UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CISCO SYSTEMS, INC.,
Petitioner,
- vs. -
CENTRIPETAL NETWORKS, INC.,
Patent Owner

Case No.: IPR2018-01512

US Patent 9,565,213

PETITIONER'S DEMONSTRATIVE EXHIBITS

PTAB ORAL ARGUMENT

December 18, 2019

What is claimed is:

1. A method comprising:

receiving, by each of a plurality of packet security gateways associated with a security policy management server and from the security policy management server, a dynamic security policy that comprises at least one rule specifying application-layer packet-header information and a packet transformation function comprising a packet digest logging function to be performed on packets comprising the application-layer packet-header information;

receiving, by a packet security gateway of the plurality of packet security gateways, packets associated with a network protected by the packet security gateway;

identifying, by the packet security gateway, from amongst the packets associated with the network protected by the packet security gateway, and on a packet-by-packet basis, one or more packets comprising the application-layer packet-header information;

performing, by the packet security gateway and on a packet-by-packet basis, the packet transformation function on each of the one or more packets comprising the application-layer packet-header information, wherein the performing the packet transformation function comprises

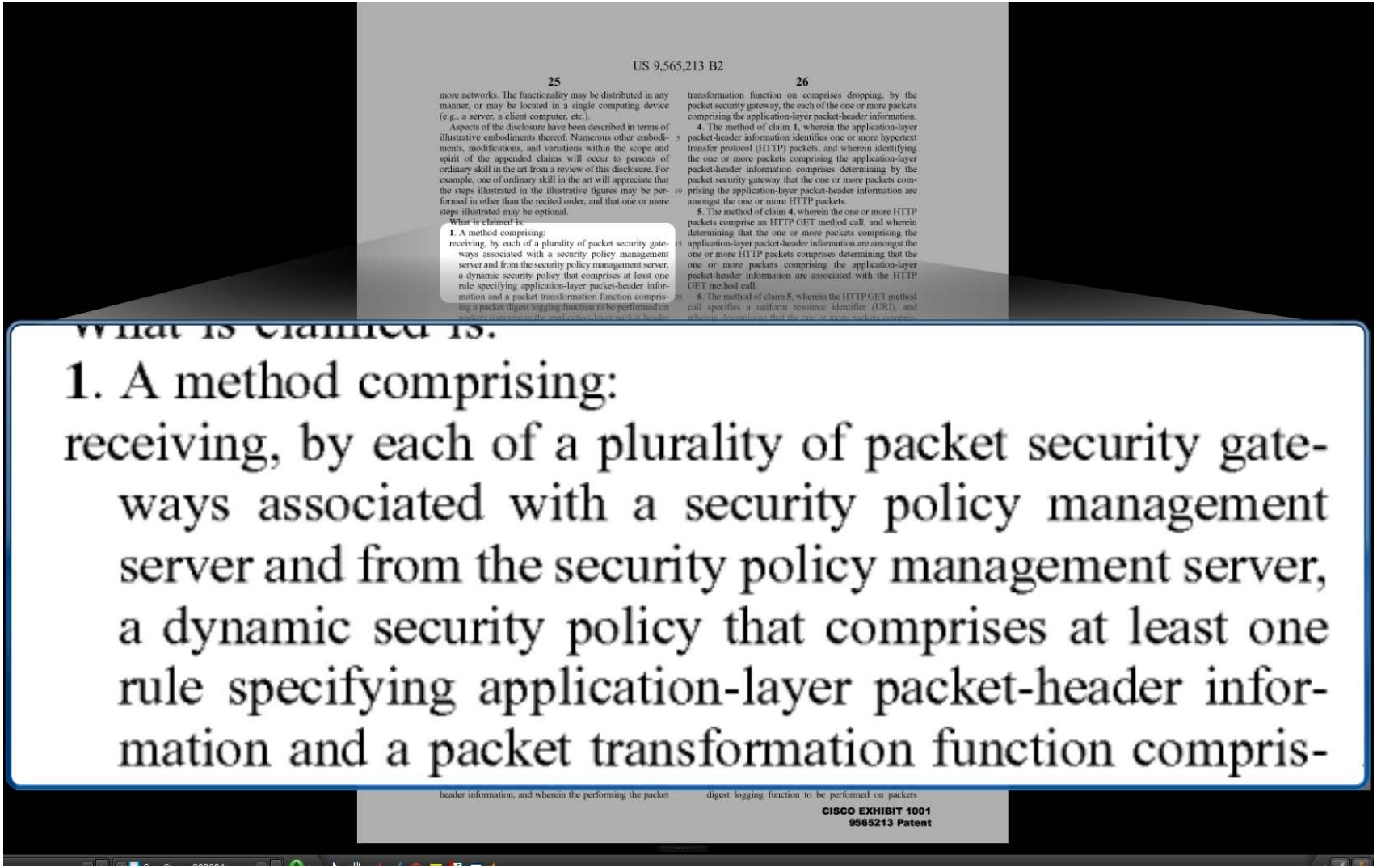
identifying a subset of information specified by the packet digest logging function for each of the one or more packets comprising the application-layer packet-header information;

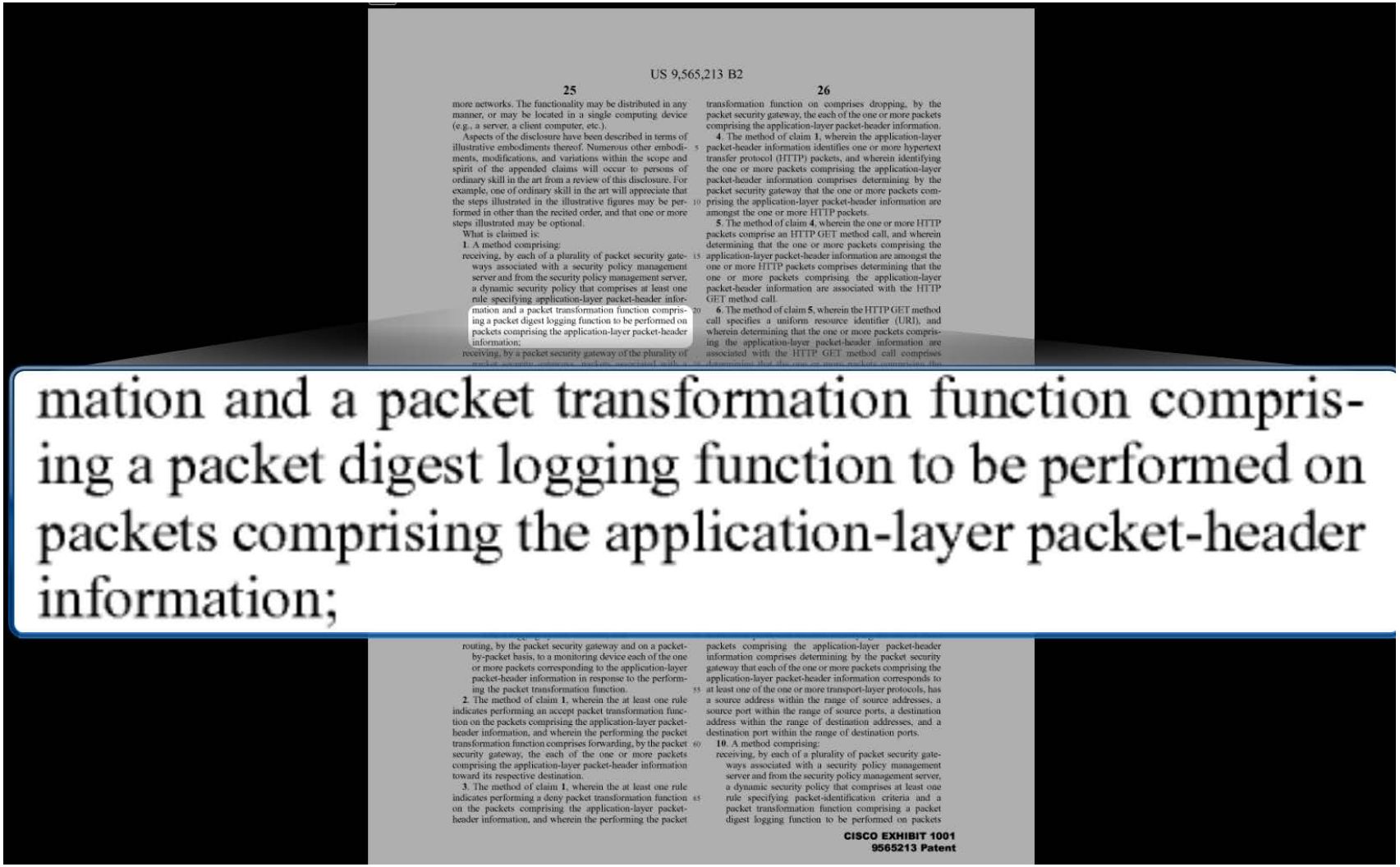
generating, for each of the one or more packets comprising the application-layer packet-header information, a record comprising the subset of information specified by the packet digest logging function; and

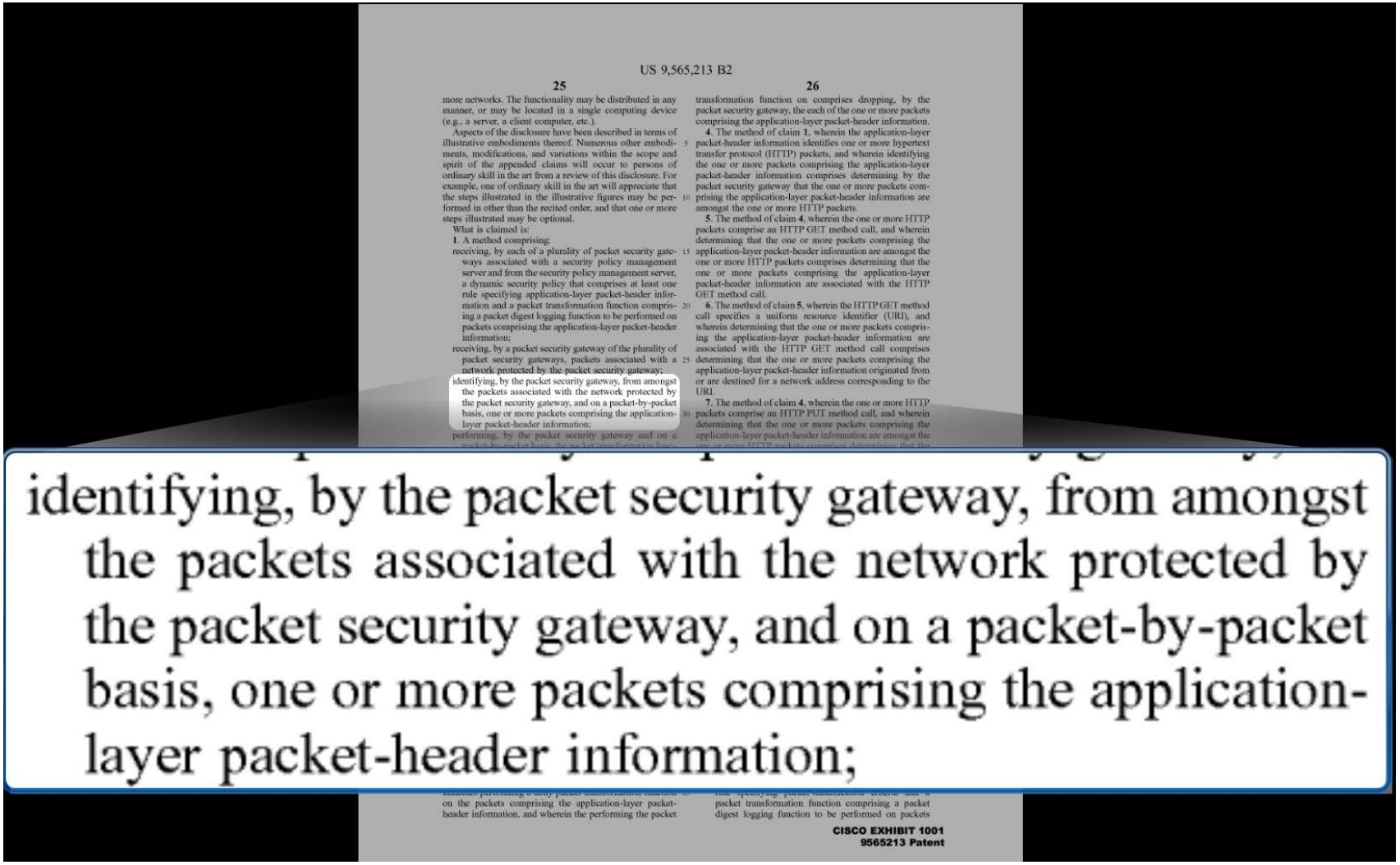
reformatting, for each of the one or more packets comprising the application-layer packet-header information, the subset of information specified by the packet digest logging function in accordance with a logging system standard; and

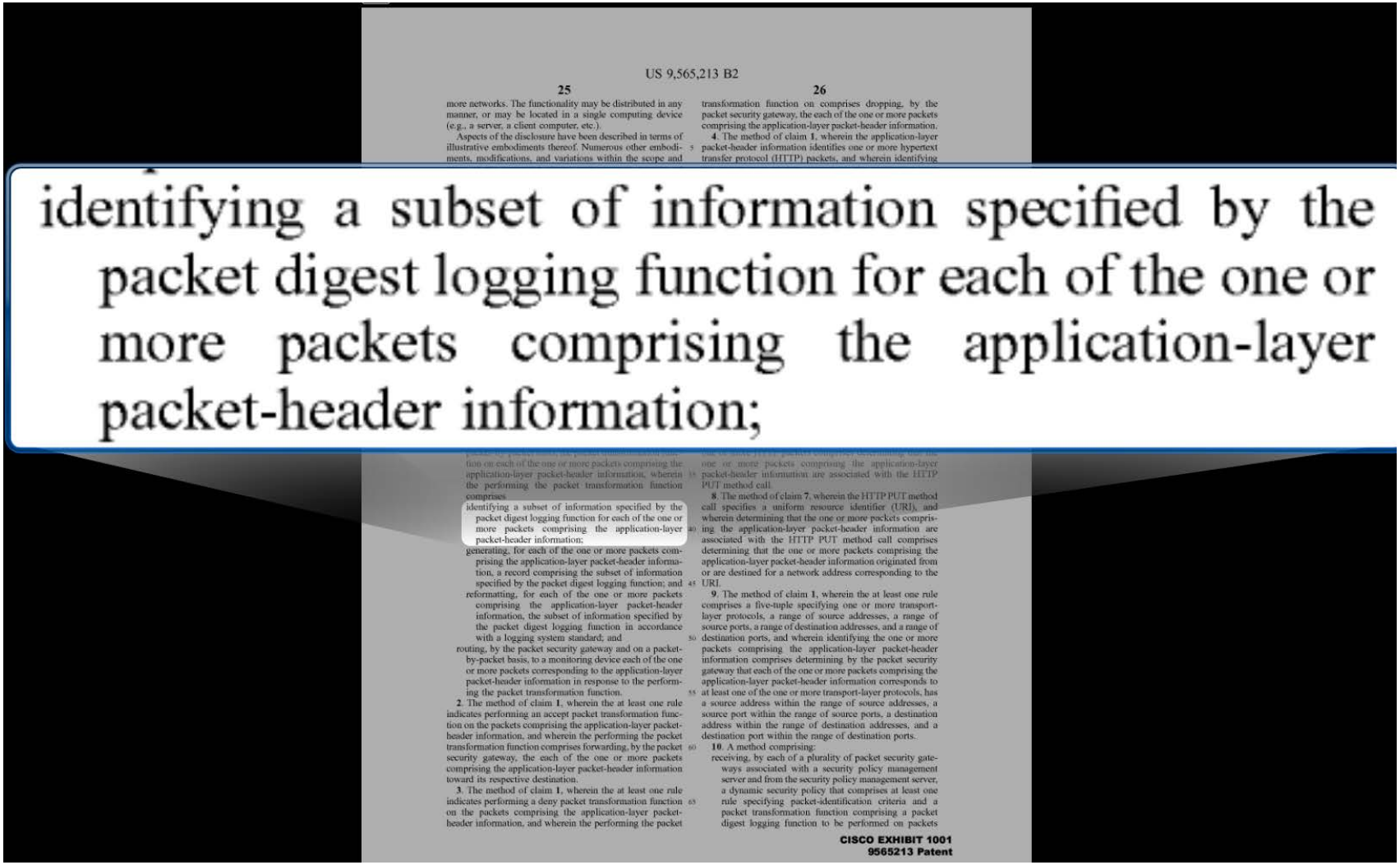
routing, by the packet security gateway and on a packet-by-packet basis, to a monitoring device each of the one or more packets corresponding to the application-layer packet-header information in response to the performing the packet transformation function.

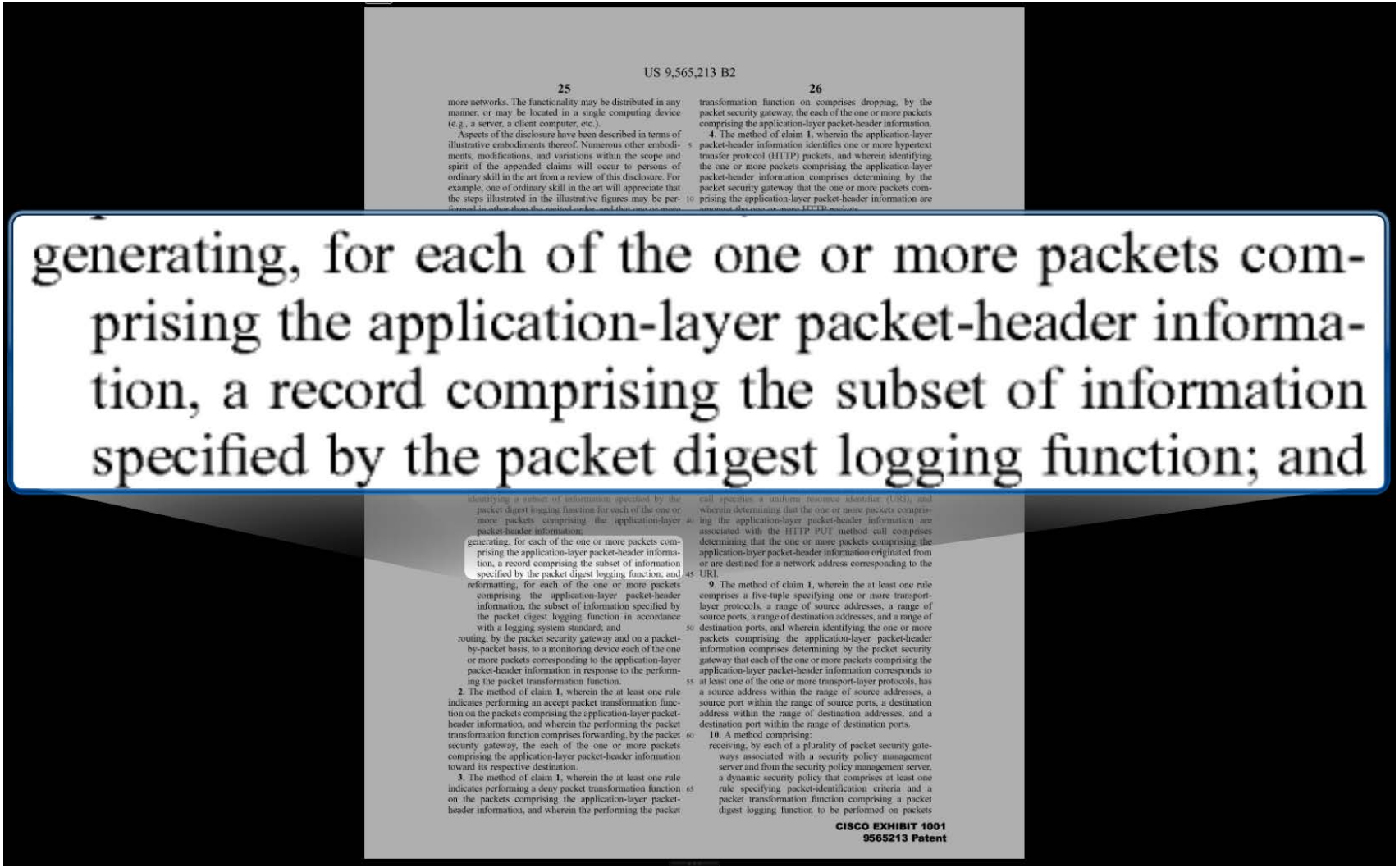
CISCO EXHIBIT 1001
9565213 Patent

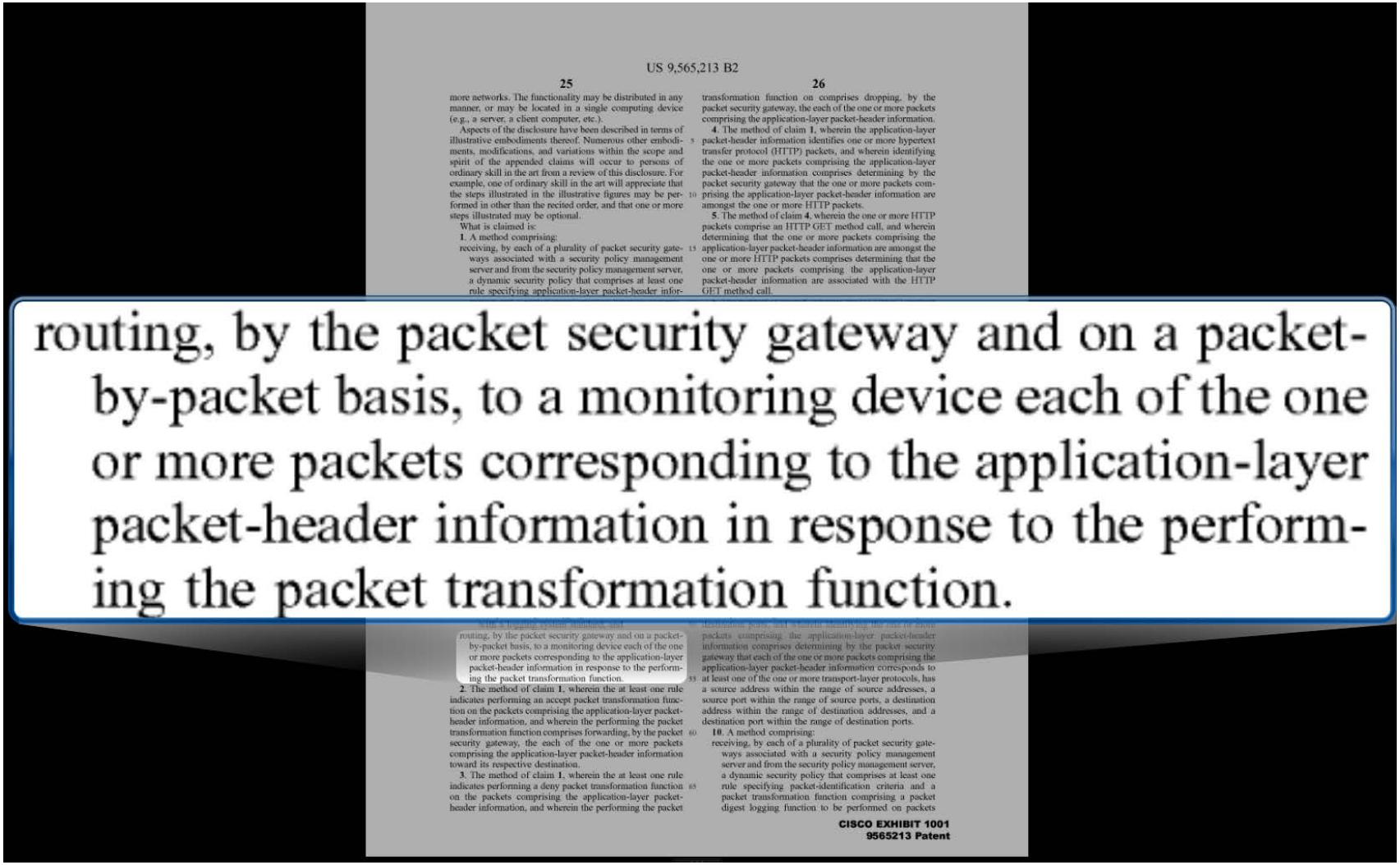












more networks. The functionality may be distributed in any manner, or may be located in a single computing device (e.g., a server, a client computer, etc.).

Aspects of the disclosure have been described in terms of illustrative embodiments thereof. Numerous other embodiments, modifications, and variations within the scope and spirit of the appended claims will occur to persons of ordinary skill in the art from a review of this disclosure. For example, one of ordinary skill in the art will appreciate that the steps illustrated in the illustrative figures may be performed in other than the recited order, and that one or more steps illustrated may be optional.

What is claimed is:

1. A method comprising:

receiving, by each of a plurality of packet security gateways associated with a security policy management server and from the security policy management server, a dynamic security policy that comprises at least one rule specifying application-layer packet-header information and a logging system address; and

routing, by the packet security gateway and on a packet-by-packet basis, to a monitoring device each of the one or more packets corresponding to the application-layer packet-header information in response to the performing the packet transformation function.

2. The method of claim **1**, wherein the at least one rule indicates performing an accept packet transformation function on the packets comprising the application-layer packet-header information, and wherein the performing the packet transformation function comprises forwarding, by the packet security gateway, the each of the one or more packets comprising the application-layer packet-header information toward its respective destination.

3. The method of claim **1**, wherein the at least one rule indicates performing a deny packet transformation function on the packets comprising the application-layer packet-header information, and wherein the performing the packet transformation function on comprises dropping, by the packet security gateway, the each of the one or more packets comprising the application-layer packet-header information.

4. The method of claim **1**, wherein the application-layer packet-header information identifies one or more hypertext transfer protocol (HTTP) packets, and wherein identifying the one or more packets comprising the application-layer packet-header information comprises determining by the packet security gateway that the one or more packets comprising the application-layer packet-header information are amongst the one or more HTTP packets.

5. The method of claim **4**, wherein the one or more HTTP packets comprise an HTTP GET method call, and wherein determining that the one or more packets comprising the application-layer packet-header information are amongst the one or more HTTP packets comprises determining that the one or more packets comprising the application-layer packet-header information are associated with the HTTP GET method call.

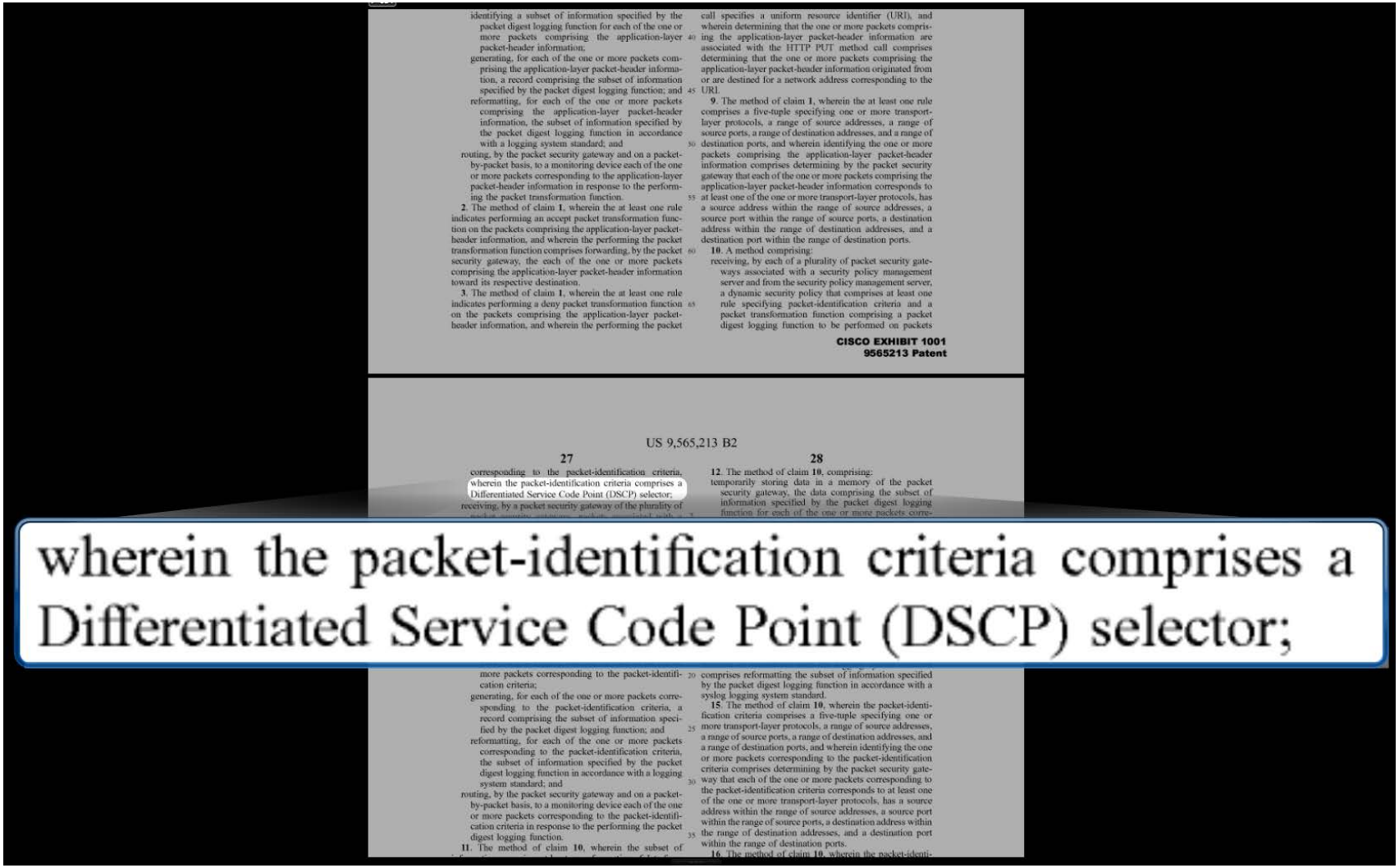
destination ports, and wherein identifying the one or more packets comprising the application-layer packet-header information comprises determining by the packet security gateway that each of the one or more packets comprising the application-layer packet-header information corresponds to at least one of the one or more transport-layer protocols, has a source address within the range of source addresses, a source port within the range of source ports, a destination address within the range of destination addresses, and a destination port within the range of destination ports.

10. A method comprising:

receiving, by each of a plurality of packet security gateways associated with a security policy management server and from the security policy management server, a dynamic security policy that comprises at least one rule specifying packet-identification criteria and a packet transformation function comprising a packet digest logging function to be performed on packets

CISCO EXHIBIT 1001
9565213 Patent

routing, by the packet security gateway and on a packet-by-packet basis, to a monitoring device each of the one or more packets corresponding to the application-layer packet-header information in response to the performing the packet transformation function.



identifying a subset of information specified by the packet digest logging function for each of the one or more packets comprising the application-layer packet-header information;

generating, for each of the one or more packets comprising the application-layer packet-header information, a record comprising the subset of information specified by the packet digest logging function; and

reformatting, for each of the one or more packets comprising the application-layer packet-header information, the subset of information specified by the packet digest logging function in accordance with a logging system standard; and

routing, by the packet security gateway and on a packet-by-packet basis, to a monitoring device each of the one or more packets corresponding to the application-layer packet-header information in response to the performing the packet transformation function.

2. The method of claim **1**, wherein the at least one rule indicates performing an accept packet transformation function on the packets comprising the application-layer packet-header information, and wherein the performing the packet transformation function comprises forwarding, by the packet security gateway, the each of the one or more packets comprising the application-layer packet-header information toward its respective destination.

3. The method of claim **1**, wherein the at least one rule indicates performing a deny packet transformation function on the packets comprising the application-layer packet-header information, and wherein the performing the packet call specifies a uniform resource identifier (URI), and wherein determining that the one or more packets comprising the application-layer packet-header information are associated with the HTTP PUT method call comprises determining that the one or more packets comprising the application-layer packet-header information originated from or are destined for a network address corresponding to the URI.

9. The method of claim **1**, wherein the at least one rule comprises a five-tuple specifying one or more transport-layer protocols, a range of source addresses, a range of source ports, a range of destination addresses, and a range of destination ports, and wherein identifying the one or more packets comprising the application-layer packet-header information comprises determining by the packet security gateway that each of the one or more packets comprising the application-layer packet-header information corresponds to at least one of the one or more transport-layer protocols, has a source address within the range of source addresses, a source port within the range of source ports, a destination address within the range of destination addresses, and a destination port within the range of destination ports.

10. A method comprising:

receiving, by each of a plurality of packet security gateways associated with a security policy management server and from the security policy management server, a dynamic security policy that comprises at least one rule specifying packet-identification criteria and a packet transformation function comprising a packet digest logging function to be performed on packets

CISCO EXHIBIT 1001
9565213 Patent

US 9,565,213 B2

27

corresponding to the packet-identification criteria, wherein the packet-identification criteria comprises a Differentiated Service Code Point (DSCP) selector;

receiving, by a packet security gateway of the plurality of

28

12. The method of claim **10**, comprising:

temporarily storing data in a memory of the packet security gateway, the data comprising the subset of information specified by the packet digest logging function for each of the one or more packets corre-

wherein the packet-identification criteria comprises a Differentiated Service Code Point (DSCP) selector;

more packets corresponding to the packet-identification criteria;

generating, for each of the one or more packets corresponding to the packet-identification criteria, a record comprising the subset of information specified by the packet digest logging function; and

reformatting, for each of the one or more packets corresponding to the packet-identification criteria, the subset of information specified by the packet digest logging function in accordance with a logging system standard; and

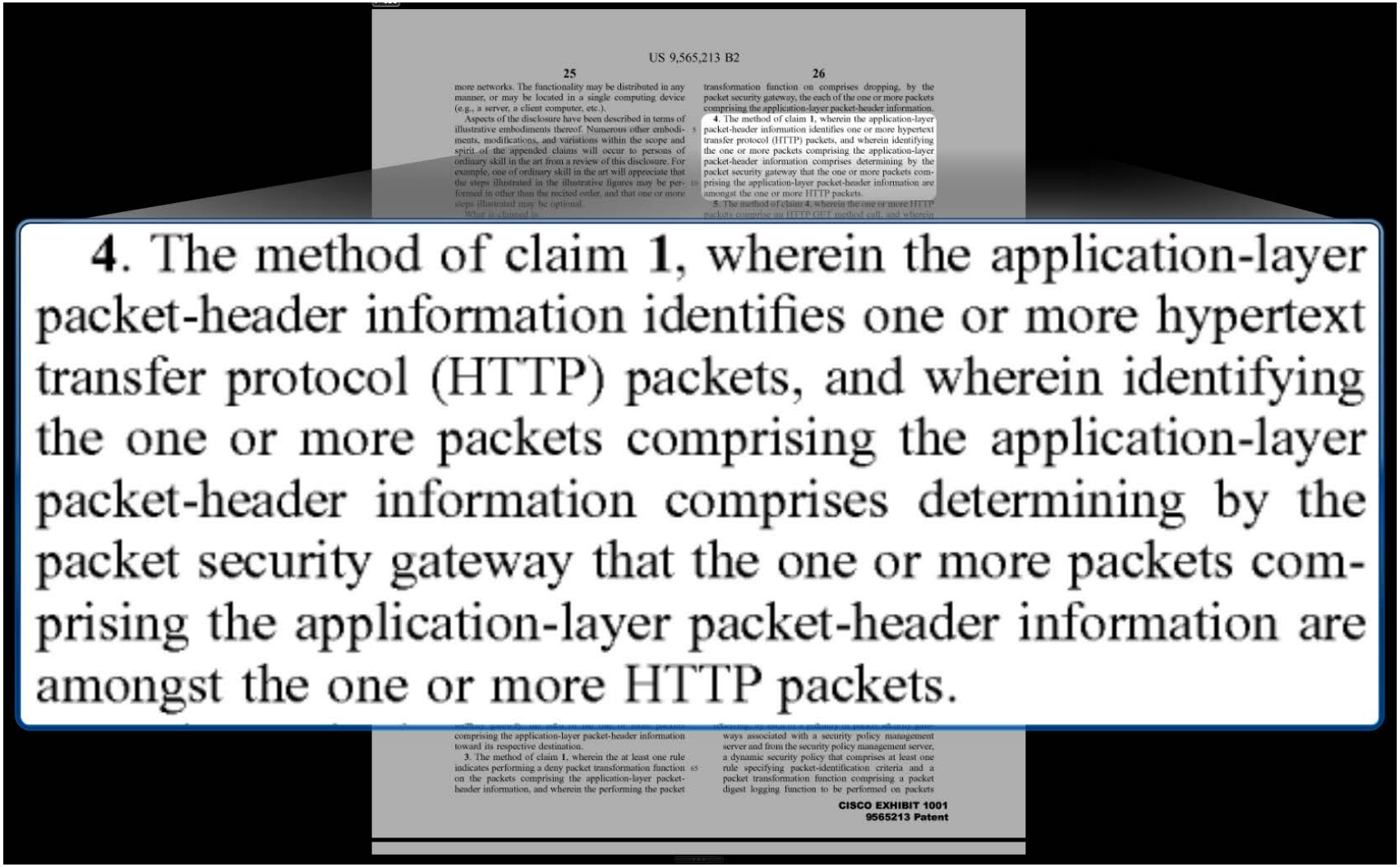
routing, by the packet security gateway and on a packet-by-packet basis, to a monitoring device each of the one or more packets corresponding to the packet-identification criteria in response to the performing the packet digest logging function.

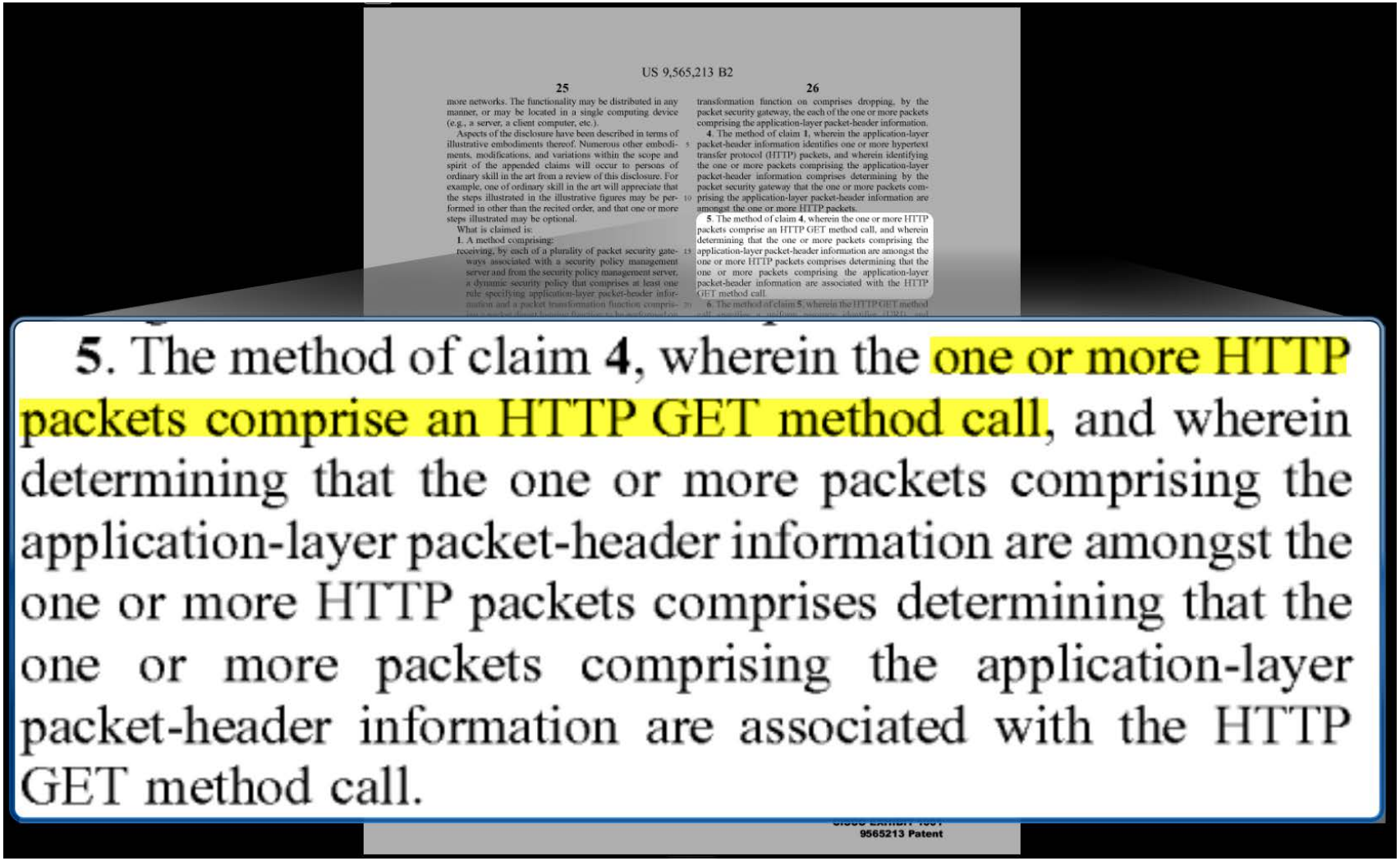
11. The method of claim **10**, wherein the subset of

comprises reformatting the subset of information specified by the packet digest logging function in accordance with a syslog logging system standard.

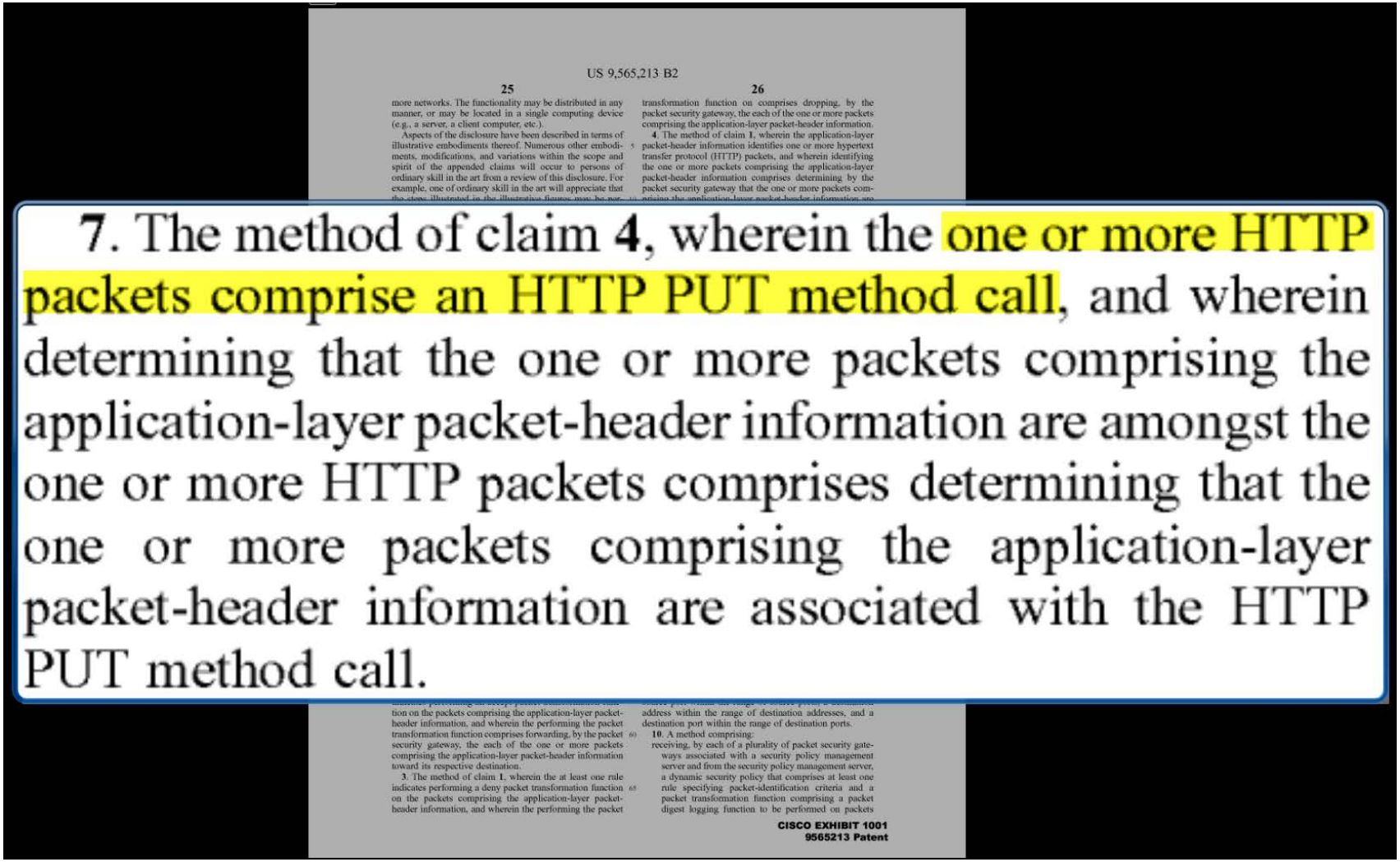
15. The method of claim **10**, wherein the packet-identification criteria comprises a five-tuple specifying one or more transport-layer protocols, a range of source addresses, a range of source ports, a range of destination addresses, and a range of destination ports, and wherein identifying the one or more packets corresponding to the packet-identification criteria comprises determining by the packet security gateway that each of the one or more packets corresponding to the packet-identification criteria corresponds to at least one of the one or more transport-layer protocols, has a source address within the range of source addresses, a source port within the range of source ports, a destination address within the range of destination addresses, and a destination port within the range of destination ports.

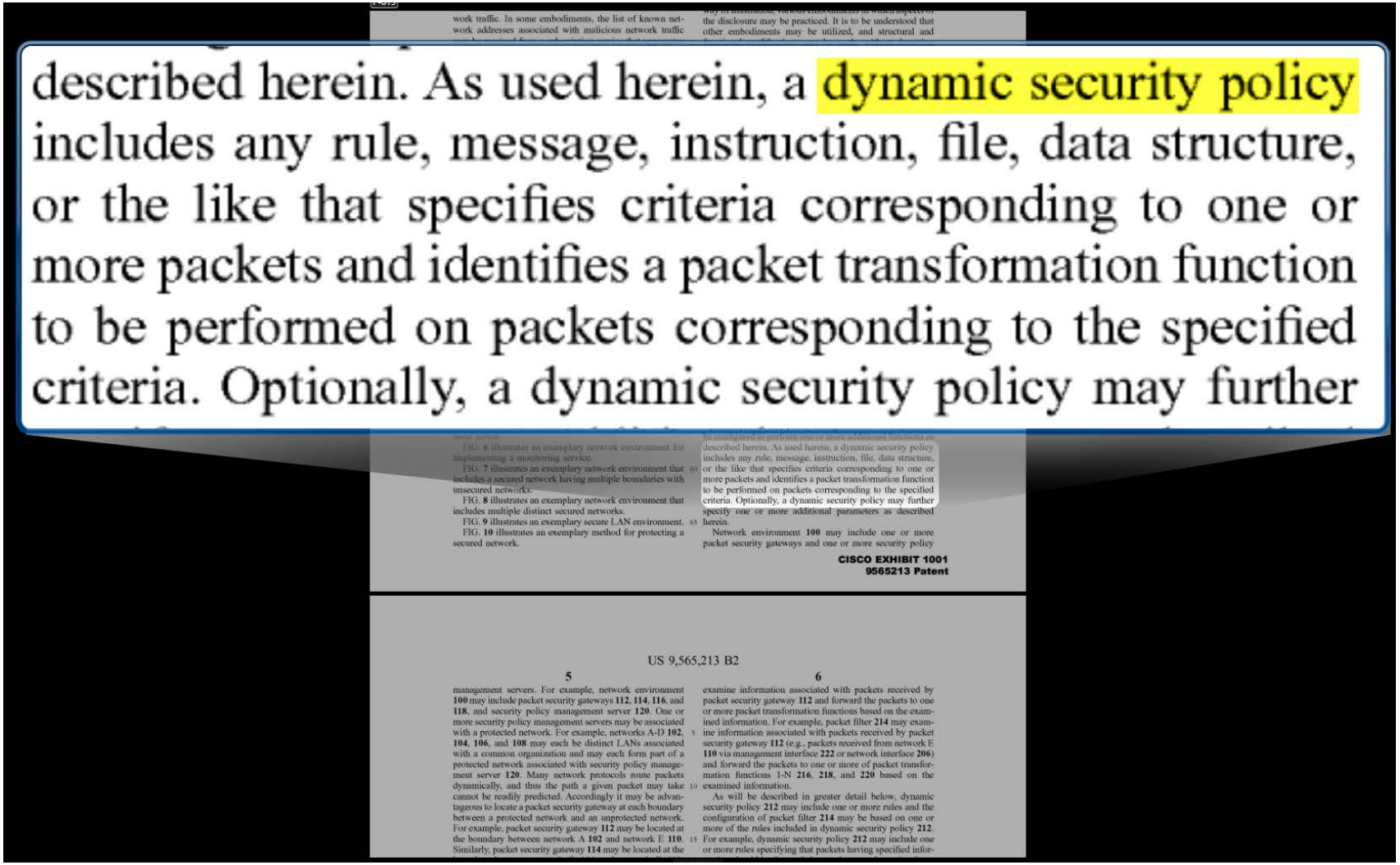
16. The method of claim **10**, wherein the packet-identi-

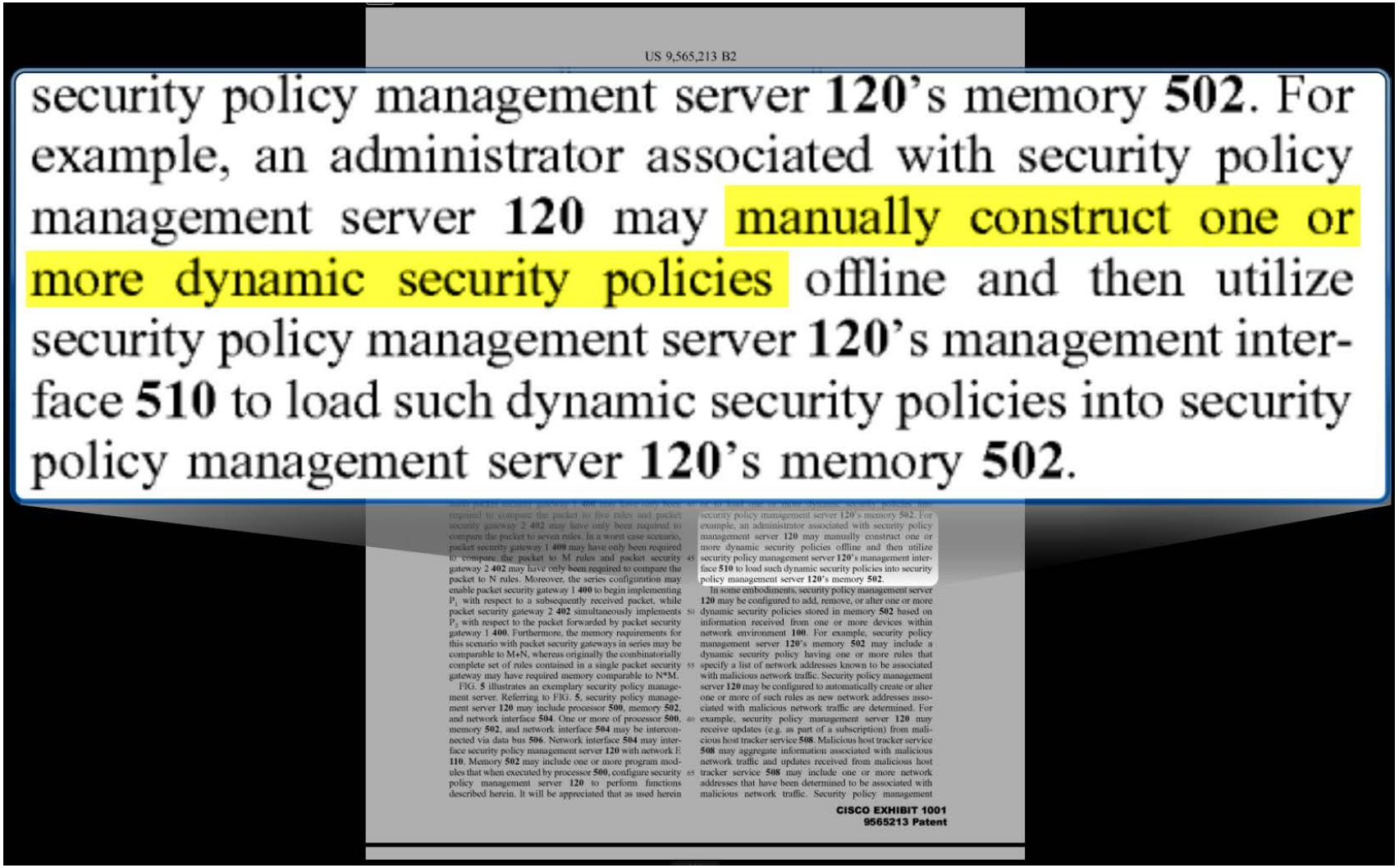


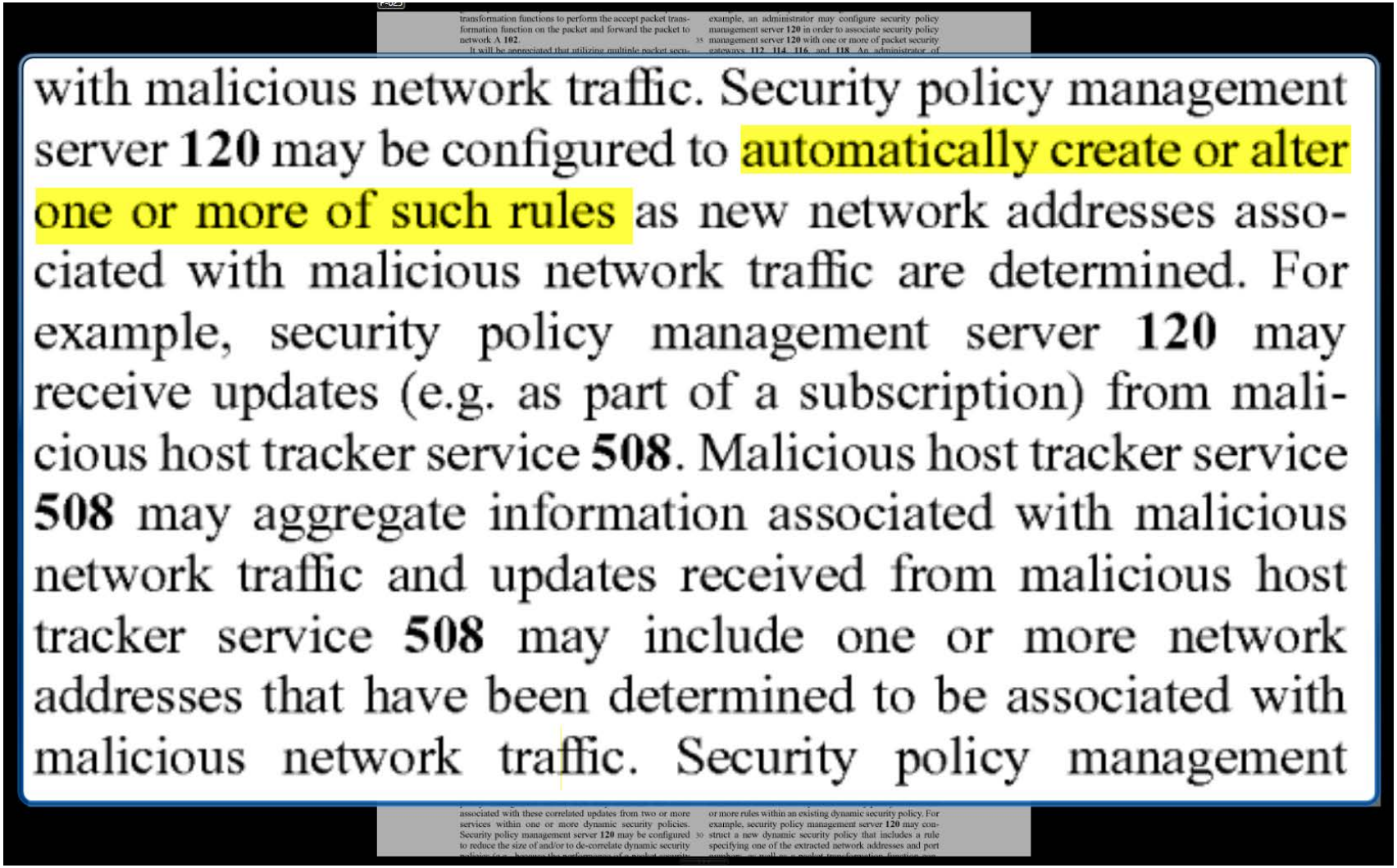


5. The method of claim **4**, wherein the one or more HTTP packets comprise an HTTP GET method call, and wherein determining that the one or more packets comprising the application-layer packet-header information are amongst the one or more HTTP packets comprises determining that the one or more packets comprising the application-layer packet-header information are associated with the HTTP GET method call.

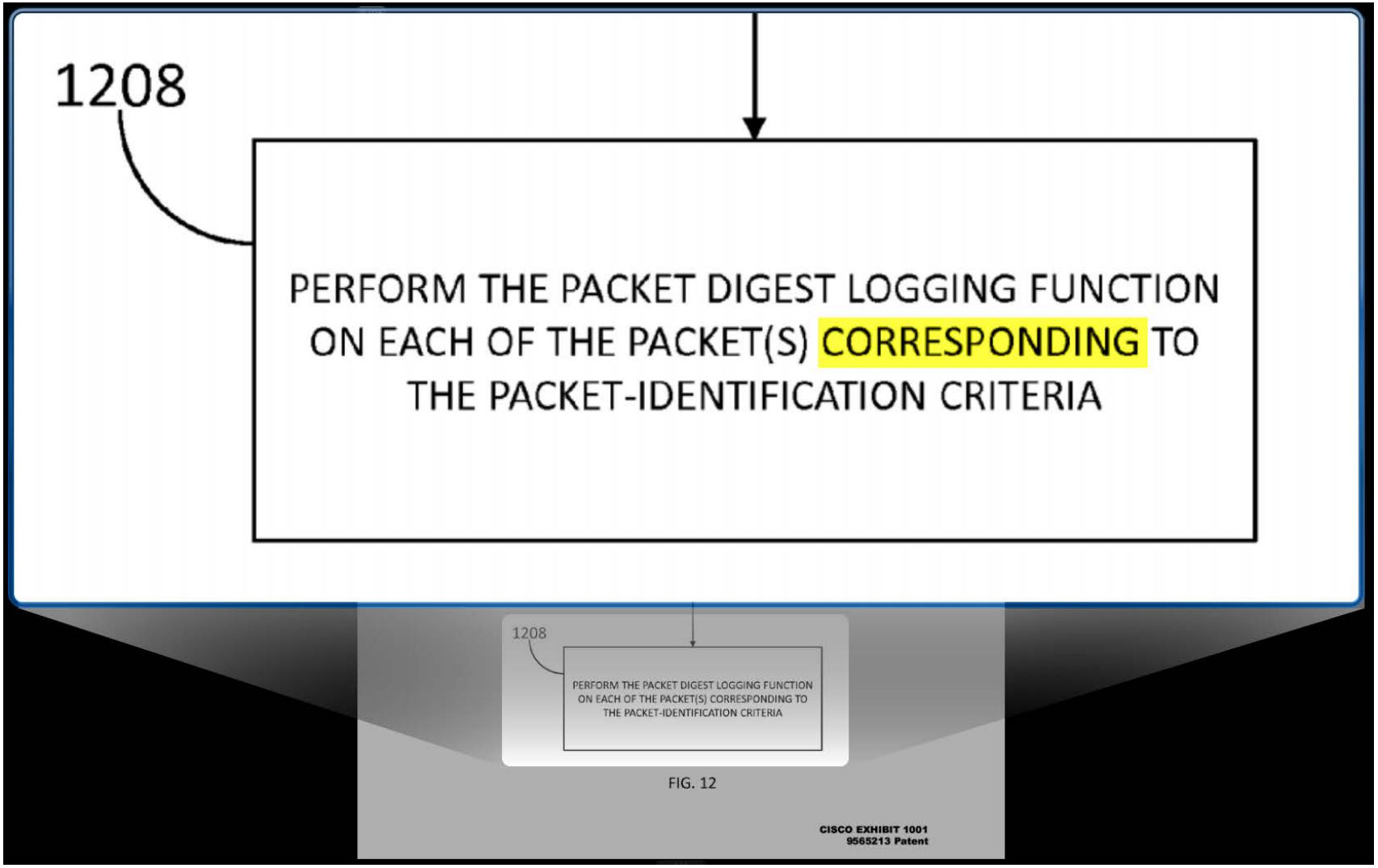


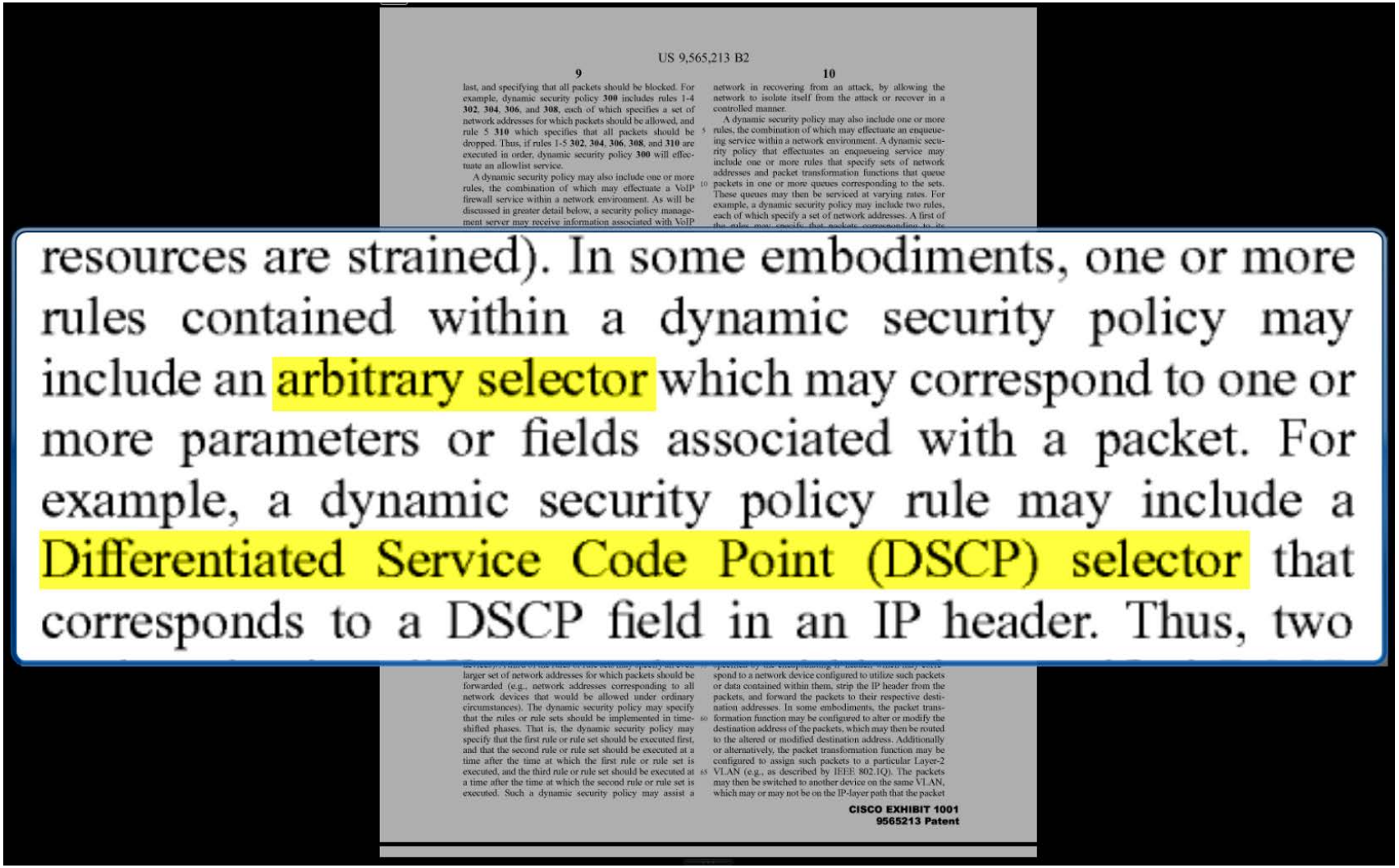






with malicious network traffic. Security policy management server **120** may be configured to automatically create or alter one or more of such rules as new network addresses associated with malicious network traffic are determined. For example, security policy management server **120** may receive updates (e.g. as part of a subscription) from malicious host tracker service **508**. Malicious host tracker service **508** may aggregate information associated with malicious network traffic and updates received from malicious host tracker service **508** may include one or more network addresses that have been determined to be associated with malicious network traffic. Security policy management





US 9,565,213 B2

9

last, and specifying that all packets should be blocked. For example, dynamic security policy **300** includes rules 1-4 **302**, **304**, **306**, and **308**, each of which specifies a set of network addresses for which packets should be allowed, and rule 5 **310** which specifies that all packets should be dropped. Thus, if rules 1-5 **302**, **304**, **306**, **308**, and **310** are executed in order, dynamic security policy **300** will effectuate an allowlist service.

A dynamic security policy may also include one or more rules, the combination of which may effectuate a VoIP firewall service within a network environment. As will be discussed in greater detail below, a security policy management server may receive information associated with VoIP

10

network in recovering from an attack, by allowing the network to isolate itself from the attack or recover in a controlled manner.

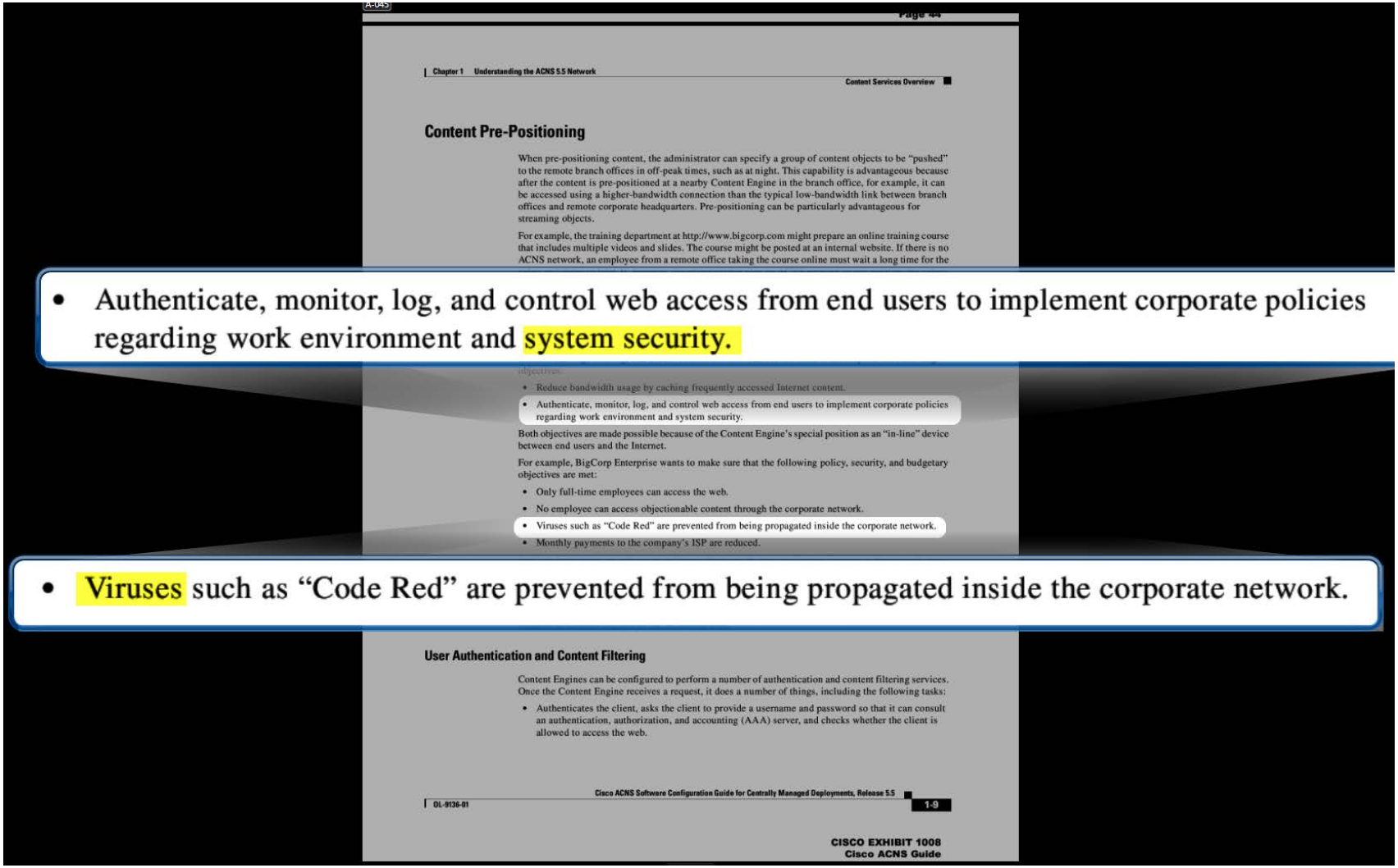
A dynamic security policy may also include one or more rules, the combination of which may effectuate an enqueueing service within a network environment. A dynamic security policy that effectuates an enqueueing service may include one or more rules that specify sets of network addresses and packet transformation functions that queue packets in one or more queues corresponding to the sets. These queues may then be serviced at varying rates. For example, a dynamic security policy may include two rules, each of which specify a set of network addresses. A first of

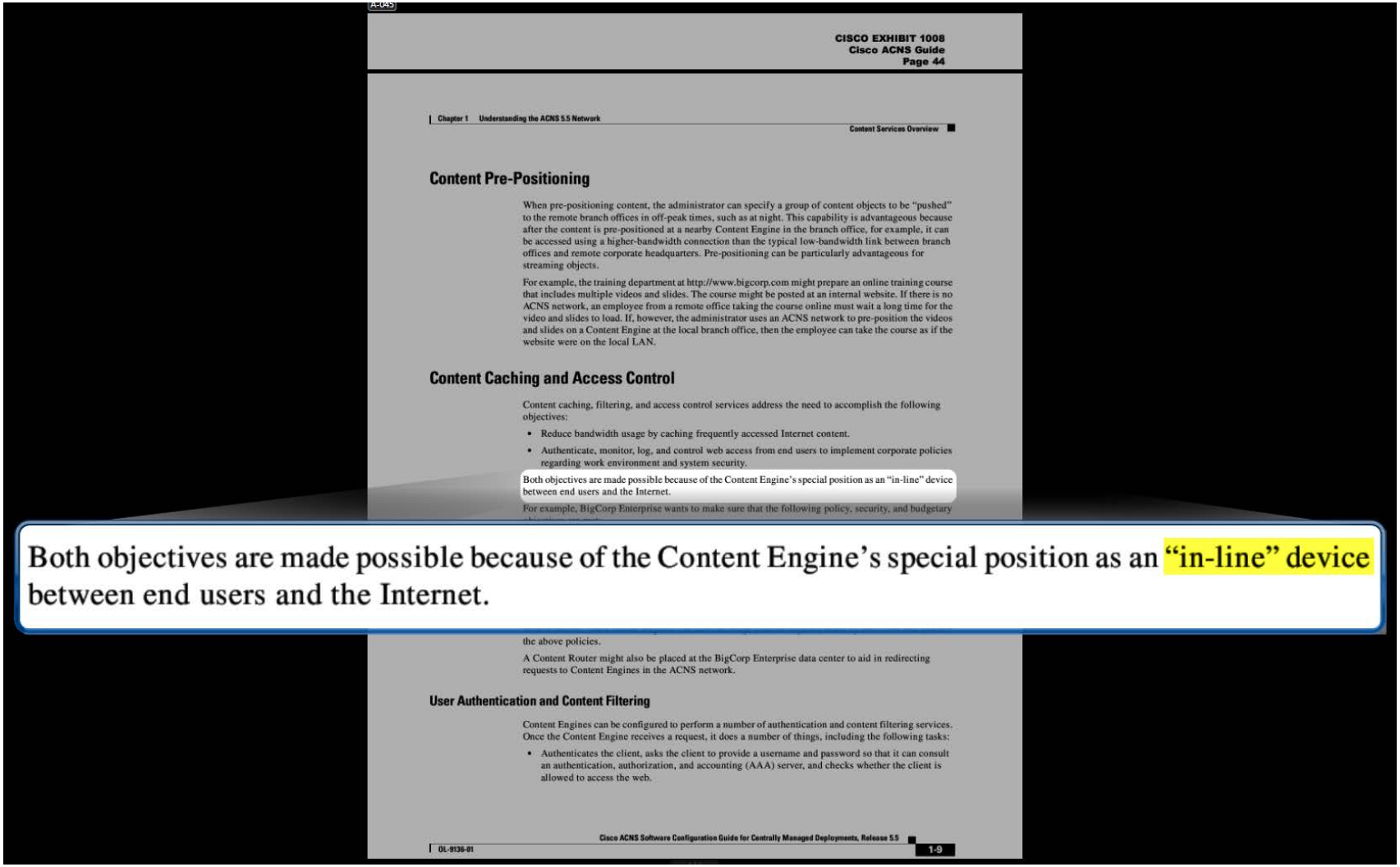
resources are strained). In some embodiments, one or more rules contained within a dynamic security policy may include an arbitrary selector which may correspond to one or more parameters or fields associated with a packet. For example, a dynamic security policy rule may include a Differentiated Service Code Point (DSCP) selector that corresponds to a DSCP field in an IP header. Thus, two

larger set of network addresses for which packets should be forwarded (e.g., network addresses corresponding to all network devices that would be allowed under ordinary circumstances). The dynamic security policy may specify that the rules or rule sets should be implemented in time-shifted phases. That is, the dynamic security policy may specify that the first rule or rule set should be executed first, and that the second rule or rule set should be executed at a time after the time at which the first rule or rule set is executed, and the third rule or rule set should be executed at a time after the time at which the second rule or rule set is executed. Such a dynamic security policy may assist a

spond to a network device configured to utilize such packets or data contained within them, strip the IP header from the packets, and forward the packets to their respective destination addresses. In some embodiments, the packet transformation function may be configured to alter or modify the destination address of the packets, which may then be routed to the altered or modified destination address. Additionally or alternatively, the packet transformation function may be configured to assign such packets to a particular Layer-2 VLAN (e.g., as described by IEEE 802.1Q). The packets may then be switched to another device on the same VLAN, which may or may not be on the IP-layer path that the packet

CISCO EXHIBIT 1001
9565213 Patent





CISCO EXHIBIT 1008
Cisco ACNS Guide
Page 44

Chapter 1 Understanding the ACNS 5.5 Network
Content Services Overview

Content Pre-Positioning

When pre-positioning content, the administrator can specify a group of content objects to be "pushed" to the remote branch offices in off-peak times, such as at night. This capability is advantageous because after the content is pre-positioned at a nearby Content Engine in the branch office, for example, it can be accessed using a higher-bandwidth connection than the typical low-bandwidth link between branch offices and remote corporate headquarters. Pre-positioning can be particularly advantageous for streaming objects.

For example, the training department at http://www.bigcorp.com might prepare an online training course that includes multiple videos and slides. The course might be posted at an internal website. If there is no ACNS network, an employee from a remote office taking the course online must wait a long time for the video and slides to load. If, however, the administrator uses an ACNS network to pre-position the videos and slides on a Content Engine at the local branch office, then the employee can take the course as if the website were on the local LAN.

Content Caching and Access Control

Content caching, filtering, and access control services address the need to accomplish the following objectives:

- Reduce bandwidth usage by caching frequently accessed Internet content.
- Authenticate, monitor, log, and control web access from end users to implement corporate policies regarding work environment and system security.

Both objectives are made possible because of the Content Engine's special position as an "in-line" device between end users and the Internet.

For example, BigCorp Enterprise wants to make sure that the following policy, security, and budgetary

Both objectives are made possible because of the Content Engine's special position as an "in-line" device between end users and the Internet.

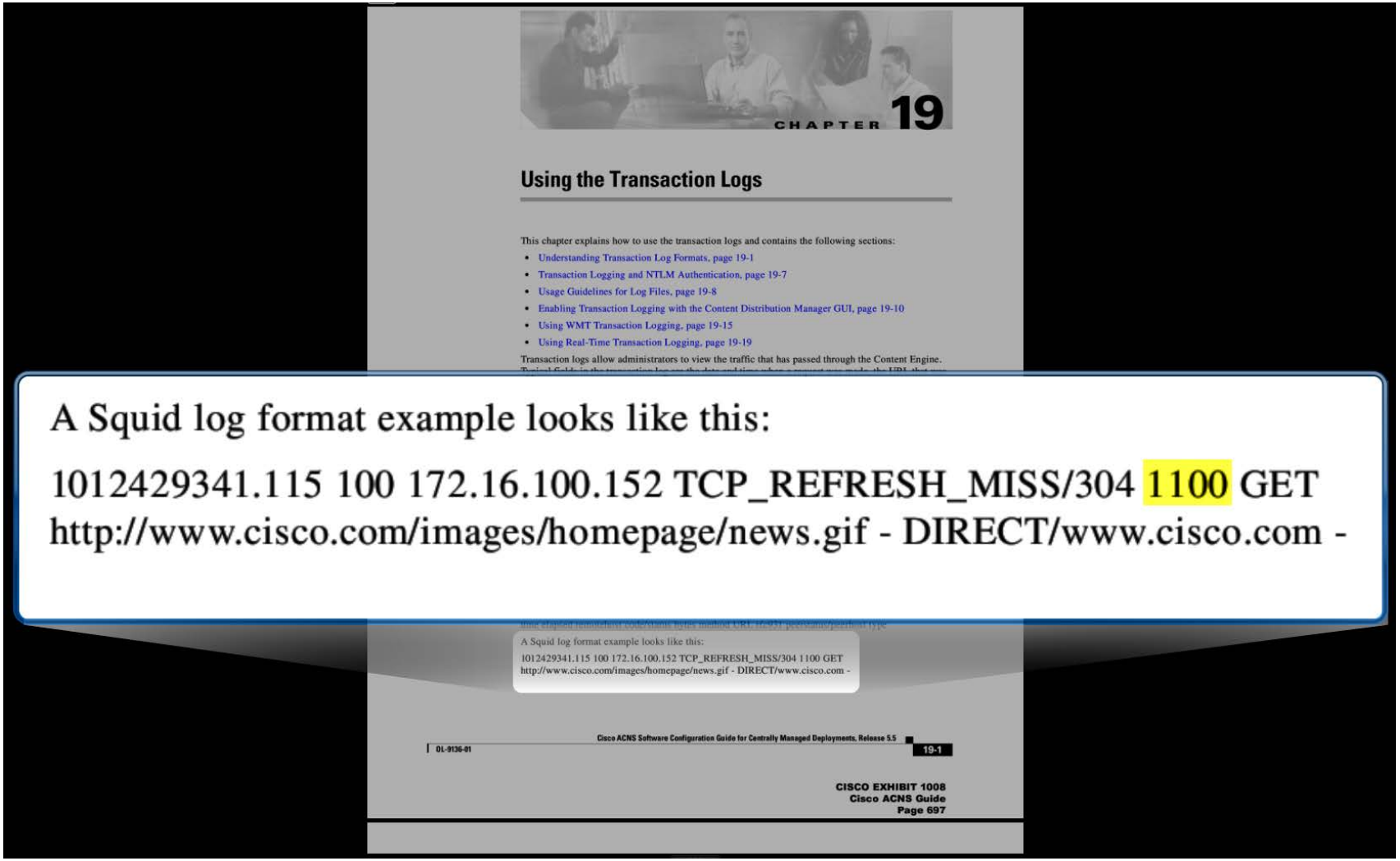
the above policies.

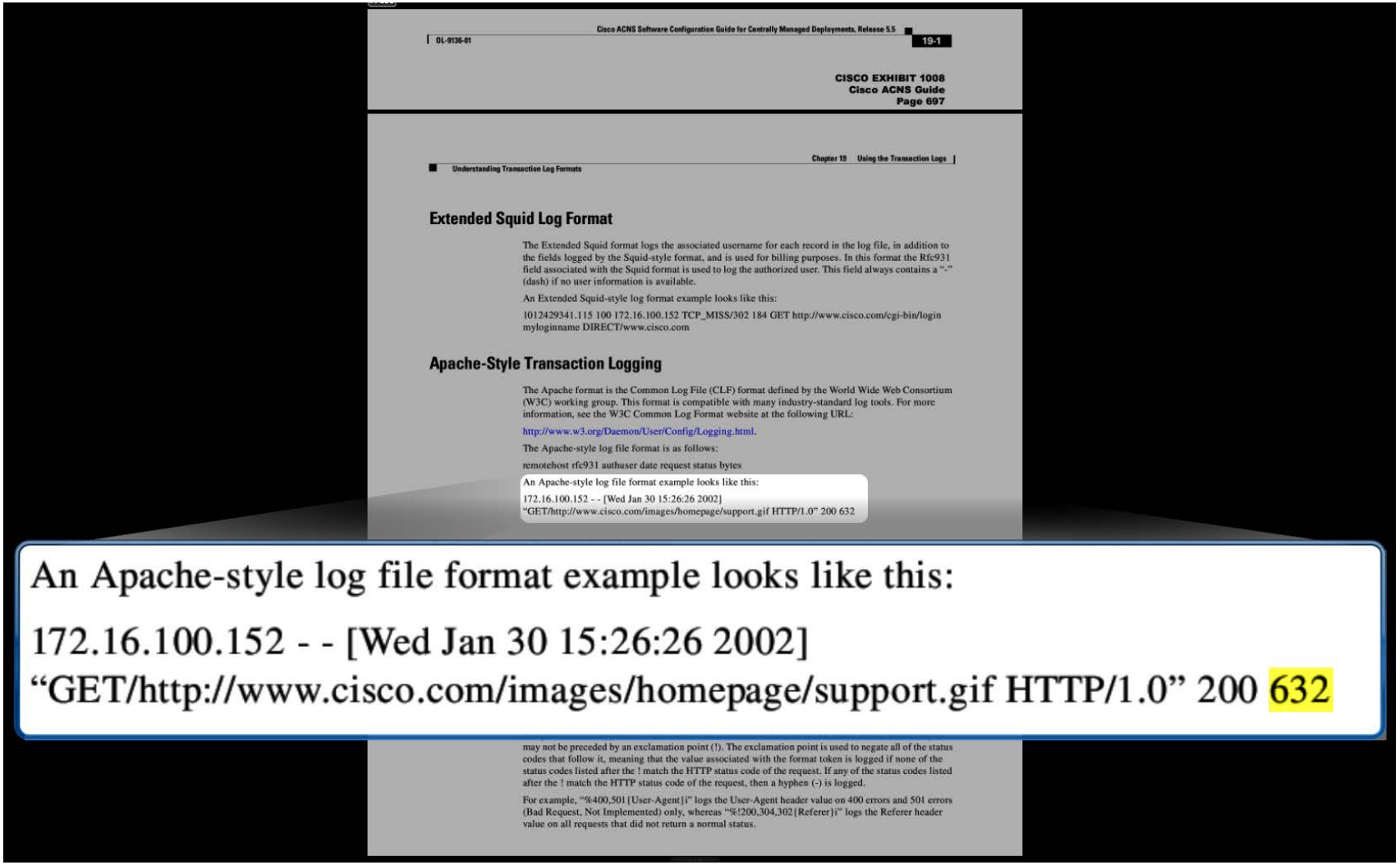
A Content Router might also be placed at the BigCorp Enterprise data center to aid in redirecting requests to Content Engines in the ACNS network.

User Authentication and Content Filtering

Content Engines can be configured to perform a number of authentication and content filtering services. Once the Content Engine receives a request, it does a number of things, including the following tasks:

- Authenticates the client, asks the client to provide a username and password so that it can consult an authentication, authorization, and accounting (AAA) server, and checks whether the client is allowed to access the web.





Cisco ACNS Software Configuration Guide for Centrally Managed Deployments, Release 5.5
OL-9136-01
19-1

CISCO EXHIBIT 1008
Cisco ACNS Guide
Page 697

Chapter 19 Using the Transaction Logs
Understanding Transaction Log Formats

Extended Squid Log Format

The Extended Squid format logs the associated username for each record in the log file, in addition to the fields logged by the Squid-style format, and is used for billing purposes. In this format the Rfc931 field associated with the Squid format is used to log the authorized user. This field always contains a "-" (dash) if no user information is available.

An Extended Squid-style log format example looks like this:

1012429341.115 100 172.16.100.152 TCP_MISS/302 184 GET http://www.cisco.com/cgi-bin/login myloginname DIRECT/www.cisco.com

Apache-Style Transaction Logging

The Apache format is the Common Log File (CLF) format defined by the World Wide Web Consortium (W3C) working group. This format is compatible with many industry-standard log tools. For more information, see the W3C Common Log Format website at the following URL:

http://www.w3.org/Daemon/User/Config/Logging.html.

The Apache-style log file format is as follows:

remotehost rfc931 authuser date request status bytes

An Apache-style log file format example looks like this:
172.16.100.152 - - [Wed Jan 30 15:26:26 2002]
"GET/http://www.cisco.com/images/homepage/support.gif HTTP/1.0" 200 632

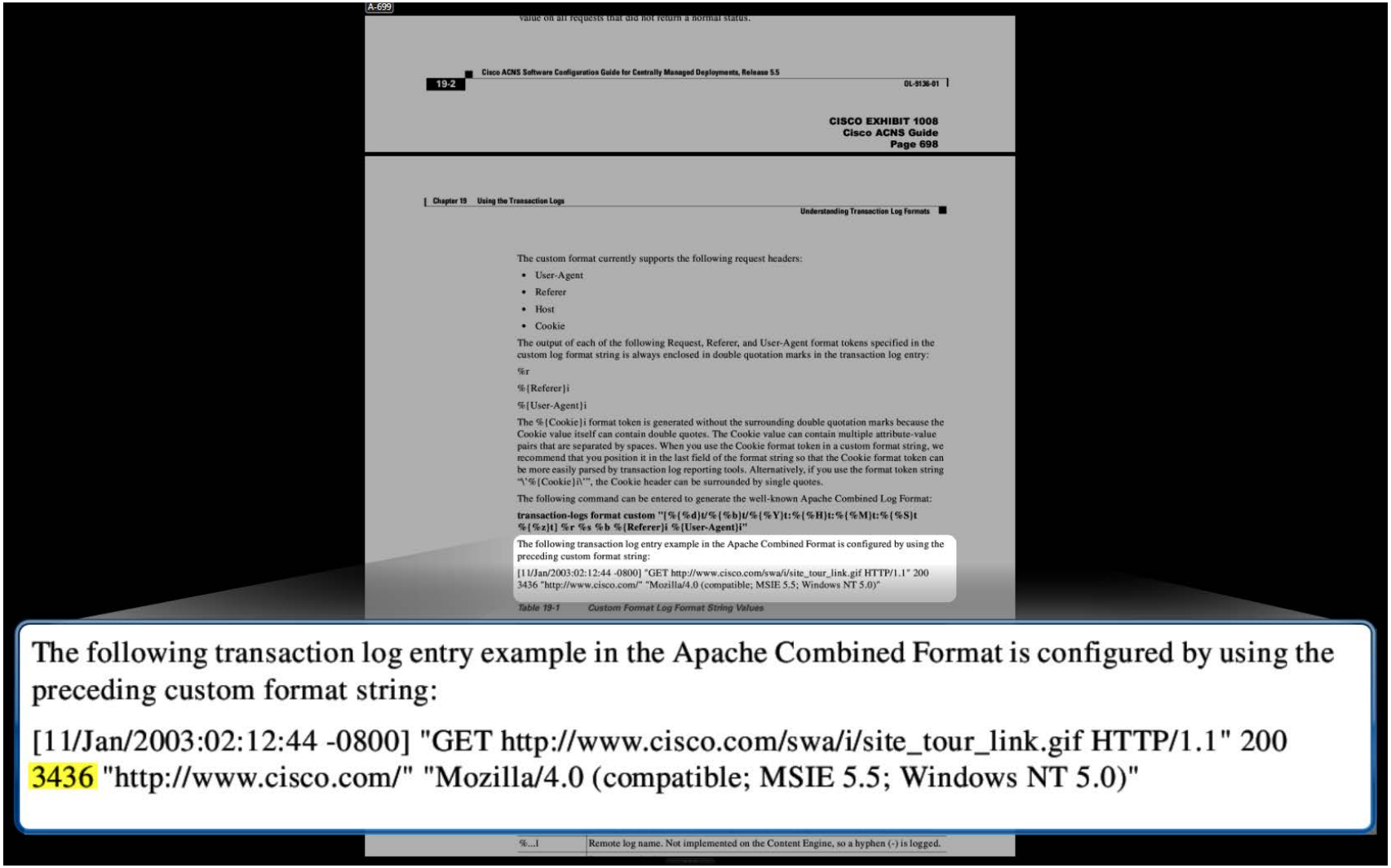
An Apache-style log file format example looks like this:

172.16.100.152 - - [Wed Jan 30 15:26:26 2002]
"GET/http://www.cisco.com/images/homepage/support.gif HTTP/1.0" 200 632

may not be preceded by an exclamation point (!). The exclamation point is used to negate all of the status codes that follow it, meaning that the value associated with the format token is logged if none of the status codes listed after the ! match the HTTP status code of the request. If any of the status codes listed after the ! match the HTTP status code of the request, then a hyphen (-) is logged.

For example, "%400,501{User-Agent}i" logs the User-Agent header value on 400 errors and 501 errors (Bad Request, Not Implemented) only, whereas "%!200,304,302{Referer}i" logs the Referer header value on all requests that did not return a normal status.

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE



The following transaction log entry example in the Apache Combined Format is configured by using the preceding custom format string:

[11/Jan/2003:02:12:44 -0800] "GET http://www.cisco.com/swa/i/site_tour_link.gif HTTP/1.1" 200 3436 "http://www.cisco.com/" "Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)"

Adding or Modifying Additional HTTP Request Methods for the Content Engine

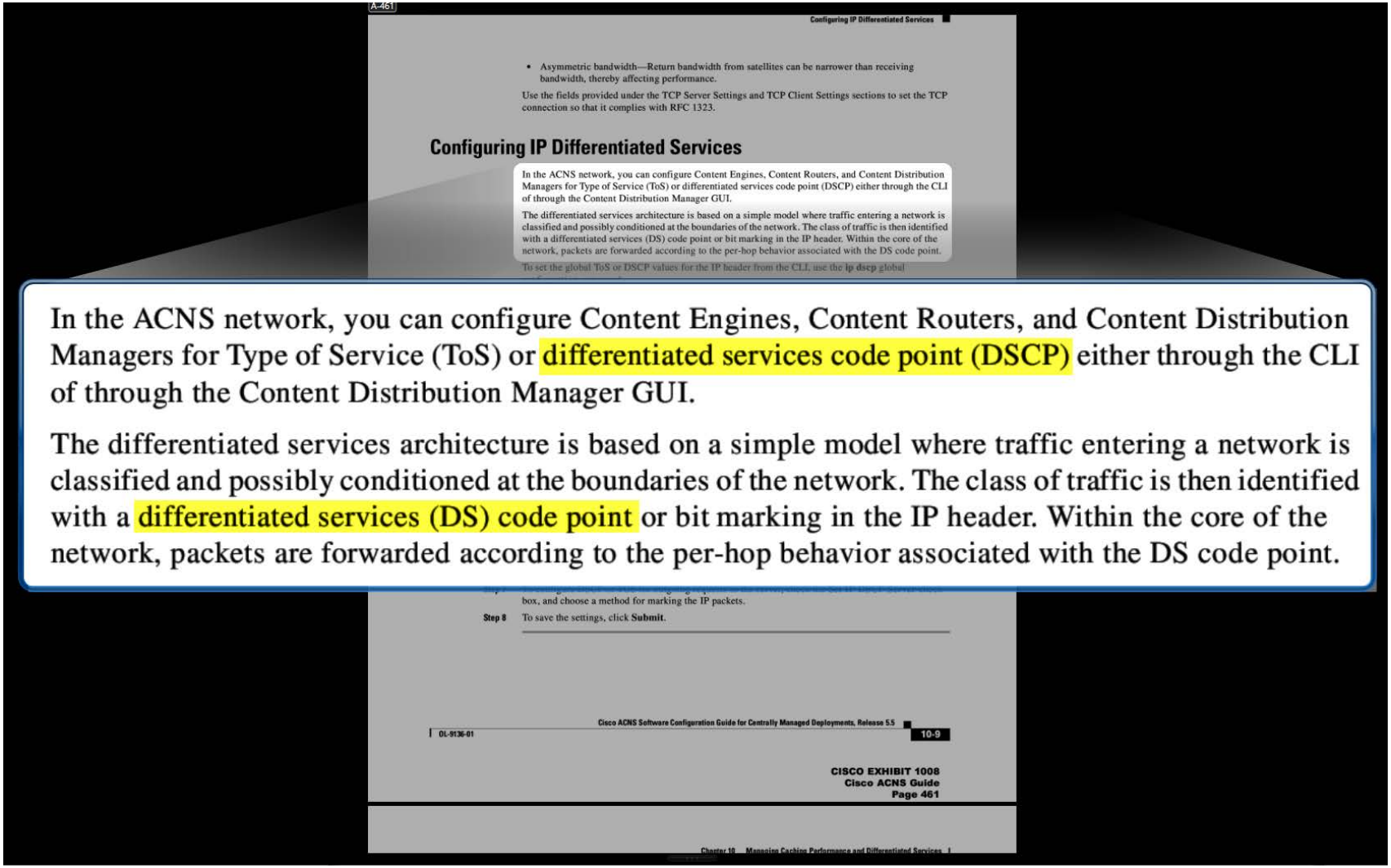
Content Engines accept or reject an HTTP request depending on whether the HTTP request method is supported. HTTP 1.1 request methods (for example, GET, HEAD, or POST) are supported by default. Nonstandard request methods, such as Web-Based Distributed Authoring and Versioning (WebDAV) are not.

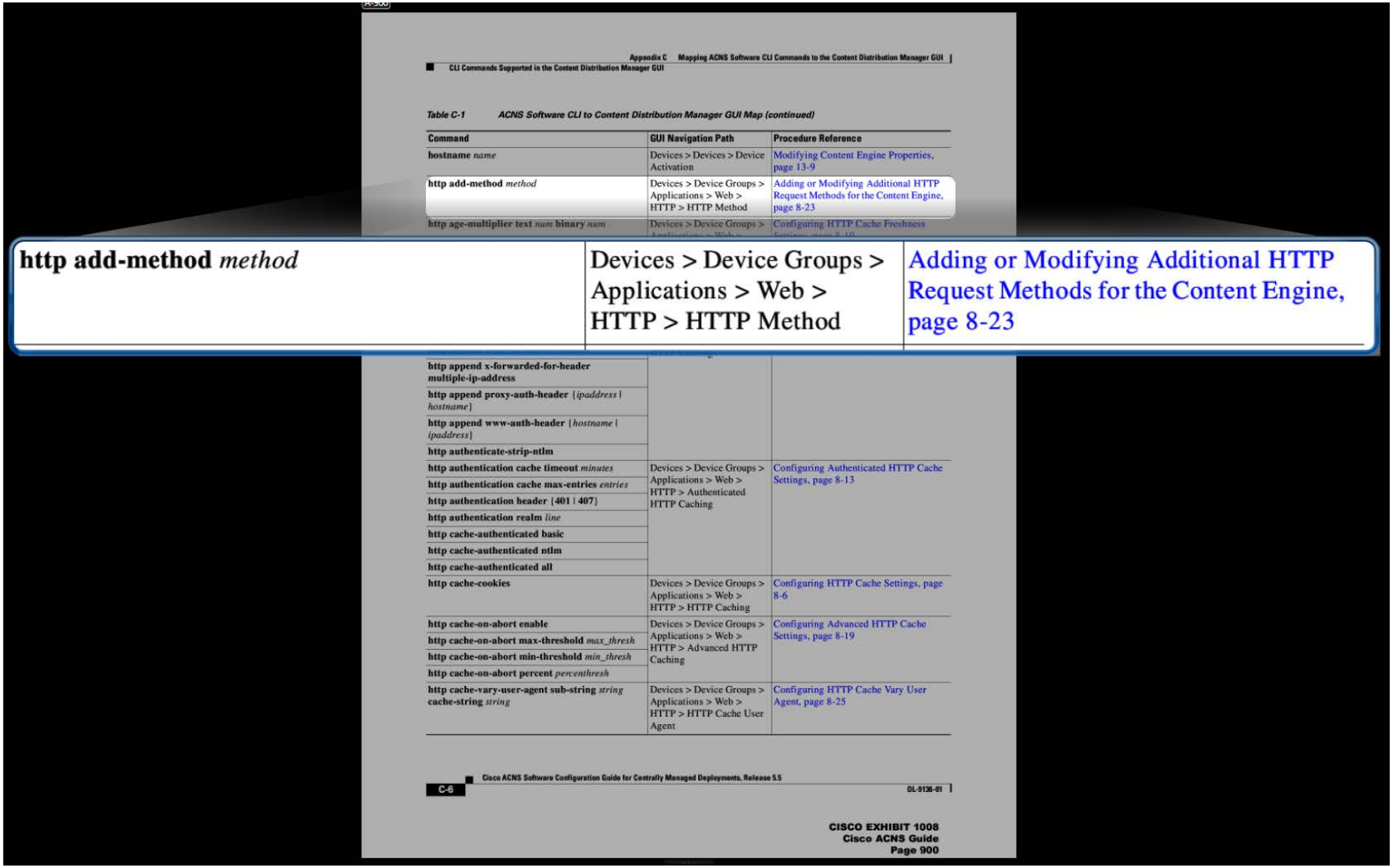
When the Content Engine receives an HTTP request from a client using an unsupported method, ACNS software adds the method to the list of unsupported methods and sends an error to the client. You can use the Creating New HTTP Method for Content Engine window to add a method to the list of methods already supported by the Content Engine.

Adding or Modifying Additional HTTP Request Methods for the Content Engine

Content Engines accept or reject an HTTP request depending on whether the HTTP request method is supported. HTTP 1.1 request methods (for example, GET, HEAD, or POST) are supported by default. Nonstandard request methods, such as Web-Based Distributed Authoring and Versioning (WebDAV) are not.

When the Content Engine receives an HTTP request from a client using an unsupported method, ACNS software adds the method to the list of unsupported methods and sends an error to the client. You can use the Creating New HTTP Method for Content Engine window to add a method to the list of methods already supported by the Content Engine.





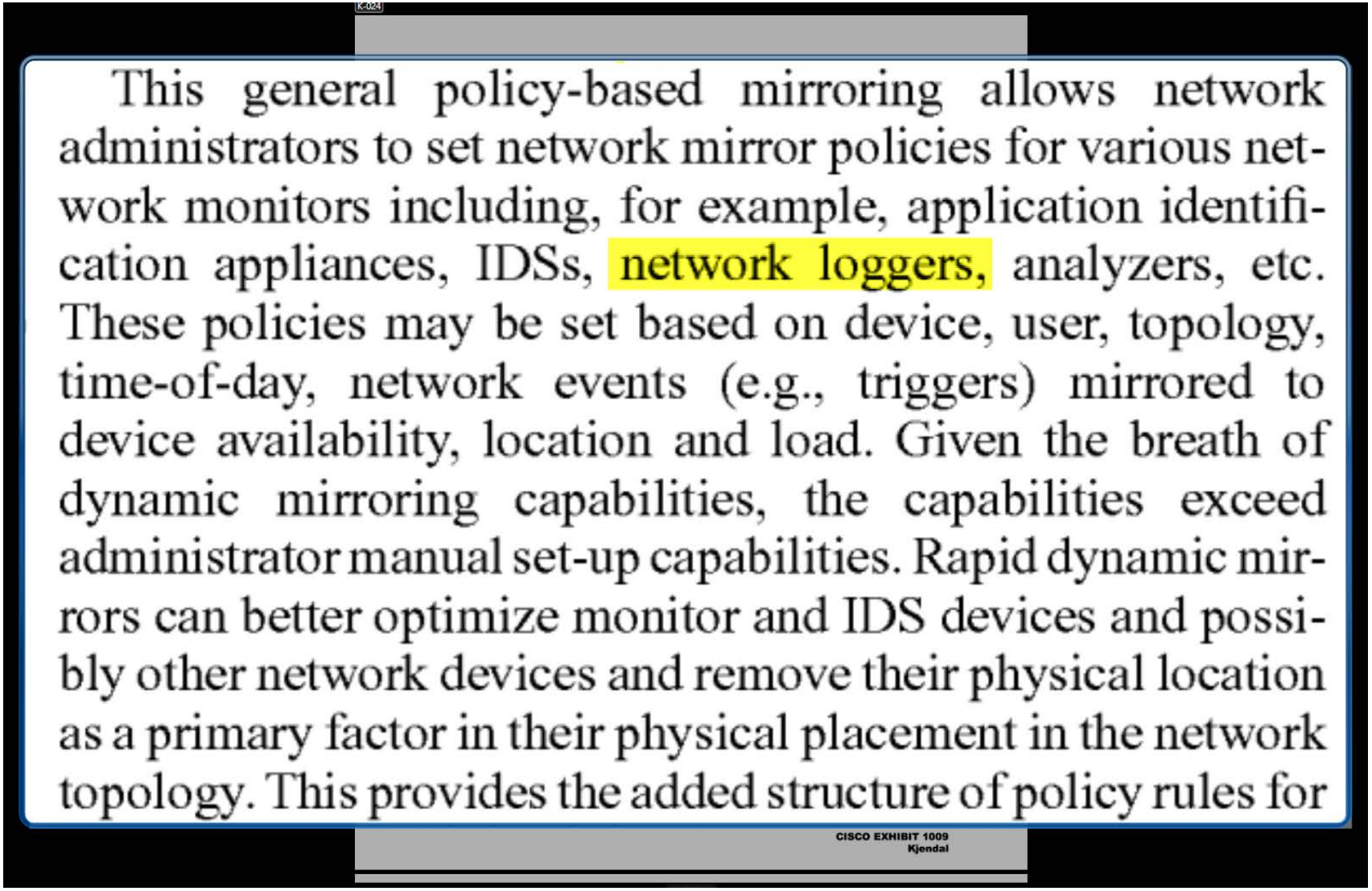
Appendix C Mapping ACNS Software CLI Commands to the Content Distribution Manager GUI
CLI Commands Supported in the Content Distribution Manager GUI

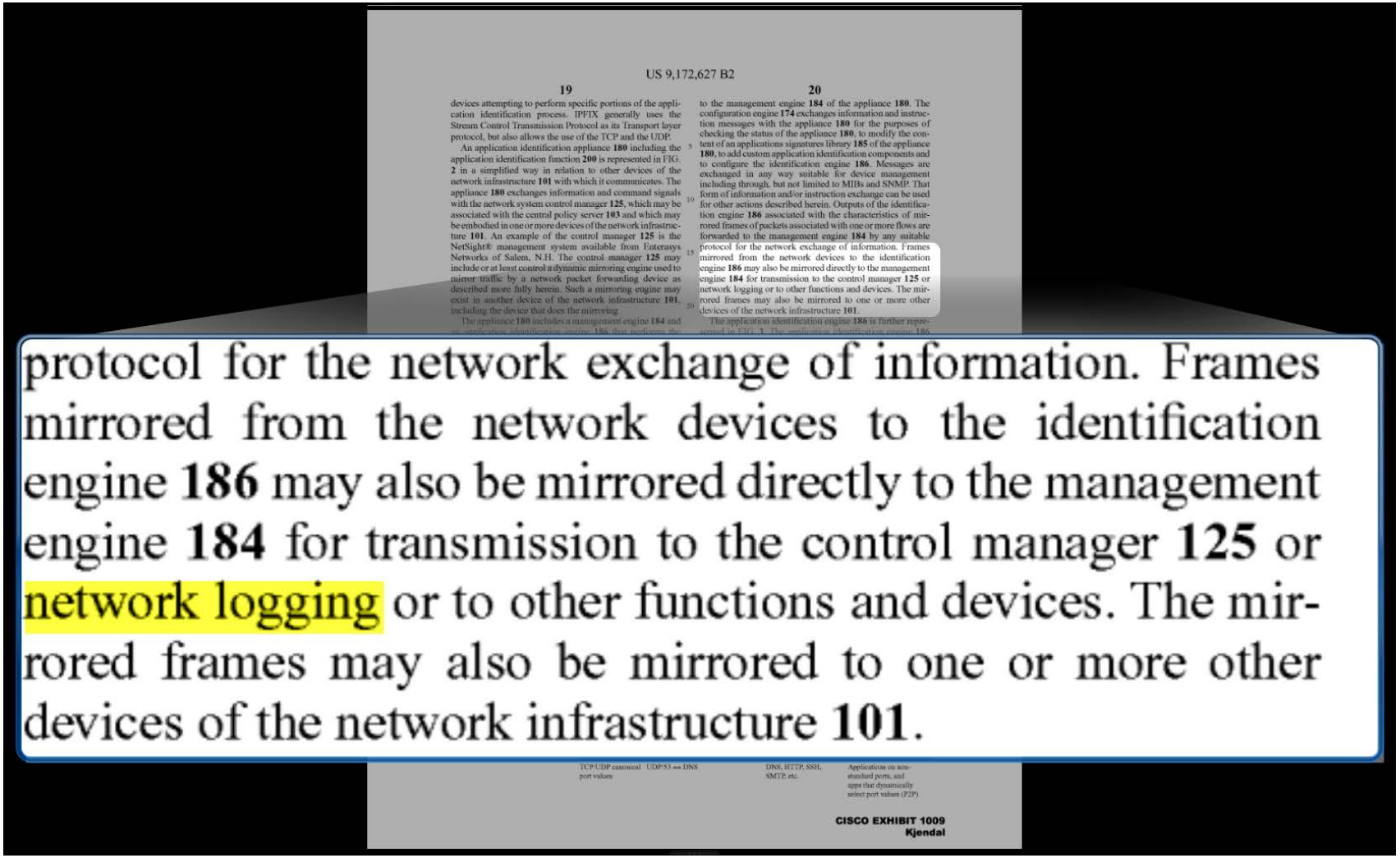
Table C-1 ACNS Software CLI to Content Distribution Manager GUI Map (continued)

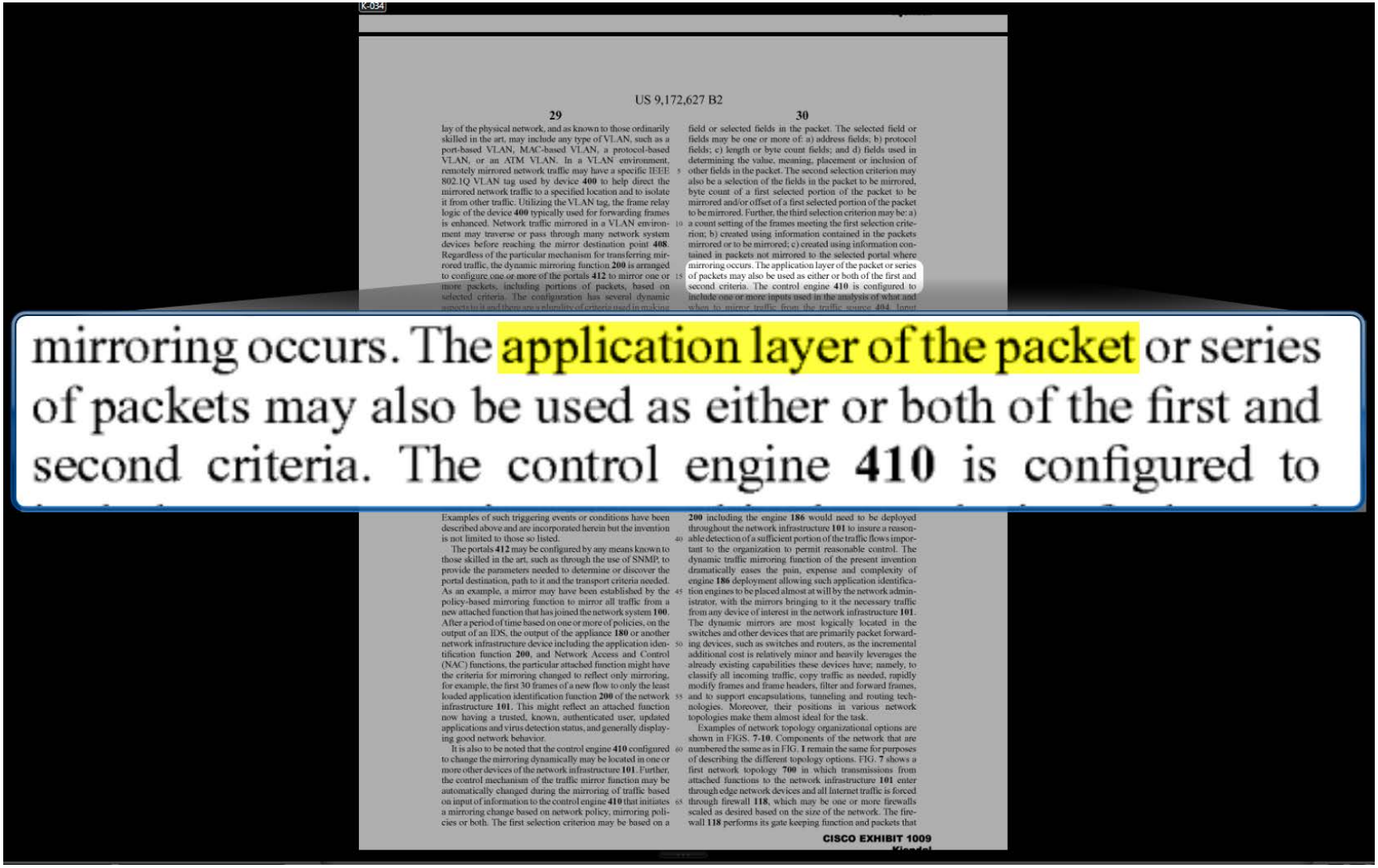
Command	GUI Navigation Path	Procedure Reference
hostname *name*	Devices > Devices > Device Activation	Modifying Content Engine Properties, page 13-9
http add-method *method*	Devices > Device Groups > Applications > Web > HTTP > HTTP Method	Adding or Modifying Additional HTTP Request Methods for the Content Engine, page 8-23
http age-multiplier text *num* **binary** *num*	Devices > Device Groups >	Configuring HTTP Cache Freshness
http append x-forwarded-for-header multiple-ip-address		
http append proxy-auth-header {*ipaddress* | *hostname*}		
http append www-auth-header {*hostname* | *ipaddress*}		
http authenticate-strip-ntlm		
http authentication cache timeout *minutes*	Devices > Device Groups > Applications > Web > HTTP > Authenticated HTTP Caching	Configuring Authenticated HTTP Cache Settings, page 8-13
http authentication cache max-entries *entries*		
http authentication header {**401** | **407**}		
http authentication realm *line*		
http cache-authenticated basic		
http cache-authenticated ntlm		
http cache-authenticated all		
http cache-cookies	Devices > Device Groups > Applications > Web > HTTP > HTTP Caching	Configuring HTTP Cache Settings, page 8-6
http cache-on-abort enable	Devices > Device Groups > Applications > Web > HTTP > Advanced HTTP Caching	Configuring Advanced HTTP Cache Settings, page 8-19
http cache-on-abort max-threshold *max_thresh*		
http cache-on-abort min-threshold *min_thresh*		
http cache-on-abort percent *percenthresh*		
http cache-vary-user-agent sub-string *string* **cache-string** *string*	Devices > Device Groups > Applications > Web > HTTP > HTTP Cache User Agent	Configuring HTTP Cache Vary User Agent, page 8-25

Cisco ACNS Software Configuration Guide for Centrally Managed Deployments, Release 5.5
C-6
OL-9136-01

CISCO EXHIBIT 1008
Cisco ACNS Guide
Page 900







US 9,172,627 B2

29

lay of the physical network, and as known to those ordinarily skilled in the art, may include any type of VLAN, such as a port-based VLAN, MAC-based VLAN, a protocol-based VLAN, or an ATM VLAN. In a VLAN environment, remotely mirrored network traffic may have a specific IEEE 802.1Q VLAN tag used by device **400** to help direct the mirrored network traffic to a specified location and to isolate it from other traffic. Utilizing the VLAN tag, the frame relay logic of the device **400** typically used for forwarding frames is enhanced. Network traffic mirrored in a VLAN environment may traverse or pass through many network system devices before reaching the mirror destination point **408**. Regardless of the particular mechanism for transferring mirrored traffic, the dynamic mirroring function **200** is arranged to configure one or more of the portals **412** to mirror one or more packets, including portions of packets, based on selected criteria. The configuration has several dynamic

30

field or selected fields in the packet. The selected field or fields may be one or more of: a) address fields; b) protocol fields; c) length or byte count fields; and d) fields used in determining the value, meaning, placement or inclusion of other fields in the packet. The second selection criterion may also be a selection of the fields in the packet to be mirrored, byte count of a first selected portion of the packet to be mirrored and/or offset of a first selected portion of the packet to be mirrored. Further, the third selection criterion may be: a) a count setting of the frames meeting the first selection criterion; b) created using information contained in the packets mirrored or to be mirrored; c) created using information contained in packets not mirrored to the selected portal where mirroring occurs. The application layer of the packet or series of packets may also be used as either or both of the first and second criteria. The control engine **410** is configured to include one or more inputs used in the analysis of what and when its mirror traffic from the traffic source 404. Input

mirroring occurs. The application layer of the packet or series of packets may also be used as either or both of the first and second criteria. The control engine **410** is configured to

Examples of such triggering events or conditions have been described above and are incorporated herein but the invention is not limited to those so listed.

The portals **412** may be configured by any means known to those skilled in the art, such as through the use of SNMP, to provide the parameters needed to determine or discover the portal destination, path to it and the transport criteria needed. As an example, a mirror may have been established by the policy-based mirroring function to mirror all traffic from a new attached function that has joined the network system **100**. After a period of time based on one or more of policies, on the output of an IDS, the output of the appliance **180** or another network infrastructure device including the application identification function **200**, and Network Access and Control (NAC) functions, the particular attached function might have the criteria for mirroring changed to reflect only mirroring, for example, the first 30 frames of a new flow to only the least loaded application identification function **200** of the network infrastructure **101**. This might reflect an attached function now having a trusted, known, authenticated user, updated applications and virus detection status, and generally displaying good network behavior.

It is also to be noted that the control engine **410** configured to change the mirroring dynamically may be located in one or more other devices of the network infrastructure **101**. Further, the control mechanism of the traffic mirror function may be automatically changed during the mirroring of traffic based on input of information to the control engine **410** that initiates a mirroring change based on network policy, mirroring policies or both. The first selection criterion may be based on a

200 including the engine **186** would need to be deployed throughout the network infrastructure **101** to insure a reasonable detection of a sufficient portion of the traffic flows important to the organization to permit reasonable control. The dynamic traffic mirroring function of the present invention dramatically eases the pain, expense and complexity of engine **186** deployment allowing such application identification engines to be placed almost at will by the network administrator, with the mirrors bringing to it the necessary traffic from any device of interest in the network infrastructure **101**. The dynamic mirrors are most logically located in the switches and other devices that are primarily packet forwarding devices, such as switches and routers, as the incremental additional cost is relatively minor and heavily leverages the already existing capabilities these devices have; namely, to classify all incoming traffic, copy traffic as needed, rapidly modify frames and frame headers, filter and forward frames, and to support encapsulations, tunneling and routing technologies. Moreover, their positions in various network topologies make them almost ideal for the task.

Examples of network topology organizational options are shown in FIGS. **7-10**. Components of the network that are numbered the same as in FIG. **1** remain the same for purposes of describing the different topology options. FIG. **7** shows a first network topology **700** in which transmissions from attached functions to the network infrastructure **101** enter through edge network devices and all Internet traffic is forced through firewall **118**, which may be one or more firewalls scaled as desired based on the size of the network. The firewall **118** performs its gate keeping function and packets that

CISCO EXHIBIT 1009