(12) UK Patent (19) GB (11) 2531677 (13) B

(45) Date of B Publication 22.07.2020

(54) Title of the Invention: A network security system

(51) INT CL: ***H04L 29/06*** (2006.01) ***G06F 21/55*** (2013.01)

(21) Application No: **1602072.9**

(22) Date of Filing: **04.04.2014**

Date Lodged: **05.02.2016**

(30) Priority Data:
(31) **1312298.1** (32) **09.07.2013** (33) **GB**

(86) International Application Data:
PCT/IB2014/060428 En 04.04.2014

(87) International Publication Data:
WO2015/004543 En 15.01.2015

(43) Date of Reproduction by UK Office **27.04.2016**

(72) Inventor(s):
Ian Robertson

(73) Proprietor(s):
International Business Machines Corporation
New Orchard Road, Armonk 10504, New York, United States of America

(74) Agent and/or Address for Service:
IBM United Kingdom Limited
Intellectual Property Law, Hursley Park, WINCHESTER, Hampshire, SO21 2JN, United Kingdom

(56) Documents Cited:
JP 2003067269 A

(58) Field of Search:
As for published application 2531677 A viz:
INT CL **G06F**
updated as appropriate

Additional Fields
INT CL **H04L**
Other: **WPI, EPODOC**

GB 2531677 B

FIG. 1

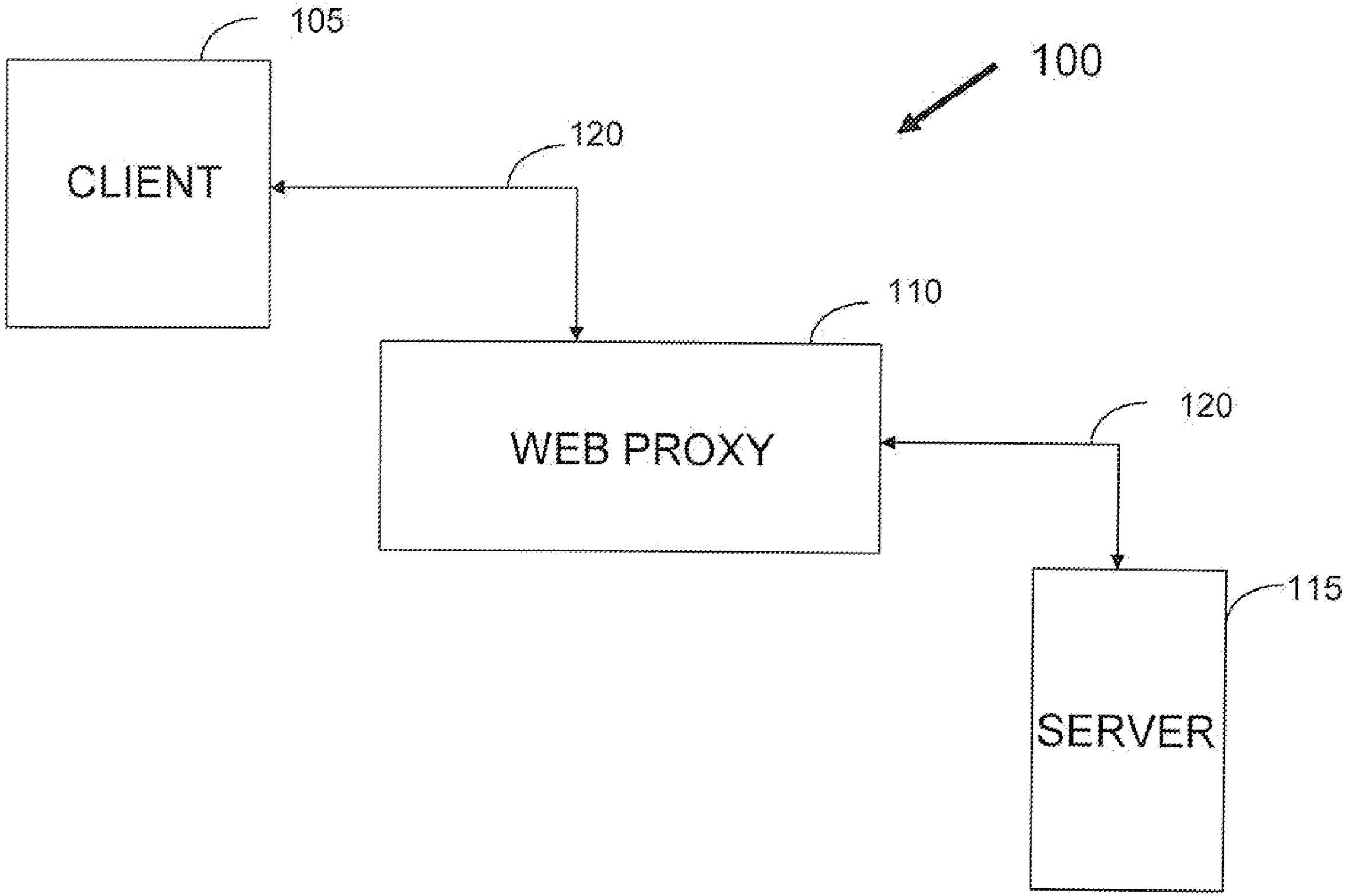


FIG. 2

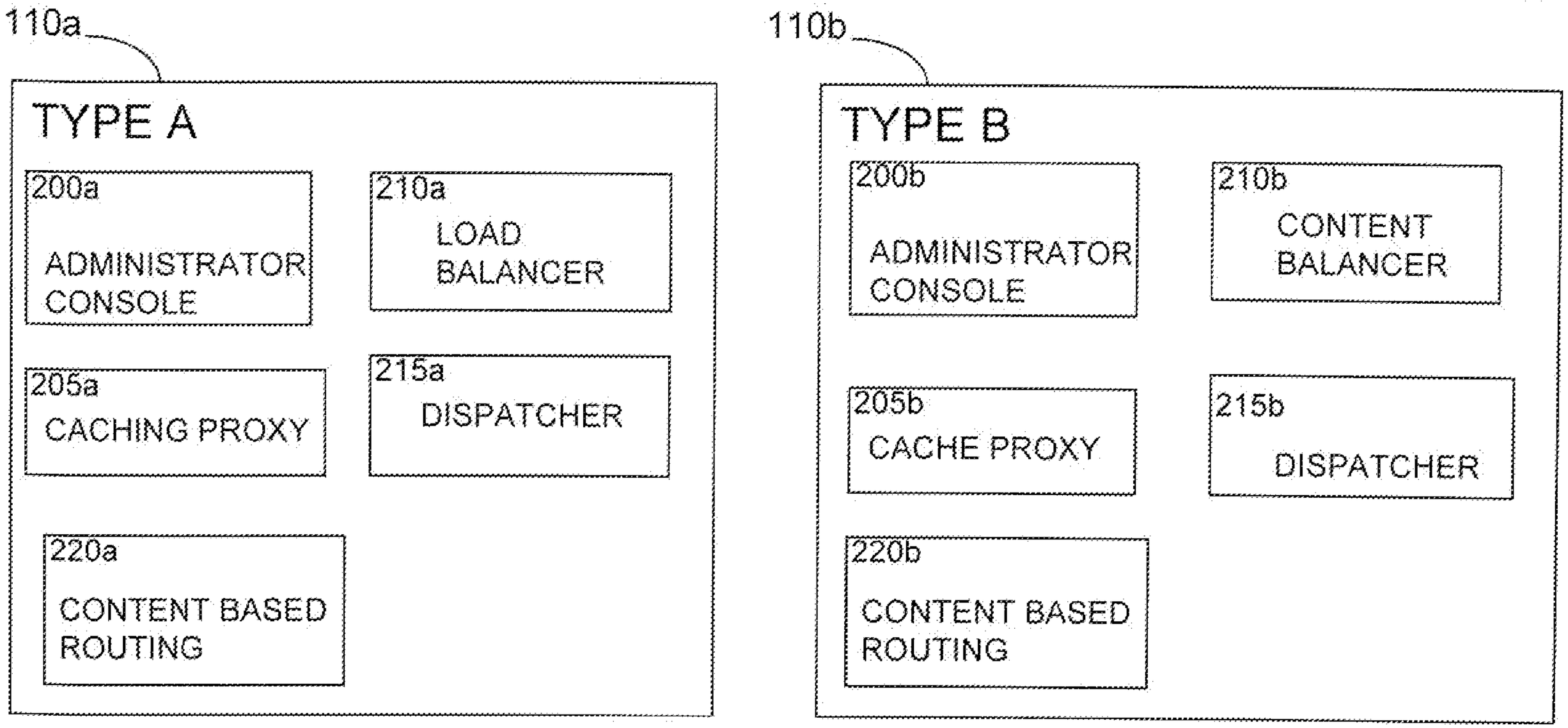

110a
TYPE A
200a
ADMINISTRATOR CONSOLE
210a
LOAD BALANCER
205a
CACHING PROXY
215a
DISPATCHER
220a
CONTENT BASED ROUTING
110b
TYPE B
200b
ADMINISTRATOR CONSOLE
210b
CONTENT BALANCER
205b
CACHE PROXY
215b
DISPATCHER
220b
CONTENT BASED ROUTING

FIG. 3

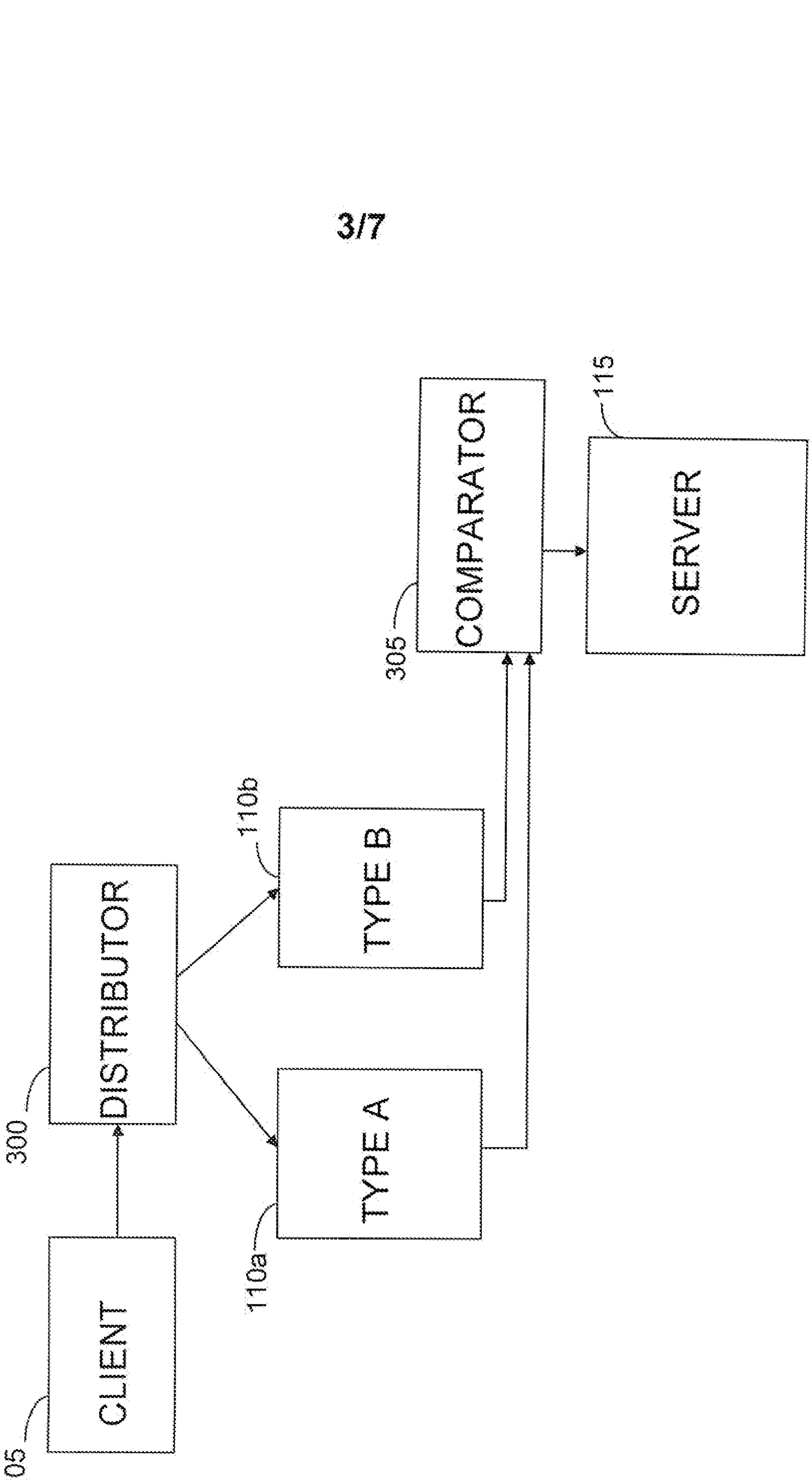

105
CLIENT
300
DISTRIBUTOR
110a
TYPE A
110b
TYPE B
305
COMPARATOR
SERVER

FIG. 4

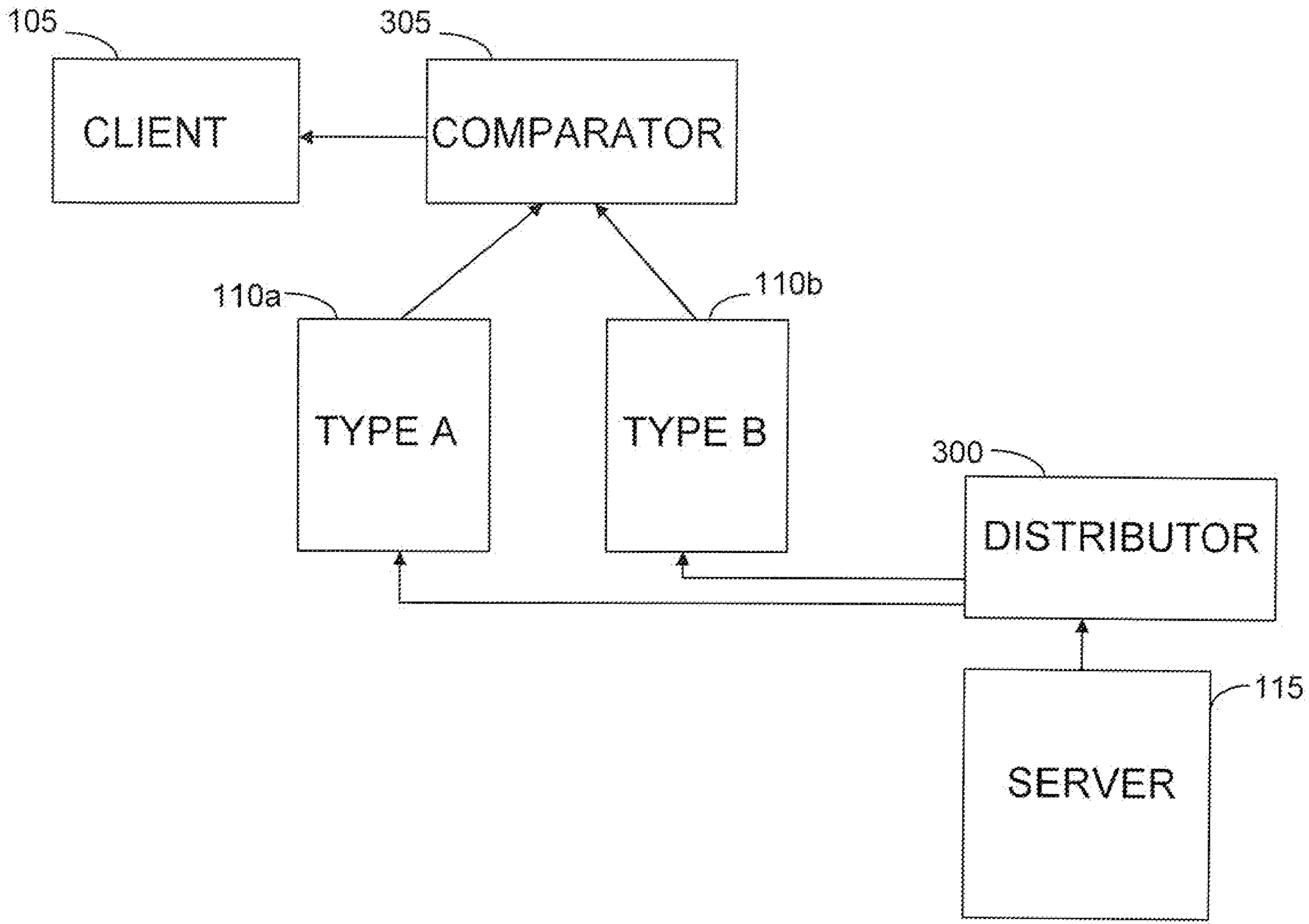

105
CLIENT
305
COMPARATOR
110a
TYPE A
110b
TYPE B
300
DISTRIBUTOR
115
SERVER

FIG. 5a

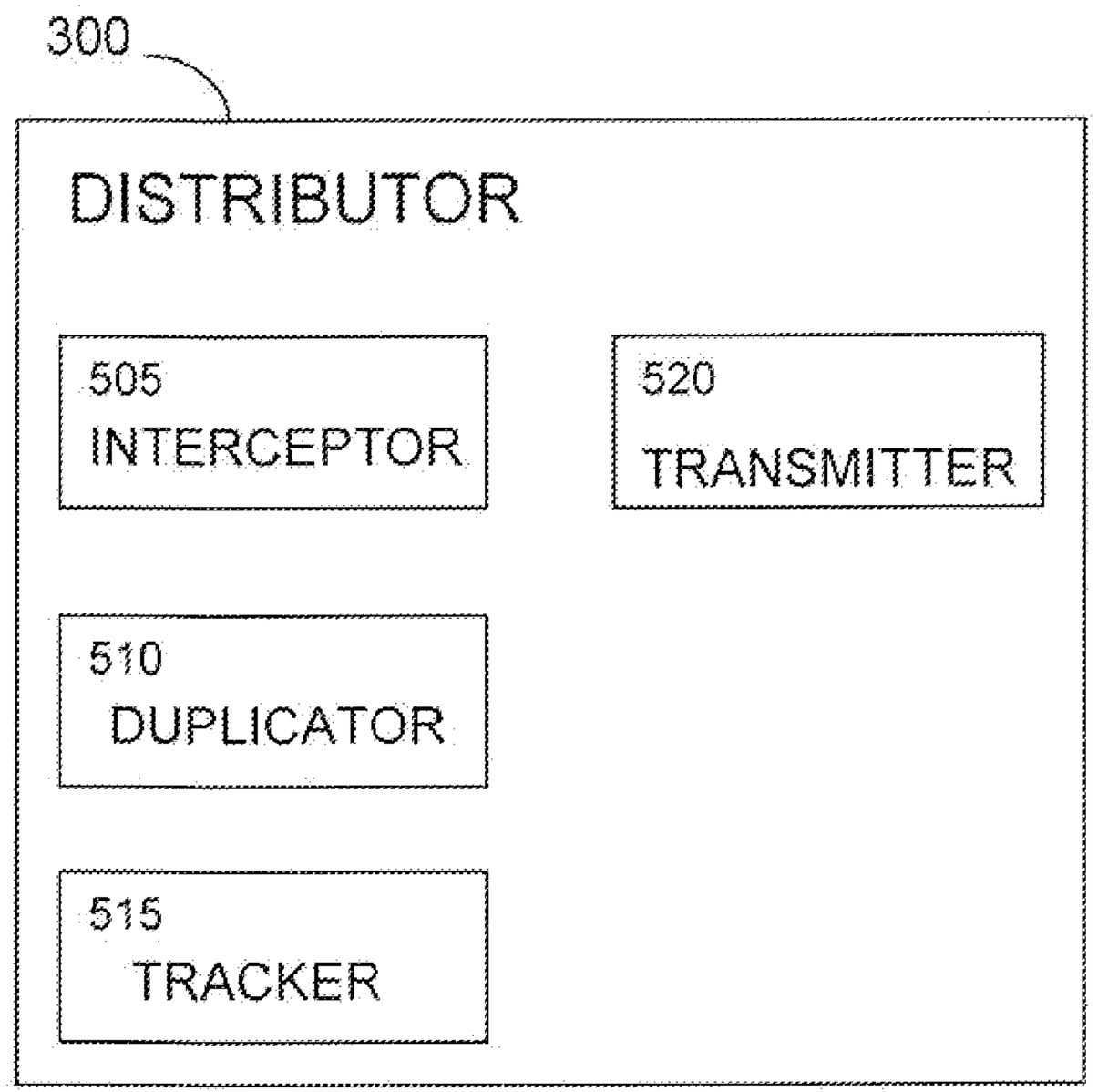

300
DISTRIBUTOR
505
INTERCEPTOR
520
TRANSMITTER
510
DUPLICATOR
515
TRACKER

FIG. 5b

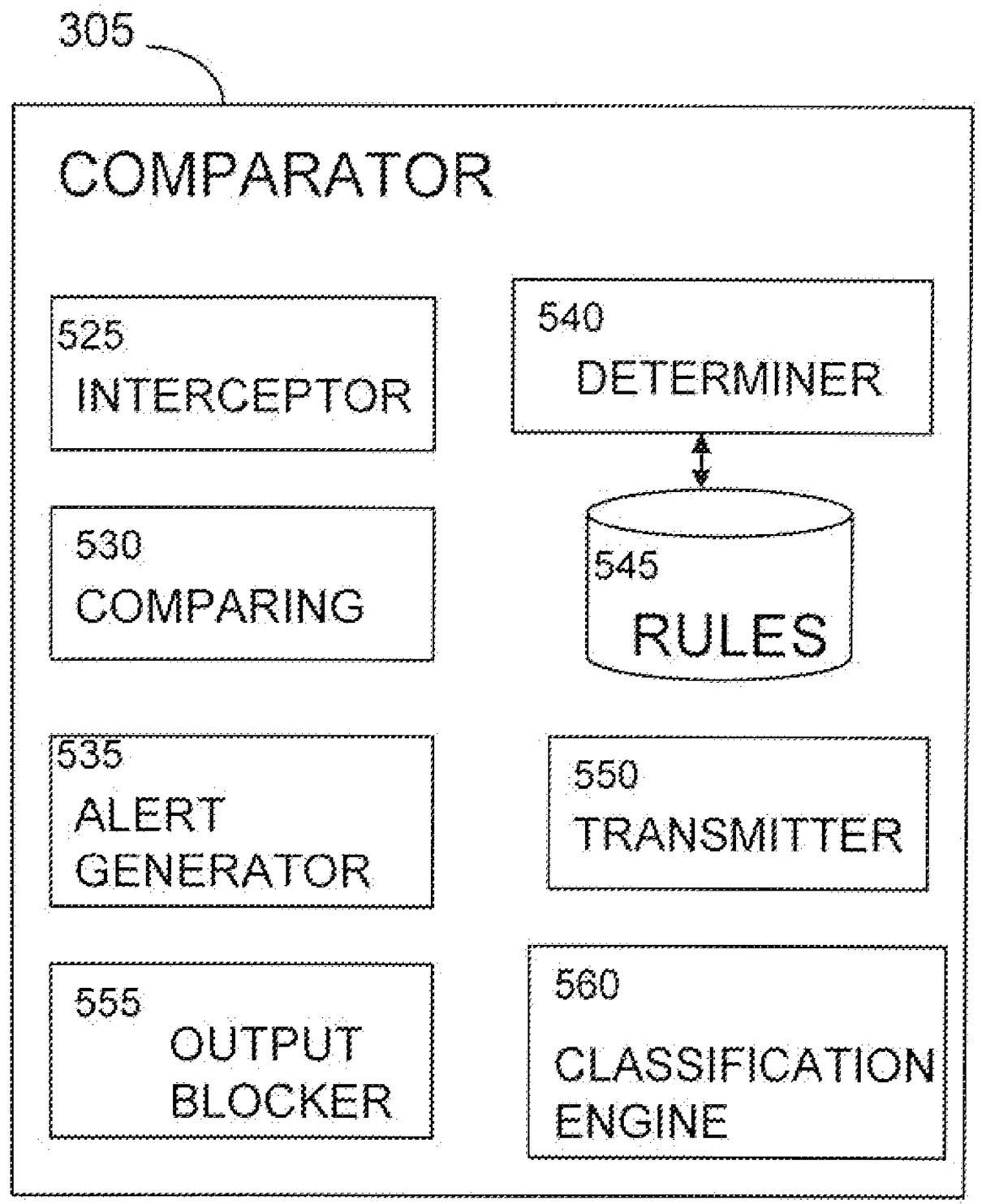

305
COMPARATOR
525
INTERCEPTOR
540
DETERMINER
530
COMPARING
545
RULES
535
ALERT
GENERATOR
550
TRANSMITTER
555
OUTPUT
BLOCKER
560
CLASSIFICATION
ENGINE

FIG. 6

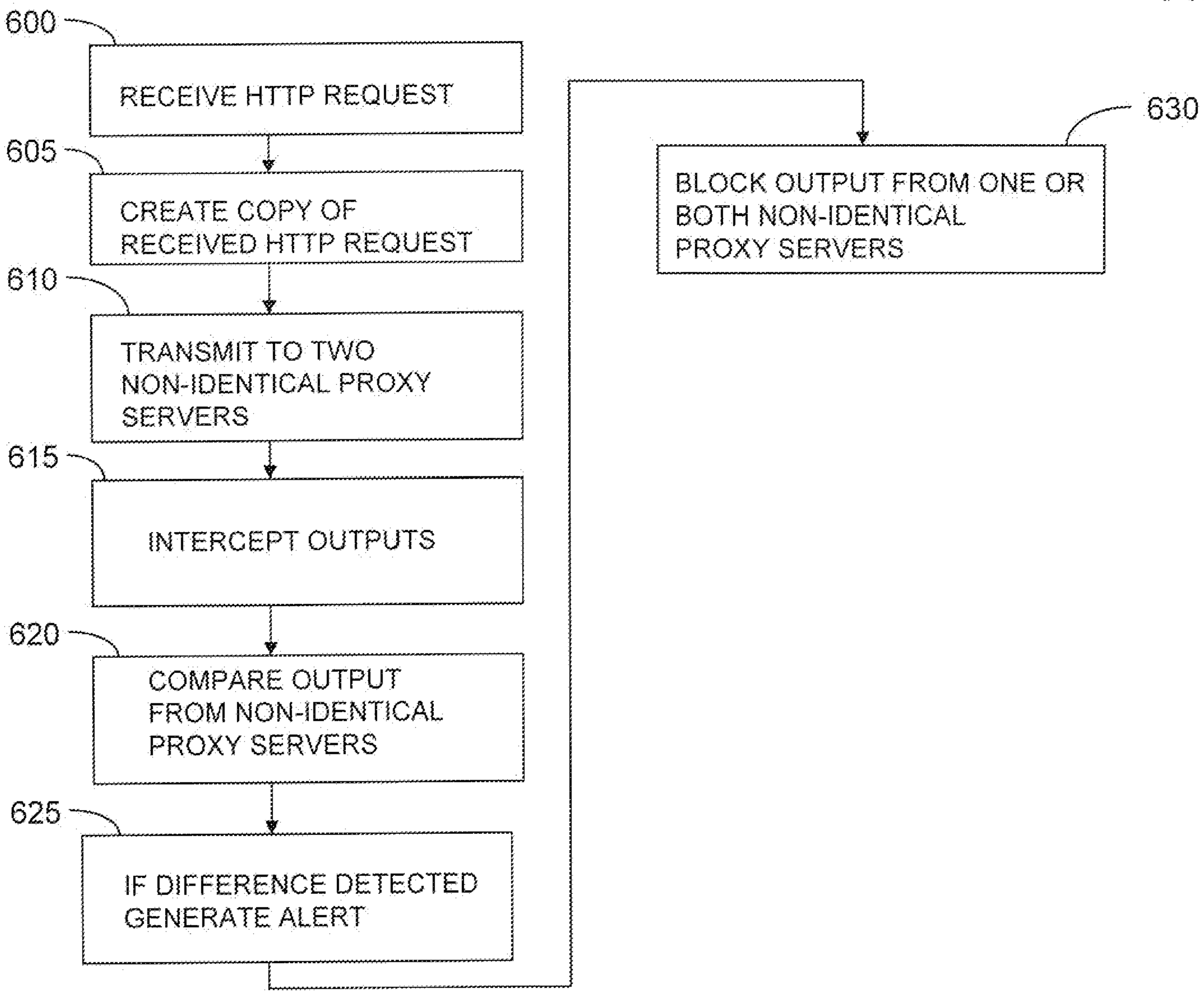
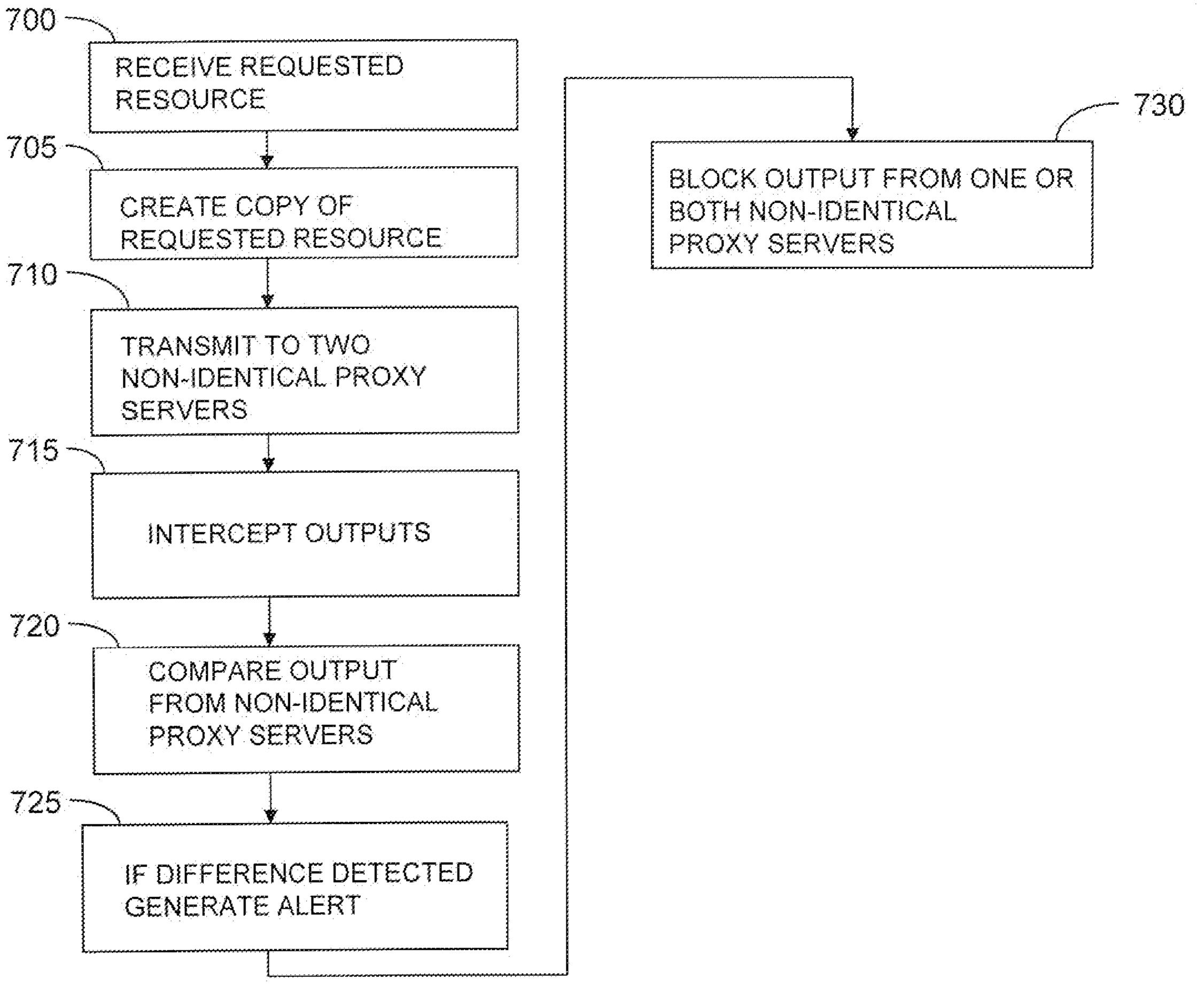

600
RECEIVE HTTP REQUEST
605
CREATE COPY OF RECEIVED HTTP REQUEST
610
TRANSMIT TO TWO NON-IDENTICAL PROXY SERVERS
615
INTERCEPT OUTPUTS
620
COMPARE OUTPUT FROM NON-IDENTICAL PROXY SERVERS
625
IF DIFFERENCE DETECTED GENERATE ALERT
630
BLOCK OUTPUT FROM ONE OR BOTH NON-IDENTICAL PROXY SERVERS

FIG. 7


700
RECEIVE REQUESTED RESOURCE
705
CREATE COPY OF REQUESTED RESOURCE
710
TRANSMIT TO TWO NON-IDENTICAL PROXY SERVERS
715
INTERCEPT OUTPUTS
720
COMPARE OUTPUT FROM NON-IDENTICAL PROXY SERVERS
725
IF DIFFERENCE DETECTED GENERATE ALERT
730
BLOCK OUTPUT FROM ONE OR BOTH NON-IDENTICAL PROXY SERVERS

A NETWORK SECURITY SYSTEM

Field of the invention

[001] The invention relates to the field of network security and in particular, the present invention relates to a system, method and a computer program for detecting a security vulnerability in a data processing network.

Background of the invention

[002] A web proxy server is a server that acts as an intermediary for requests from client devices requesting resources from servers in a network. A client device typically sends a request for a resource over the network and begins by connecting to a web proxy server. The web proxy server evaluates the request, performs a number of functions and fetches the requested resource and sends it back to the requesting client. A web proxy server may perform a number of functions, for example, enabling client devices downstream of the proxy server to maintain anonymity, speeding up access to resources using techniques such as caching for preventing download of the same content multiple times, auditing of access to internet resources, scanning of transmitted content for malware before delivery to requesting clients and scanning of outbound content for data loss prevention and other access and management tasks.

[003] Proxy servers, in particular web proxy servers are vulnerable to attack. Often attackers look for vulnerabilities in the design and the source code of components that perform some of the functions already mentioned. Vulnerabilities can be exploited to alter the behaviour of the software components and potentially threaten the confidentiality, integrity and/or availability of data within the proxy server itself or a larger network system that the proxy server is part of.

[004] Typically this type of vulnerability exploitation attack is mitigated by using one or more specialist components to protect the web proxy server. These components can include firewalls, intrusion detection systems, web application firewalls and virus scanners. Typical

vulnerabilities in web proxy servers allow an attacker to send carefully crafted messages to a component which can result in an attacker being able to take control of the web proxy server and use the web proxy server as a "launch pad" for attacks on "downstream" components of the network system. The impact of such attacks can be to degrade the confidentiality, integrity or availability of information within the overall system.

[005] Suppliers of web proxy software make every effort to avoid introducing vulnerabilities into their products and typically act quickly to remedy any problems that are identified by releasing software "patches" which replace or supplement the defective code. However, this approach has been found to be unsatisfactory for several reasons:

- Vulnerabilities are often exploited faster than manufactures can create and disseminate patches;
- Means to exploit known vulnerabilities are frequently published (on the internet) and consequently wide-spread attacks can be perpetrated by attackers who have little skill or knowledge themselves; and
- The process of applying patches is frequently delayed while patched copies of the web proxies are tested. This adds additional cost and complexity to the process and many large scale commercial systems remain "unpatched" for a considerable length of time. The UK Government agency responsible for information assurance (CESG) estimates that the majority (70% to 80%) of successful attacks on IT systems in the Public Sector could be avoided if all components were correctly patched.

[006] Despite technical advances made in the design of such specialist components over the years, successful attacks on web proxy servers still continue and are one of the main sources of risk in large commercial information processing systems.

Summary of the invention

[007] Viewed from a first aspect, the present invention provides a network security system for generating an alert in response to a security breach being detected in a proxy server in a data processing network, the system comprising: a distributor component comprising: an interceptor component for intercepting a request for a resource from a client device; and a

duplicator component for creating a duplicate of the intercepted request and forwarding the intercepted request to a first type of proxy server and forwarding the duplicate of the intercepted request to a second type of proxy server, where in the first type of proxy server and the second type of proxy server are non-identical proxy server types; a comparator component comprising: an interceptor component for intercepting the output from the first type of proxy server and the second type of proxy server; a comparing component for comparing the output of the first type of proxy server to the output of the second type of proxy server; a determining component determining whether the output of the first type of proxy server and the output of the second type of proxy server differ, in response to comparing the outputs, and an alert generator component for generating an alert, in response to determining a difference in the outputs; and further comprising the determiner component determining that the output from the first type of proxy server and the second type of proxy server do not differ and a routing component for routing the output from one of the first type of proxy server or the second type of proxy server to a server for fetching the requested resource.

[008] Preferably, the present invention provides a system further comprising an output blocker of the comparator component detecting that an alert has been generated and blocking the output from one or both of the first type of proxy server and the second type of proxy server.

[009] *[deleted paragraph]*

[010] Preferably, the present invention provides a system further comprising the determiner component detecting an action to be performed in each of the outputs and further determining that one or both of the actions to be performed are different actions.

[011] Preferably, the present invention provides a system further comprising the determiner component determining that one of the actions to be performed will cause a processing error in one or both of the first type and second type of web proxy servers or the server.

[012] Preferably, the present invention provides a system further comprising the determiner component determining that the processing error is a security breach.

[013] Viewed from a second aspect, the present invention provides network security system for generating an alert in response to a security breach being detected in a proxy server in a data processing network, the system comprising: a distributor component comprising: an interceptor component for intercepting a requested resource from a server; and a duplicator component for creating a duplicate of the intercepted requested resource and forwarding the intercepted requested resource to a first type of proxy server and forwarding the duplicate of the intercepted requested resource to a second type of proxy server, where in the first type of proxy server and the second type of proxy server are non-identical proxy servers; a comparator component comprising: an intercepting component for intercepting the output from the first type of proxy server and the second type of proxy server; a comparing component for comparing the output of the first type of proxy server to the output of the second type of proxy server; a determining component for determining whether the output of the first type of proxy server and the output of the second type of proxy server differ, in response to comparing the outputs, and an alert generator component for generating an alert, in response to determining a difference in the outputs; and further comprising the determiner component determining that the output from the first type of proxy server and the second type of proxy server do not differ and a routing component for routing the output from one of the first type of proxy server or the second type of proxy server to a server for fetching the requested resource.

[014] Preferably, the present invention provides a system further comprising an output blocker of the comparator component detecting that an alert has been generated and blocking the output from one or both of the first type of proxy server and the second type of proxy server.

[015] *[deleted paragraph]*

[016] Preferably, the present invention provides a system further comprising the determiner component detecting an action to be performed in each of the outputs and further determining that one or both of the actions to be performed are different actions.

[017] Preferably, the present invention provides a system further comprising the determiner component determining that one of the actions to be performed will cause a processing error in one or both of the first type and second type of web proxy servers or the client device.

[018] Preferably, the present invention provides a system further comprising the determiner component determining that the processing error is a security breach.

[019] Viewed from a third aspect, the present invention provides a method for generating an alert in response to a security breach being detected in a proxy server in a data processing network, the method comprising: intercepting a request for a resource from a client device; creating a duplicate of the intercepted request and forwarding the intercepted request to a first type of proxy server and forwarding the duplicate of the intercepted request to a second type of proxy server, where in the first type of proxy server and the second type of proxy server are non-identical proxy server types; intercepting the output from the first type of proxy server and the second type of proxy server; comparing the output of the first type of proxy server to the output of the second type of proxy server; determining whether the output of the first type of proxy server and the output of the second type of proxy server differ, in response to comparing the outputs, and generating an alert, in response to determining a difference in the outputs; and further comprising determining that the output from the first type of proxy server and the second type of proxy server do not differ and a routing component for routing the output from one of the first type of proxy server or the second type of proxy server to a server for fetching the requested resource.

[020] Viewed from a fourth aspect, the present invention provides a method for generating an alert in response to a security breach being detected in a proxy sever of a data processing network, the system comprising: intercepting a requested resource from a server; and creating a duplicate of the intercepted requested resource and forwarding the intercepted requested resource to a first type of proxy server and forwarding the duplicate of the intercepted requested resource to a second type of proxy server, where in the first type of proxy server and the second type of proxy server are non-identical proxy servers; intercepting the output from the first type of proxy server and the second type of proxy server; comparing the output of the first type of proxy server to the output of the second type

of proxy server; determining whether the output of the first type of proxy server and the output of the second type of proxy server differ, in response to comparing the outputs and generating an alert, in response to determining a difference in the outputs; and further comprising determining that the output from the first type of proxy server and the second type of proxy server do not differ and a routing component for routing the output from one of the first type of proxy server or the second type of proxy server to a server for fetching the requested resource.

[021] Viewed from a fifth aspect, the present invention provides a computer program comprising computer program code to, when loaded into a computer system and executed, perform all the steps of the method as described above.

Brief Description of the Drawings

[022] A preferred embodiment of the present invention will now be described by way of example only, with reference to the accompanying drawings in which:

Figure 1 is a block diagram illustrating a network security system in which the present invention may be implemented in accordance with preferred embodiment of the present invention;

Figure 2 is a block diagram illustrating the components of a proxy server as is known in the art;

Figure 3 is a block diagram illustrating a process flow of a request for resources from a plurality of client devices to a distributor component and a comparator component in accordance with a preferred embodiment of the present invention;

Figure 4 is a block diagram illustrating a process flow of the request for resources of Figure 4 being satisfied by a server and communicated to a distributor component and a comparator component in accordance with a preferred embodiment of the present invention;

Figures 5a and 5b are block diagrams illustrating the sub components of the comparator and the distributor component of Figure 3 and 4;

Figure 6 is a flow chart illustrating a process flow of an HTTP request through a distributor component and a comparator component in accordance with a preferred embodiment of the present invention; and

Figure 7 is a flow chart illustrating a process flow of a fetched request through a distributor component and comparator component in accordance with a preferred embodiment of the present invention.

Detailed Description of the Invention

[023] Aspects of the present invention are described below with reference to flowchart illustrations and/or block diagrams of methods, systems and computer program products

according to embodiments of the invention. It will be understood that each block of the flowchart illustrations and/or block diagrams, and combinations of blocks in the flowchart illustrations and/or block diagrams, can be implemented by computer program instructions. These computer program instructions may be provided to a processor of a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable data processing apparatus, create means for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks.

[024] These computer program instructions may also be stored in a computer readable medium that can direct a computer, other programmable data processing apparatus, or other devices to function in a particular manner, such that the instructions stored in the computer readable medium produce an article of manufacture including instructions which implement the function/act specified in the flowchart and/or block diagram block or blocks.

[025] The computer program instructions may also be loaded onto a computer, other programmable data processing apparatus, or other devices to cause a series of operational steps to be performed on the computer, other programmable apparatus or other devices to produce a computer implemented process such that the instructions which execute on the computer or other programmable apparatus provide processes for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks.

[026] Referring to Figure 1, a simplified example of a known networked security system 100 in which a preferred embodiment of the present invention may be implemented, is shown. The network security system, of Figure 1, comprises a client device 105, a server 115, a network 120 and a web proxy server 110.

[027] Examples of client devices 105 include, but are not limited to, personal computer systems, thin clients, hand-held or laptop devices, mobile devices, set top boxes, programmable consumer electronics, network PCs, minicomputer systems etc.

[028] Client devices 105 send requests for resources, across a network 120, using a web browser or other type of software which is suitable for sending and receiving requests for resources across the network 120.

[029] An example of a network 120 is the internet, but any other type of network 120 in which a first device sends a request/communication for resources to a second device may be suitable.

[030] The client device 105, typically, submits a Hyper Text Transfer Protocol (HTTP) request message to a server. The HTTP request message typically comprises a request for a resource. A resource is defined as a chunk of information or data that can be identified by a URL. A common type of resource is a file, but a resource may also be a dynamically generated query result, the output of a CGI script, a document, an image or a music file etc.

[031] A Server 115 receives the HTTP request message and either sends the requested resource back to the client device 105 or sends the request on to another server 115 that can satisfy the client request. The requested resource is sent back in the form of a response message which contains additional information such as status information, length of message etc. The server 115 can be any type of server 115 that is capable of storing and satisfying requests from client devices 115.

[032] A web proxy server 110 intercepts requests for resources from the client 105 to the server 115 and vice versa when the server 115 fetches the resource for sending back to the client device 105 (reverse proxy mode). Typically, a client device 105 sends a request to a server 115 requesting a resource, such as a file, connection, web page or other type of resource from one or more servers 115. The request is intercepted by the web proxy server 110 and the web proxy server 110 evaluates the received request before fetching the requested resources from one or more severs 115.

[033] A web proxy server 110 can perform many different types of functions. Some of the main tasks performed by a web proxy server 110 are:

- Providing security functions, such as keeping the IP address of client machines anonymous,
- Caching of resources
- Scanning outbound and inbound resources for viruses

[034] However, a person skilled in the art will realize that there are many other types of functions that a web proxy server can perform and the above mentioned functions are merely for illustration purposes and are not intended to limit the scope of protection.

[035] A simplified example of process flow from client 105 to web proxy server 110 to a server 115 is given below:

- Employee A launches a web browser on a client device 105 and types in the URL www.IBM.com
- The resource is routed to a web proxy server 110 and the web proxy server 110 checks the IP address of the request against administrator defined access control.
- If it is determined that the IP address is not forbidden the web proxy server 110 executes the request by sending a request message to the server 115 storing the resource.
- The web proxy server 115 receives the requested resource and stores a copy of the resource in its cache.
- The proxy server transmits the requested resource to the requesting client device 105.

[036] Typical components of a web proxy server 110 are illustrated in Figure 2 and include an administrator console 200a, 200b, a load balancer 210a, 210b, a caching proxy 205a, 205b, a dispatcher component 215a, 215b and a content routing component 220a, 220b. A person skilled in the art will realize that other sub components can be integrated within or interface to the web proxy server 110 without departing from the scope of protection.

[037] The administrator console 200a, 200b provides an interface for an administrator to configure, maintain and control the web proxy server 110a, 110b and its sub components.

[038] A caching proxy 205a, 205b is deployed for preventing download of the same content multiple times. Either passive caching or active caching is deployed. The caching

proxy 205a, 205b reduces bandwidth use and improves a web site's speed and reliability by providing a point-of-presence node for one or more back-end content servers 115. The caching proxy 205a, 205b can cache and serve static content and content dynamically generated by, for example IBM's WebSphere Application Server.

[039] The caching proxy 205a, 205b may be configured in the role of a reverse proxy server 110 or a forward proxy server 110, providing either a point-of-presence for an external network, such as the internet 120, or an internal network (not illustrated).

[040] The caching proxy 205a, 205b intercepts data requests from the client 105, retrieves the requested information from the server(s) 115, and delivers that content back to the client 105. Most commonly, the requests are for documents stored on servers and delivered using the Hypertext Transfer Protocol (HTTP). However, a person skilled in the art will realize that a proxy server can be configured to handle other protocols, such as File Transfer Protocol (FTP) and Gopher etc.

[041] The caching proxy 205a, 205b stores cacheable content in a local cache before delivering it to the requesting client 105. Examples of cacheable content may include static Web pages and JavaServer Pages files that may comprise dynamically generated, but infrequently changing, information. Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates. Caching enables the proxy server 110 to satisfy subsequent requests for the same content by delivering it directly from the local cache, which is much quicker than retrieving it again from the content host.

[042] A load balancer 210a, 210b creates edge-of-network systems that direct network traffic flow, reducing congestion and balancing the load on various other services and systems. Load balancer 210a, 210b provides site selection, workload management, session affinity, and transparent failover.

[043] Load balancer 210a, 210b intercepts data requests from clients 105 and forwards each request to the server 115 that is currently able to fulfill the request. In other words, it

balances the load of incoming requests among a defined set of servers 115 that service the same type of requests.

[044] A content based routing component 220a, 220b together with the caching proxy server 205a, 205b allow HTTP and HTTPS requests to be distributed based on URLs or other administrator-determined characteristics, eliminating the need to store identical content on all servers 115.

[045] Each web proxy server 110 may comprise a combination of the above components. However, each of the above components may operate in a different manner. This is akin to a car – wherein a car comprises a body, wheels, an engine, a steering wheel and a gear lever etc. Different car manufactures all produce cars with the aforementioned specification but each of the parts are designed and manufactured in a unique manner that enables car manufactures to distinguish their cars from one another, for example, car A is more fuel efficient than car B based on the aerodynamics of the chassis etc. Therefore, if web proxy servers 110 are designed/manufactures by different vendors ('Type A' web proxy server and 'Type B' web proxy server), typically 'Type A' and 'Type B' web proxy servers may behave differently in response to a given type of security attack. For example, the vendor of 'Type A' web proxy server may have released a 'security patch' to fix a security vulnerability of 'Type A'. However, the vendor of 'Type B' web proxy server may have not released a 'security patch' and therefore the vulnerability still exists in 'Type B' web proxy server. Generating a security patch can take many days and leave the web proxy server 110 and the entire network 100 vulnerable to security attacks.

[046] In order to protect the web proxy server 110 and make the patching process easier and more reliable and with reference to Figure 3, 4 and 5, a distributor component 300 and a comparator component 305 are incorporated into the network security system 100, in accordance with a preferred embodiment of the present invention. The distributor component 300 intercepts requests for resource from clients 105 or intercepts the requested resources from servers 115 as these requests are on route to the web proxy server 110. In a preferred embodiment at least two non-identical web proxy servers 110a, 110b are deployed. The term 'non-identical' is used throughout this specification to mean different types of web

proxy server i.e. 'not the same' or not 'identical'. This can mean web proxy servers 110 from different manufactures, different vendors, or different versions of software residing on the server 115.

[047] The distributor 300 routes the intercepted request to at least two non-identical web proxy severs 110a, 110b and the web proxy servers 110a, 110b continue to process the received requests in the normal manner.

[048] The comparator component 305 analyses the output from the two non-identical web proxy servers 110a, 110b. The comparator component 300 and the distributor component 305 interact and communicate with clients 105, servers 115 and web proxy servers 110 to enhance network security and thwart unwanted security attacks.

[049] In a preferred embodiment, for incoming message i.e. messages being sent from the client 105 to the web proxy server 110a, 110b, a distributor component 300 is introduced into the network security system 100 'upstream' of the web proxy servers 110a, 110b. By 'upstream', it is meant that the distributor component 300 is located in the network 100, between one or more requesting clients 105 and at least two non-identical web proxy servers 110a, 110b.

[050] In a preferred embodiment, for incoming message i.e. messages being sent from the client 105 to the web proxy server 110a, 110b, a comparator component 305 is introduced into the network security system 100 'downstream' of the web proxy server 110a, 110b . By 'downstream', it is meant that the comparator component 305 is located in the network 100 between two or more web proxy servers 110a, 110b and one or more servers 115 which satisfy the request for resources.

[051] In a preferred embodiment, for outgoing messages i.e. messages being sent from the server 115 to at least two web proxy servers 110a, 110b, for ongoing distribution to a requesting client 105, a distributor component 300 is introduced into the network security architecture 100 'upstream' of the web proxy server 110a, 110b. By 'upstream' it is meant

that the distributor component 300 receives requests from servers 115 for sending to two or more web proxy servers 110a, 110b.

[052] In a preferred embodiment, for outgoing messages i.e. messages being sent from the server 115 to at least two web proxy servers 110a, 110b, a comparator component 305 is introduced into the network security architecture 'downstream' of the web proxy server 110a, 110b. By 'downstream', it is meant that the comparator component 305 is located in the network 120 between two or more web proxy servers 110a, 110b and one or more requesting client devices 105.

[053] The comparator 305 and distributor 300 may be separate components residing on the same server, or alternatively, the separate components may reside on separate servers, in the same or geographically dispersed area. Alternatively, the comparator 305 and the distributor 300 may be implemented as one software component and residing on the same server i.e. the distributor 300 intercepting requests from clients 105 and servers 115 and the comparator 305 intercepting outputs from the non-identical web proxy servers 110a, 110b. A person skilled in the art will realise that other types of arrangements are possible without departing from the scope of the invention.

[054] With reference to Figure 5a and Figure 6, the distributor component 300 comprises a number of sub components that interface and interact with each other in order to process HTTP requests from requesting client devices 105. The sub components comprise a interceptor component 505 for intercepting and receiving requests from requesting client devices 105, a duplicator component 510 for creating copies of the received request, a tracker component 515 for assigning a unique identifier to the copies of the received requests and a transmitter component 520 for forwarding each of the copies of the received request to the non-identical web proxy servers 'Type A' 110a and 'Type B' 110b.

[055] The process begins with the interceptor component 505 receiving the HTTP request from the requesting client device 105 (step 600). A simplified example of a received HTTP request is shown below:

HTTP Request

GET /index.html HTTP/1.1

Host: www.IBM.com

[056] A typical attack is to make a request for a URL using standard HTTP Get or Post commands but to include sufficient data to cause the target program to store the data in memory is such a way that it overwrites the existing memory values which may be executable instructions. In this way, an attacker can modify the behaviour of a web server and potentially cause it to fail or to force it to execute instructions supplied by the attacker.

[057] This is called a buffer overflow condition and a buffer overflow condition exists when a program attempts to put more data in a buffer than it can hold or when a program attempts to put data in a memory area outside the designated address space. In this case, a buffer is a sequential section of memory allocated to contain anything from a character string to an array of integers. Writing outside the bounds of a block of allocated memory can corrupt data, crash the program, or cause the execution of malicious code.

[058] The basis of the vulnerability is improper data size checking. The data is represented in the URL request by a string. Improper string length checking typically takes place when wide or multi-byte character strings are mistaken for standard character strings.

[059] The following example illustrates the C programming language source code that is frequently used to allocate buffer space to hold data. The buffer is allocated dynamically using the C routine "malloc".

[060] The following example would be exploitable if any of the commented incorrect malloc calls were used.

Example code

#include <stdio.h>

#include <strings.h>

#include <wchar.h>

```
int main() {

wchar_t wideString[] = L" The spazzy orange tiger jumped " \
"over the tawny jaguar.";
wchar_t *newString;

printf("Strlen() output: %d\nWcslen() output: %d\n",
strlen(wideString), wcslen(wideString));

/* The following is incorrect because it is not appropriate to find the length of a string which
has been defined as a "wide string//

newString = (wchar_t *) malloc(strlen(wideString));
*/

/* The following string is incorrect because wide characters aren't 1 byte long //

newString = (wchar_t *) malloc(wcslen(wideString));
*/
/* correct! */
newString = (wchar_t *) malloc(wcslen(wideString) *
sizeof(wchar_t));
/* ... */
}
```

[061] The output from the printf() statement would be:

Strlen() output: 0

Wcslen() output: 53

Source: https://www.owasp.org/index.php/Improper_string_length_checking available under the creative commons license.

[062] A duplicator component 510 of the distributor component 300 creates a copy of the received HTTP request (step 605).

[063] The tracker component 515 appends a unique identifier to the received request and the copy of the received request. The unique identifier is used for tracking the requests (original request and duplicate request) through the security network, such that, the comparator 305 can correctly recognise the received request and the duplicate request.

[064] The transmitter component 520 transmits the received request to a 'Type A' web proxy server 110a and the duplicate request to a second web proxy server to 'Type B' web proxy server 110b (step 610). A person skilled in the art will realise that it is irrelevant which web proxy server 110a, 110b the received request or the duplicate request is sent to as long as the selected two web proxy servers 110a, 110b are not identical i.e. are of different types.

[065] Both web proxy servers 110a, 110b receive the HTTP request for resources and proceeds to process the request. As the two web proxy servers 110a, 110b are not identical and operate in different ways the following scenario may take place.

[066] 'Type A' web proxy 110a proceeds to process the received request. In this example, 'Type A' web proxy server 110a does not correctly measure the size or type of the data that is contained within the received request and allocates memory space using one of the invalid methods in the above code snippet examples. 'Type A' web proxy server 110a consequently suffers a buffer overflow which allows code instructions to be overwritten by user data. The result will depend on the instructions supplied by the attacker but typically no data is forward to the rest of the network system 100 and instead control of the 'Type A' web proxy server 110a is returned to the user i.e. the user has effectively "logged in" to the 'Type A' web proxy server 110a as the system administrator.

[067] 'Type B' web proxy server 110b receives the duplicate request and proceeds to process the request. In this example 'Type B' web proxy validates the size of the data in the received URL request and allocates memory correctly.

[068] Consequently 'Type B' web proxy server 110b processes the received request by:

- returning an error message to the user and not forwarding any data into the rest of the system, or
- truncates the user supplied data until it is of a valid length and only forwards the remaining data into the rest of network system 100, or
- allocates sufficient memory for the data in the received request and avoids overwriting code instructions with data.

[069] With reference to Figure 5b and Figure 6, the comparator component 305 comprises an interceptor component 525 for intercepting the output from the web proxy servers 110a, 110b (step 615). The interceptor component 505 receives the output from 'Type A' web proxy server 110a (buffer overflow – data overwritten) and the output from 'Type B' web proxy server 110b (the correct output).

[070] A comparing component 530 parses and compares each of the received outputs. The comparing component 530 interfaces with a determiner component 540. The determiner component 540 in combination with a database of rules 545 determines if there is a 'difference' between the two outputs (step 620). If no difference is determined, a routing component 550 forwards the HTTP request (received request or the copy of the received request) to an appropriate server 115 for satisfying the request. If the determiner component 540 detects a difference between the output of 'Type A' and the output of 'Type B', an alert is generated (step 625), by an alert generator component 535 in order to block the execution of the output from both of the web proxy servers 110a, 110b (step 630).

[071] In another embodiment, the determiner component 540 and the database 545 comprise additional logic which enables the determiner component 540 to detect an 'expected' output from a web proxy server 110a, 110b of a particular type. Thus enabling the determiner component 540 to detect which output from a web proxy server 110a, 110b can be forward to a server 115 and which output from a web proxy server 110a, 110b should be blocked.

[072] The comparator 305 and distributor components 300 also operate in a reverse direction (Figure 4 and Figure 7). Once the server 115 has satisfied the request for resources, a reply message comprising the requested resources is transmitted back over the network 120 to a distributor component 300 and its sub components 505 to 515(step 700). The distributor component 300 proceeds to make a copy of the reply message (step 705) and appends a unique identifier to the reply message and the copy of the reply message. The received reply message and the copy of the reply message are transmitted to non-identical web proxy severs 110a, 110b for processing.

[073] The output of the non-identical web proxy servers 110a, 110b is sent to a comparator component 305 for processing (step 710). The comparator component 305 and its sub components 525 to 560 receive the output from 'Type A' web proxy server and 'Type B' web proxy server (step 715). A comparing component 530 compares both of the outputs to each other. The comparing component 530 interfaces with a determiner component 540 (in combination with a database of rules 545) and determines if there is a 'difference' between the two outputs (step 730). If no difference is determined, the determiner component 540 sends the reply message which includes the requested resources to a router component 550 for transmitting to the requesting client 105. If the determiner component 540 detects a difference between the two outputs, an alert is generated (step 725) in order to block the execution of the output from both the web proxy servers 110a, 110b (step 730).

[074] An alert is generated by the alert generator component 535 of the comparator component 305. The alert not only alerts the output blocker component 555 to block one or both of the outputs of the web proxy server (s) 110a, 110b because a malicious attack has been detected, but a classification engine 560 interacts with the alert generator component 535 to classify the type of attack such that a report can be generated detailing the action that the web proxy server 110a, 110b was going to perform and why this was determined to be a malicious attack. Types of attacks can be an SQL injection attack, security misconfiguration or a buffer overflow attack for example.

[075] A person skilled in the art will realise that although a preferred embodiment of the invention has been described in terms of using two non-identical web proxy servers and

comparing the out, more that two non-identical web proxy servers can be used without departing from the scope of the invention.

[076] As will be appreciated by one skilled in the art, aspects of the present invention may be embodied as a system, method or computer program product. Accordingly, aspects of the present invention may take the form of an entirely hardware embodiment, an entirely software embodiment (including firmware, resident software, micro-code, etc.) or an embodiment combining software and hardware aspects that may all generally be referred to herein as a "circuit," "module" or "system." Furthermore, aspects of the present invention may take the form of a computer program product embodied in one or more computer readable medium(s) having computer readable program code embodied thereon.

[077] Any combination of one or more computer readable medium(s) may be utilized. The computer readable medium may be a computer readable signal medium or a computer readable storage medium. A computer readable storage medium may be, for example, but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, or device, or any suitable combination of the foregoing. More specific examples (a non-exhaustive list) of the computer readable storage medium would include the following: an electrical connection having one or more wires, a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), an optical fiber, a portable compact disc read-only memory (CD-ROM), an optical storage device, a magnetic storage device, or any suitable combination of the foregoing. In the context of this document, a computer readable storage medium may be any tangible medium that can contain, or store a program for use by or in connection with an instruction execution system, apparatus, or device.

[078] A computer readable signal medium may include a propagated data signal with computer readable program code embodied therein, for example, in baseband or as part of a carrier wave. Such a propagated signal may take any of a variety of forms, including, but not limited to, electro-magnetic, optical, or any suitable combination thereof. A computer readable signal medium may be any computer readable medium that is not a computer

readable storage medium and that can communicate, propagate, or transport a program for use by or in connection with an instruction execution system, apparatus, or device.

[079] Program code embodied on a computer readable medium may be transmitted using any appropriate medium, including but not limited to wireless, wireline, optical fiber cable, RF, etc., or any suitable combination of the foregoing.

[080] Computer program code for carrying out operations for aspects of the present invention may be written in any combination of one or more programming languages, including an object oriented programming language such as Java, Smalltalk, C++ or the like and conventional procedural programming languages, such as the "C" programming language or similar programming languages. The program code may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and partly on a remote computer or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user's computer through any type of network, including a local area network (LAN) or a wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider).

CLAIMS

1. A network security system for generating an alert in response to a security breach being detected in a proxy server in a data processing network, the system comprising:

a distributor component comprising:

an interceptor component for intercepting a request for a resource from a client device; and

a duplicator component for creating a duplicate of the intercepted request and forwarding the intercepted request to a first type of proxy server and forwarding the duplicate of the intercepted request to a second type of proxy server, where in the first type of proxy server and the second type of proxy server are non-identical proxy server types;

a comparator component comprising:

an interceptor component for receiving the output from the first type of proxy server and the second type of proxy server;

a comparing component for comparing the output of the first type of proxy server to the output of the second type of proxy server;

a determining component determining whether the output of the first type of proxy server and the output of the second type of proxy server differ, in response to comparing the outputs,

an alert generator component for generating an alert, in response to determining a difference in the outputs; and

further comprising the determiner component determining that the output from the first type of proxy server and the second type of proxy server do not differ and a routing component for routing the output from one of the first type of proxy server or the second type of proxy server to a server for fetching the requested resource.

2. A system as claimed in claim 1, further comprising an output blocker of the comparator component detecting that an alert has been generated and blocking the output from one or both of the first type of proxy server and the second type of proxy server.

3. A system as claimed in claim 1 further comprising the determiner component detecting an action to be performed in each of the outputs and further determining that one or both of the actions to be performed are different actions.

4. A system as claimed in claim 3, further comprising the determiner component determining that one of the actions to be performed will cause a processing error in one or both of the first type and second type of web proxy servers or the server.

5. A system as claimed in claim 4 further comprising the determiner component determining that the processing error is a security breach.

6. A network security system for generating an alert in response to a security breach being detected in a proxy server in a data processing network, the system comprising:

a distributor component comprising:

an interceptor component for intercepting a requested resource from a server; and

a duplicator component for creating a duplicate of the intercepted requested resource and forwarding the intercepted requested resource to a first type of proxy server and forwarding the duplicate of the intercepted requested resource to a second type of proxy server, where in the first type of proxy server and the second type of proxy server are non-identical proxy servers;

a comparator component comprising:

an interceptor component for receiving the output from the first type of proxy server and the second type of proxy server;

a comparing component for comparing the output of the first type of proxy server to the output of the second type of proxy server;

a determining component for determining whether the output of the first type of proxy server and the output of the second type of proxy server differ, in response to comparing the outputs;

an alert generator component for generating an alert, in response to determining a difference in the outputs; and

further comprising the determiner component determining that the output from the first type of proxy server and the second type of proxy server do not differ and routing the

output from one of the first type of proxy server or the second type of proxy server to the client device.

7. A system as claimed in claim 6, further comprising an output blocker of the comparator component detecting that an alert has been generated and blocking the output from one or both of the first type of proxy server and the second type of proxy server.

8. A system as claimed in claim 6 further comprising the determiner component detecting an action to be performed in each of the outputs and further determining that one or both of the actions to be performed are different actions.

9. A system as claimed in claim 8, further comprising the determiner component determining that one of the actions to be performed will cause a processing error in one or both of the first type and second type of web proxy servers or the client device.

10. A system as claimed in claim 9 further comprising the determiner component determining that the processing error is a security breach.

11. A method for generating an alert in response to a security breach being detected in a proxy server in a data processing network, the method comprising:

intercepting a request for a resource from a client device;

creating a duplicate of the intercepted request and forwarding the intercepted request to a first type of proxy server and forwarding the duplicate of the intercepted request to a second type of proxy server, where in the first type of proxy server and the second type of proxy server are non-identical proxy server types;

intercepting the output from the first type of proxy server and the second type of proxy server;

comparing the output of the first type of proxy server to the output of the second type of proxy server;

determining whether the output of the first type of proxy server and the output of the second type of proxy server differ, in response to comparing the outputs;

generating an alert, in response to determining a difference in the outputs; and

further comprising determining that the output from the first type of proxy server and the second type of proxy server do not differ and routing the output from one of the first type of proxy server or the second type of proxy server to the client device.

12. A method for generating an alert in response to a security breach being detected in a proxy sever of a data processing network, the system comprising:

intercepting a requested resource from a server; and

creating a duplicate of the intercepted requested resource and forwarding the intercepted requested resource to a first type of proxy server and forwarding the duplicate of the intercepted requested resource to a second type of proxy server, where in the first type of proxy server and the second type of proxy server are non-identical proxy servers;

intercepting the output from the first type of proxy server and the second type of proxy server;

comparing the output of the first type of proxy server to the output of the second type of proxy server;

determining whether the output of the first type of proxy server and the output of the second type of proxy server differ, in response to comparing the outputs;

generating an alert, in response to determining a difference in the outputs; and

further comprising determining that the output from the first type of proxy server and the second type of proxy server do not differ and routing the output from one of the first type of proxy server or the second type of proxy server to the client device.

13. A computer program comprising computer program code to, when loaded into a computer system and executed, perform all the steps of the method according to any one of claims 11 and 12.