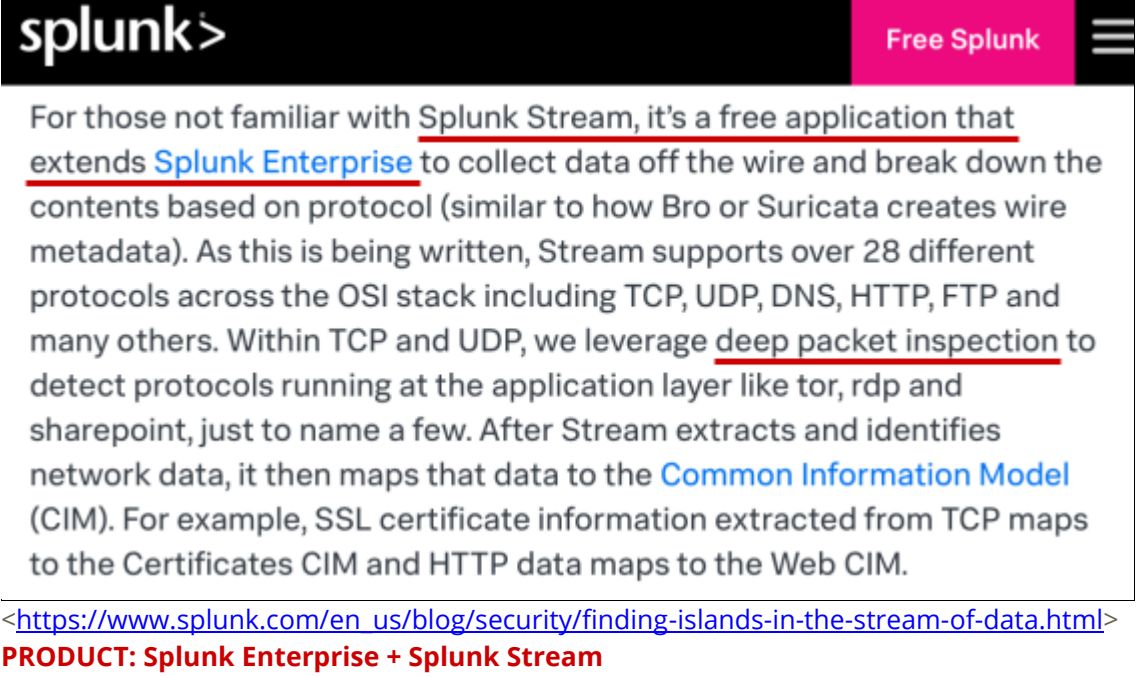
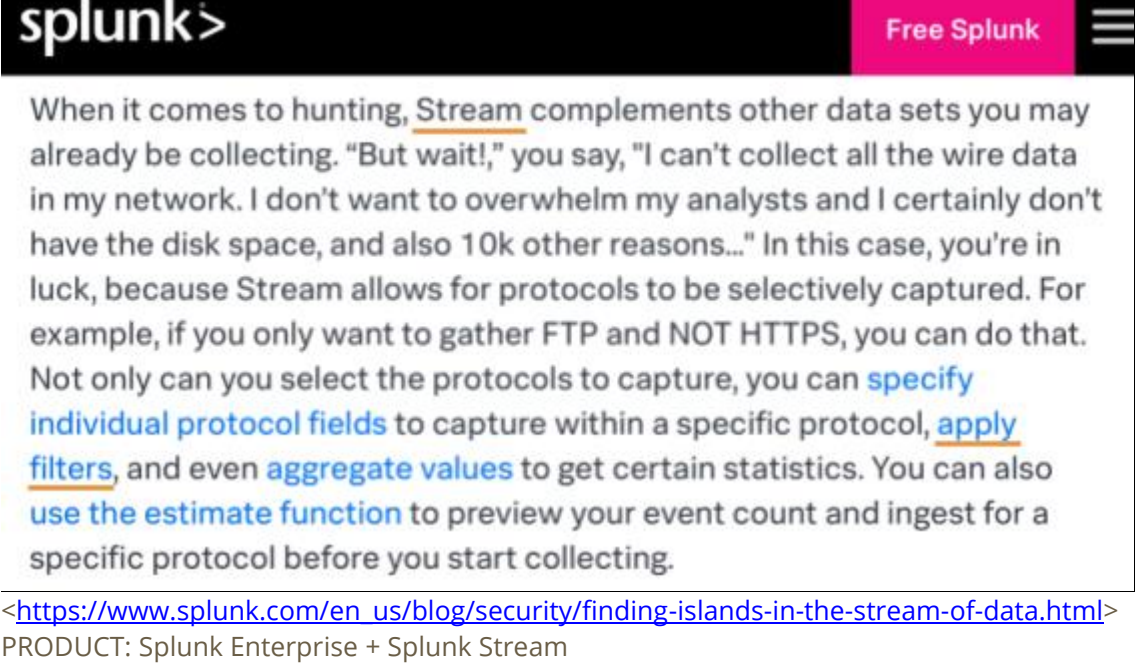
US7523497 B2 Claim 8	Splunk® Enterprise
freeing a portion of the network providing data packet transmission to said computer.	<pre> maxSockets = <integer> * The number of HTTP connections that the HTTP event collector input accepts simultaneously. * Set this setting to constrain resource usage. * If you set this setting to 0, the input automatically sets it to one third of the maximum allowable open files on the host. * If this value is less than 50, the input sets it to 50. If this value is greater than 400000, the input sets it to 400000. * If set to a negative value, the input does not enforce a limit on connections. * Default: 0 maxThreads = <integer> * The number of threads that can be used by active HTTP transactions. * Set this to constrain resource usage. * If you set this setting to 0, the input automatically sets the limit to one third of the maximum allowable threads on the host. * If this value is less than 20, the input sets it to 20. If this value is greater than 150000, the input sets it to 150000. * If the 'maxSockets' setting has a positive value and 'maxThreads' is greater than 'maxSockets', then the input sets 'maxThreads' to be equal to 'maxSockets'. * If set to a negative value, the input does not enforce a limit on threads. * Default: 0 </pre> https://docs.splunk.com/Documentation/Splunk/8.2.6/Admin/Inputsconf The reference describes associating a maximum acceptable processing rate with each class of data packet received at said host computer. PRODUCT: Splunk Enterprise + Splunk Stream

US7523497 B2 Claim 10	Splunk® Enterprise
10. A method of providing packet flooding defense for a network comprising a plurality of host computers, routers, communication lines and transmitted data packets, said method comprising the steps of:	 <https://www.splunk.com/> Splunk® Enterprise has a method of providing packet flooding defense for a network comprising a plurality of host computers, routers, communication lines and transmitted data packets. The reference includes subject matter disclosed by the claims of the patent after the priority date.

US7523497 B2 Claim 10	Splunk® Enterprise
determining a path by which data packets arrive at said router via packet marks provided by routers leading to said host computer; said path comprising all routers in said network via which said packets are routed to said computer;	 The screenshot shows the Splunk Stream product page. At the top is the Splunk logo and a 'Free Splunk' button. The main text describes Splunk Stream as a free application that extends Splunk Enterprise to collect data off the wire and break down the contents based on protocol. It lists supported protocols (TCP, UDP, DNS, HTTP, FTP) and mentions deep packet inspection for detecting protocols at the application layer. A URL is provided at the bottom: <https://www.splunk.com/en_us/blog/security/finding-islands-in-the-stream-of-data.html>. PRODUCT: Splunk Enterprise + Splunk Stream
US7523497 B2 Claim 10	Splunk® Enterprise
classifying data packets received at said router via packet marks provided by routers leading to said host computer by path;	 The screenshot shows the Splunk Stream product page. At the top is the Splunk logo and a 'Free Splunk' button. The main text describes how Stream complements other data sets by allowing selective capture of protocols. It provides an example of capturing FTP but not HTTPS, and mentions the ability to specify individual protocol fields, apply filters, and use the estimate function. A URL is provided at the bottom: <https://www.splunk.com/en_us/blog/security/finding-islands-in-the-stream-of-data.html>. PRODUCT: Splunk Enterprise + Splunk Stream

US7523497 B2 Claim 10	Splunk® Enterprise
associating a maximum acceptable transmission rate with each class of data packet received at said router; and	<pre> maxSockets = <integer> * The number of HTTP connections that the HTTP event collector input accepts simultaneously. * Set this setting to constrain resource usage. * If you set this setting to 0, the input automatically sets it to one third of the maximum allowable open files on the host. * If this value is less than 50, the input sets it to 50. If this value is greater than 400000, the input sets it to 400000. * If set to a negative value, the input does not enforce a limit on connections. * Default: 0 maxThreads = <integer> * The number of threads that can be used by active HTTP transactions. * Set this to constrain resource usage. * If you set this setting to 0, the input automatically sets the limit to one third of the maximum allowable threads on the host. * If this value is less than 20, the input sets it to 20. If this value is greater than 150000, the input sets it to 150000. * If the 'maxSockets' setting has a positive value and 'maxThreads' is greater than 'maxSockets', then the input sets 'maxThreads' to be equal to 'maxSockets'. * If set to a negative value, the input does not enforce a limit on threads. * Default: 0 </pre> <https://docs.splunk.com/Documentation/Splunk/8.2.6/Admin/Inputsconf> The reference describes associating a maximum acceptable processing rate with each class of data packet received at said host computer. PRODUCT: Splunk Enterprise + Splunk Stream
US7523497 B2 Claim 10	Splunk® Enterprise
allocating a transmission rate equal to or less than said maximum acceptable transmission rate for unwanted data packets.	<pre> sslCommonNameToCheck = <commonName1>, <commonName2>, ... * A list of SSL Common Names to match against certificates that incoming HTTPS connections present to this instance. * If you configure this setting and also set 'requireClientCert' to "true", splunkd limits most inbound HTTPS connections to hosts that use a cert with one of the listed common names. * The most important scenario to use this setting is distributed search. * This feature does not work with the deployment server and client communication over SSL. * This setting is optional. * Default: empty string (no common name checking) </pre> <https://docs.splunk.com/Documentation/Splunk/8.2.6/Admin/Inputsconf> The reference describes allocating a processing rate less than or equal to said maximum acceptable processing rate for unwanted data packets. PRODUCT: Splunk Enterprise + Splunk Stream

US7523497 B2 Claim 16	Splunk® Enterprise
16. A packet flooding defense system for a network comprising a plurality of host computers, routers, communication lines and transmitted data packets, said system comprising:	 <https://www.splunk.com/> Splunk® Enterprise has a method of providing packet flooding defense for a network comprising a plurality of host computers, routers, communication lines and transmitted data packets. The reference includes subject matter disclosed by the claims of the patent after the priority date.

US7523497 B2 Claim 16	Splunk® Enterprise
means for a router to determine a path by which said packets came to said router via packet marks provided by routers leading to said router; said path comprising all routers in said network via which said packets are routed to said computer;	 The screenshot shows the Splunk logo and a navigation bar with 'Free Splunk' and a menu icon. The main text discusses Splunk Stream as a free application that extends Splunk Enterprise to collect data off the wire and break down contents based on protocol. It mentions supporting over 28 different protocols across the OSI stack, including TCP, UDP, DNS, HTTP, FTP, and many others. It also mentions leveraging deep packet inspection to detect protocols running at the application layer like tor, rdp, and sharepoint. The text concludes by stating that after Stream extracts and identifies network data, it maps that data to the Common Information Model (CIM). For example, SSL certificate information extracted from TCP maps to the Certificates CIM and HTTP data maps to the Web CIM. <https://www.splunk.com/en_us/blog/security/finding-islands-in-the-stream-of-data.html> PRODUCT: Splunk Enterprise + Splunk Stream
US7523497 B2 Claim 16	Splunk® Enterprise
means for classifying data packets received at said router into wanted data packets and unwanted data packets by path;	 The screenshot shows the Splunk logo and a navigation bar with 'Free Splunk' and a menu icon. The main text discusses how Stream complements other data sets you may already be collecting. It quotes a user saying, 'But wait!', and explains that Stream allows for protocols to be selectively captured. For example, if you only want to gather FTP and NOT HTTPS, you can do that. Not only can you select the protocols to capture, you can specify individual protocol fields to capture within a specific protocol, apply filters, and even aggregate values to get certain statistics. You can also use the estimate function to preview your event count and ingest for a specific protocol before you start collecting. <https://www.splunk.com/en_us/blog/security/finding-islands-in-the-stream-of-data.html> PRODUCT: Splunk Enterprise + Splunk Stream

US7523497 B2 Claim 16	Splunk® Enterprise
means for associating a maximum acceptable transmission rate with each class of data packet received at said router; and	<pre> maxSockets = <integer> * The number of HTTP connections that the HTTP event collector input accepts simultaneously. * Set this setting to constrain resource usage. * If you set this setting to 0, the input automatically sets it to one third of the maximum allowable open files on the host. * If this value is less than 50, the input sets it to 50. If this value is greater than 400000, the input sets it to 400000. * If set to a negative value, the input does not enforce a limit on connections. * Default: 0 maxThreads = <integer> * The number of threads that can be used by active HTTP transactions. * Set this to constrain resource usage. * If you set this setting to 0, the input automatically sets the limit to one third of the maximum allowable threads on the host. * If this value is less than 20, the input sets it to 20. If this value is greater than 150000, the input sets it to 150000. * If the 'maxSockets' setting has a positive value and 'maxThreads' is greater than 'maxSockets', then the input sets 'maxThreads' to be equal to 'maxSockets'. * If set to a negative value, the input does not enforce a limit on threads. * Default: 0 </pre> <https://docs.splunk.com/Documentation/Splunk/8.2.6/Admin/Inputsconf> The reference describes associating a maximum acceptable processing rate with each class of data packet received at said host computer. PRODUCT: Splunk Enterprise + Splunk Stream
US7523497 B2 Claim 16	Splunk® Enterprise <pre> sslCommonNameToCheck = <commonName1>, <commonName2>, ... * A list of SSL Common Names to match against certificates that incoming HTTPS connections present to this instance. * If you configure this setting and also set 'requireClientCert' to "true", splunkd limits most inbound HTTPS connections to hosts that use a cert with one of the listed common names. * The most important scenario to use this setting is distributed search. * This feature does not work with the deployment server and client communication over SSL. * This setting is optional. * Default: empty string (no common name checking) </pre> <https://docs.splunk.com/Documentation/Splunk/8.2.6/Admin/Inputsconf> The reference describes allocating a processing rate less than or equal to said maximum acceptable processing rate for unwanted data packets. PRODUCT: Splunk Enterprise + Splunk Stream