US 20250267159A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2025/0267159 A1**

Joseph et al. (43) **Pub. Date: Aug. 21, 2025**

(54) **ACTIVE DIRECTORY SECURITY ENFORCEMENT AND THREAT INSIGHTS ON ZERO TRUST NETWORKS**

(71) Applicant: **Zscaler, Inc.**, San Jose, CA (US)

(72) Inventors: **Jane Joseph**, Leander, TX (US); **Pankaj Kumar**, Columbus, OH (US); **Abhinav Saund**, Mohali (IN); **Kanti Varanasi**, Sunnyvale, CA (US); **Shyam Pullela**, San Jose, CA (US)

(73) Assignee: **Zscaler, Inc.**, San Jose, CA (US)

(21) Appl. No.: **18/973,641**

(22) Filed: **Dec. 9, 2024**

Related U.S. Application Data

(63) Continuation-in-part of application No. 18/621,258, filed on Mar. 29, 2024.

(30) **Foreign Application Priority Data**

Feb. 15, 2024 (IN) 202441010596

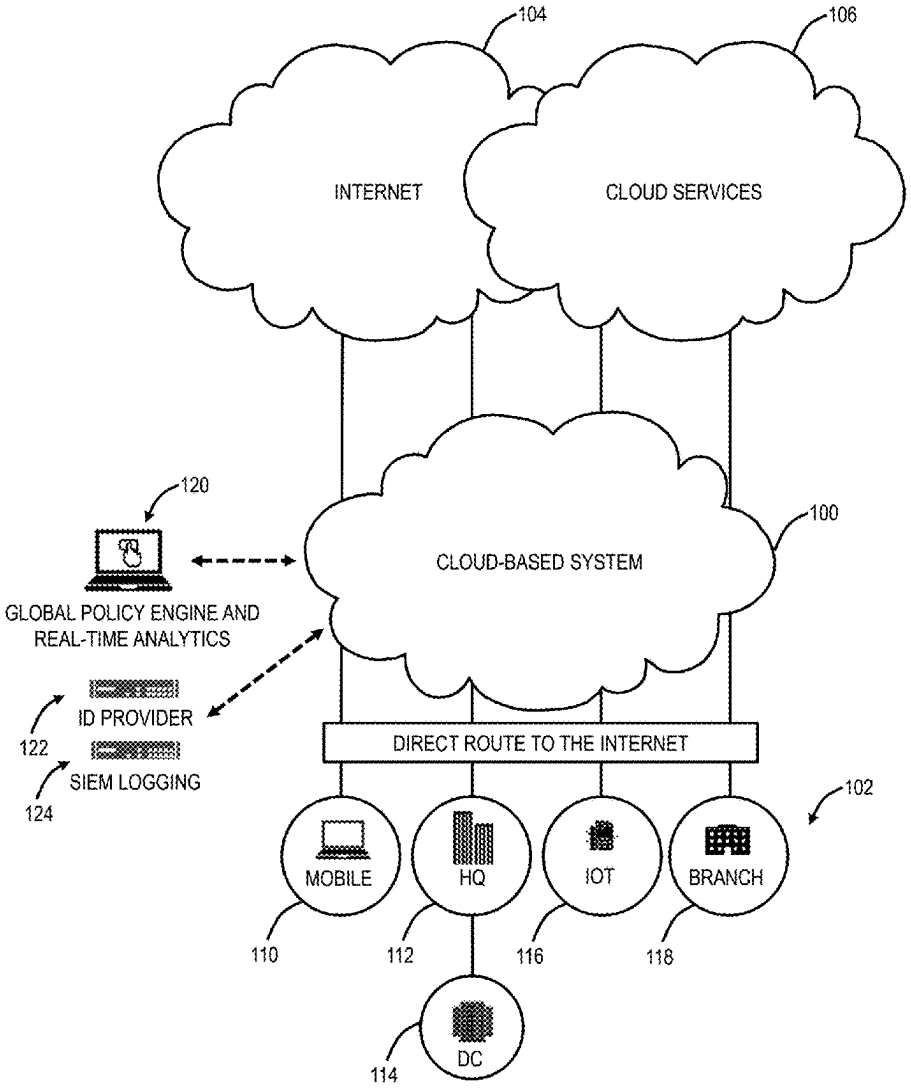
Publication Classification

(51) **Int. Cl.**
H04L 9/40 (2022.01)

(52) **U.S. Cl.**
CPC ***H04L 63/1425*** (2013.01); ***H04L 63/1416*** (2013.01); ***H04L 63/1441*** (2013.01)

(57) **ABSTRACT**

Systems and methods for active directory security enforcement and threat insights on zero trust networks include performing inline monitoring of traffic associated with a plurality of tenants of the cloud-based system; classifying the traffic as being associated with any of one or more active directory protocols; inspecting the traffic associated with the one or more detected active directory protocols; and performing one or more actions on the traffic based on the inspecting.



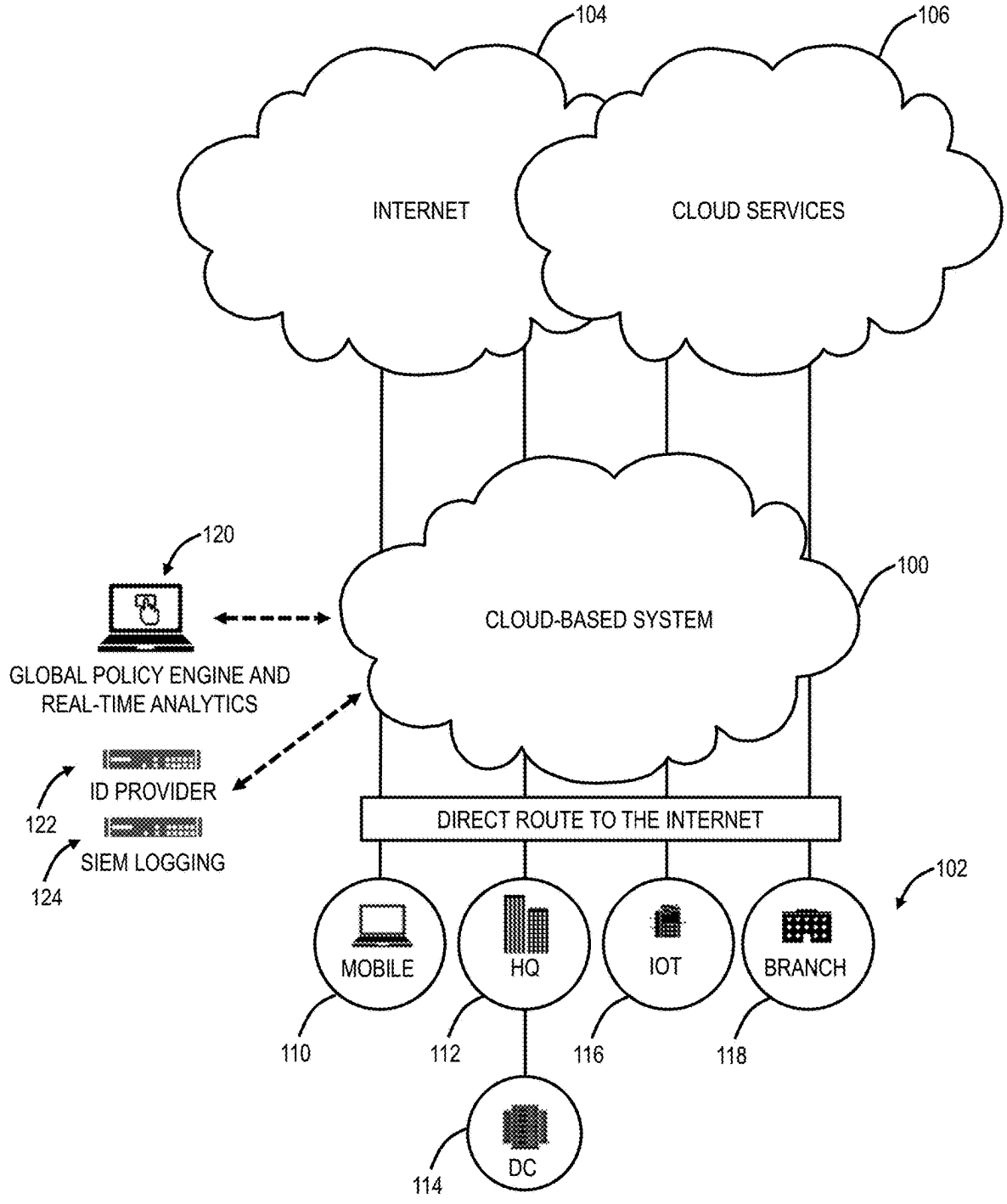

104
106
INTERNET
CLOUD SERVICES
120
100
CLOUD-BASED SYSTEM
GLOBAL POLICY ENGINE AND
REAL-TIME ANALYTICS
ID PROVIDER
122
SIEM LOGGING
124
DIRECT ROUTE TO THE INTERNET
102
MOBILE
HQ
IOT
BRANCH
110
112
116
118
DC
114

FIG. 1A

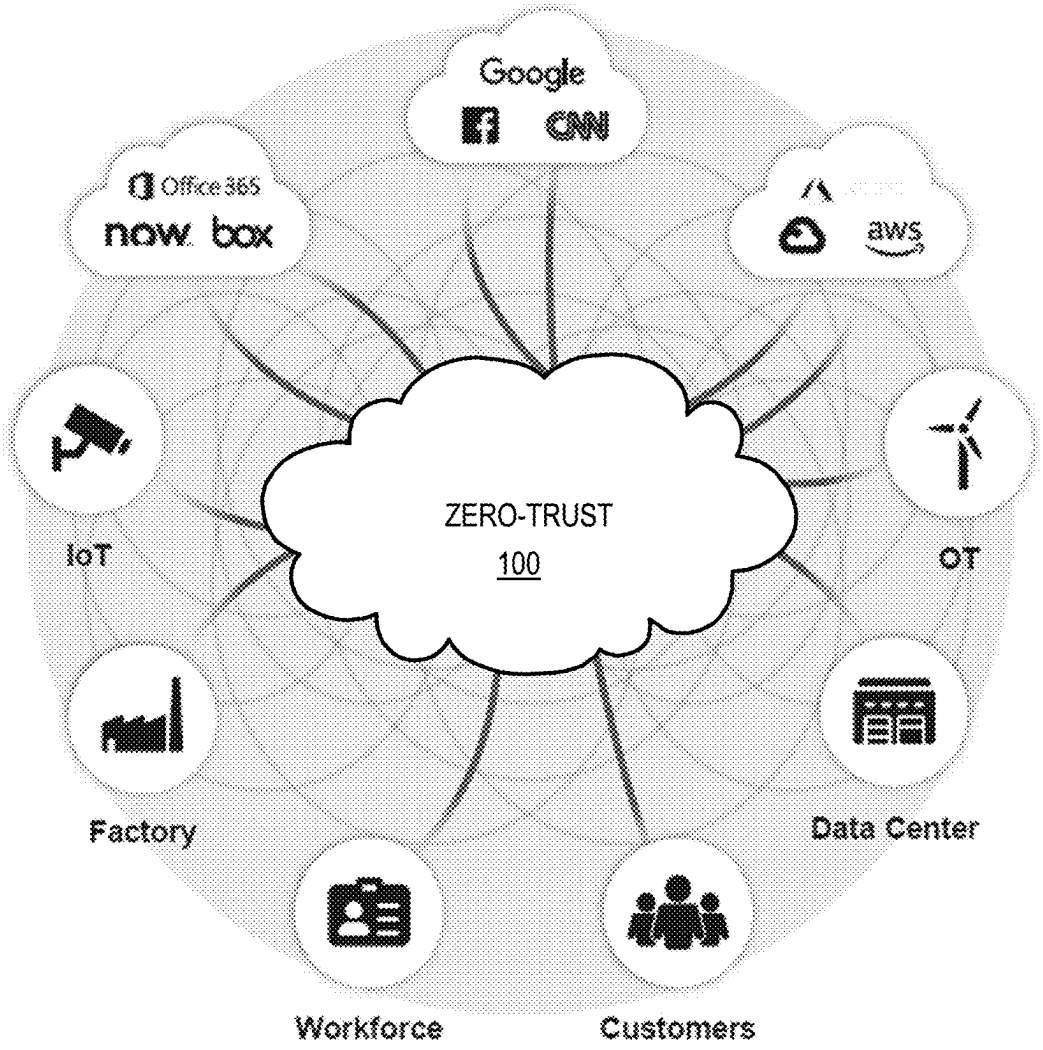

Google
CNN
Office 365
now
box
aws
ZERO-TRUST
100
IoT
OT
Factory
Data Center
Workforce
Customers

FIG. 1B

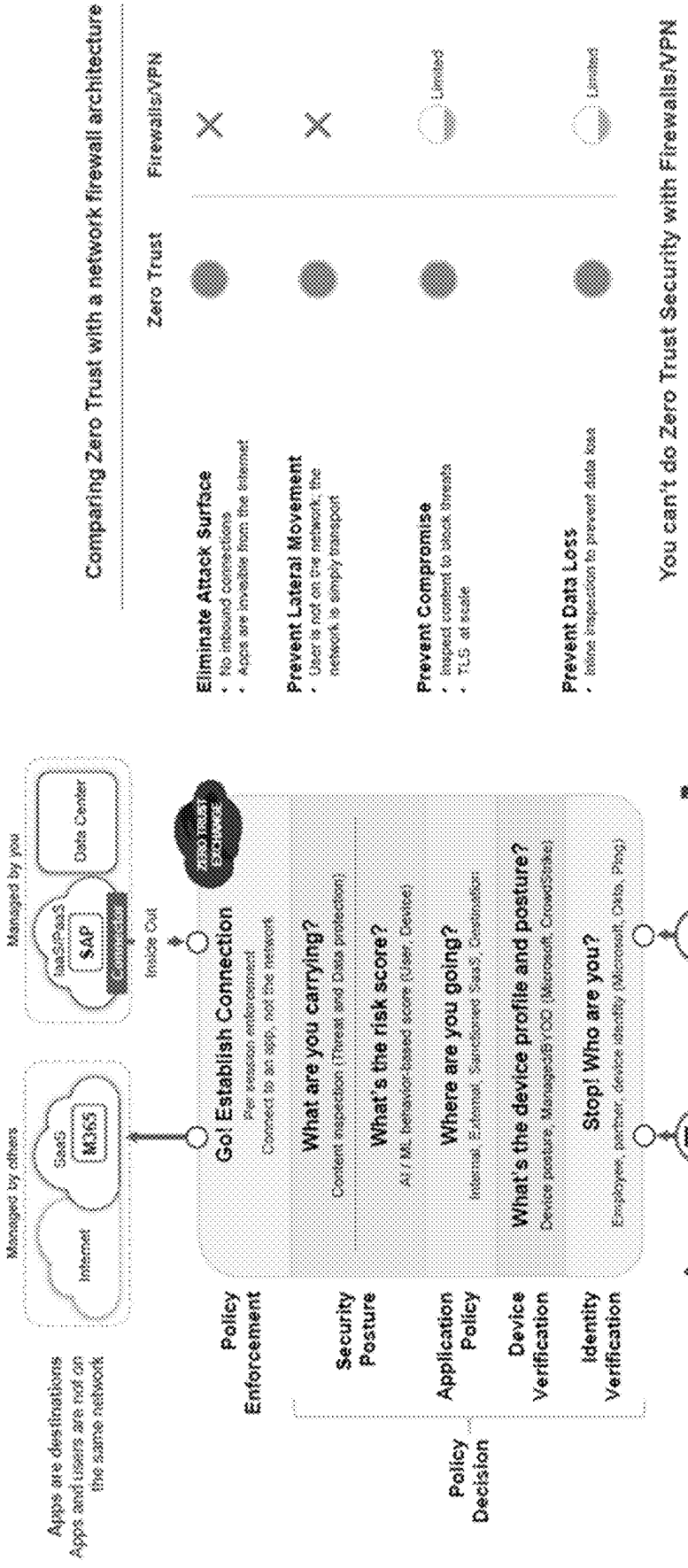

Comparing Zero Trust with a network firewall architecture
Zero Trust
Firewalls/VPN
Eliminate Attack Surface
No inbound connections
Apps are invisible from the internet
Prevent Lateral Movement
User is not on the network, the network is simply transport
Prevent Compromise
Inspect content to block threats
TLS at scale
Limited
Prevent Data Loss
Allow inspection to prevent data loss
Limited
You can't do Zero Trust Security with Firewalls/VPN
Managed by others
Internet
SaaS
M365
Managed by you
IaaS/PaaS
SAP
Data Center
Apps are destinations
Apps and users are not on the same network
Inside Out
Go! Establish Connection
Per-session enforcement
Connect to an app, not the network
Policy Enforcement
What are you carrying?
Content inspection (Threat and Data protection)
What's the risk score?
AI / ML behavior-based score (User, Device)
Where are you going?
Internet, External, Sanctioned SaaS, Destination
What's the device profile and posture?
Device posture: Managed or not (Microsoft Corestack)
Stop! Who are you?
Employee, partner, device identity (Microsoft, Okta, Ping)
Security Posture
Application Policy
Device Verification
Identity Verification
Policy Decision
Request origination
Any User, Any Device,
Any App, Any Location
IoT/OT
Mobile
Laptop
Apps

FIG. 1C

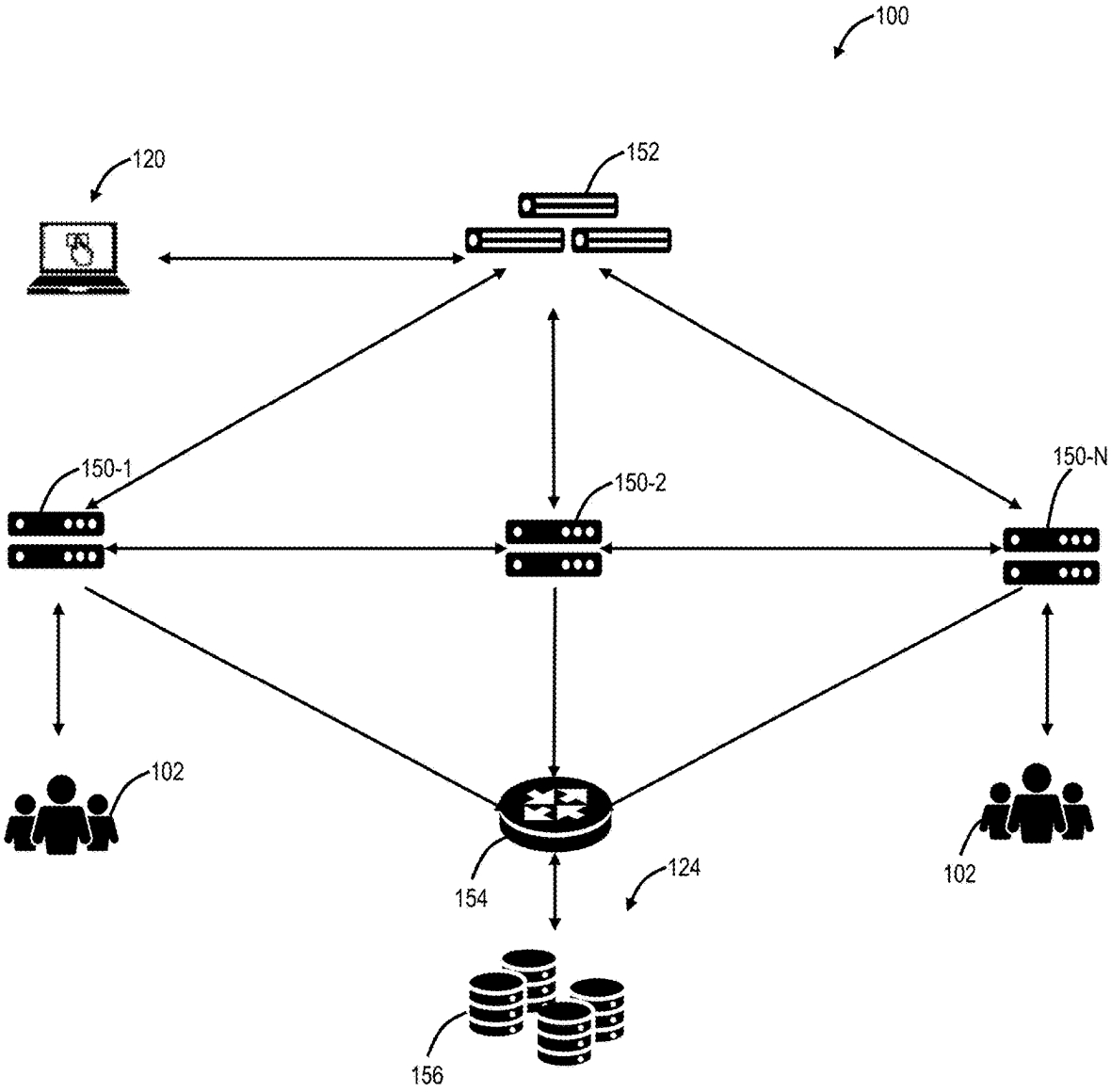


FIG. 2

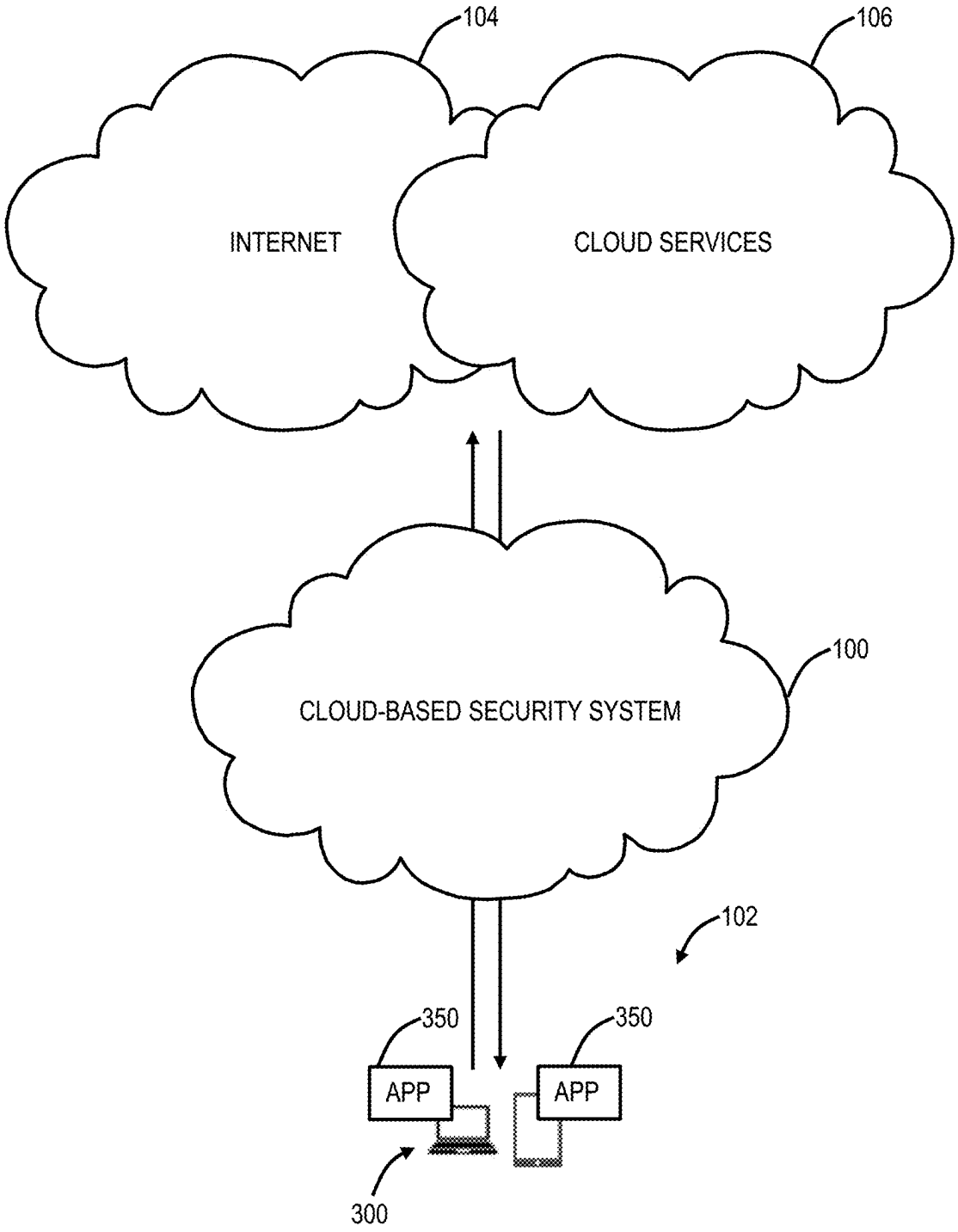

104
106
INTERNET
CLOUD SERVICES
CLOUD-BASED SECURITY SYSTEM
100
102
350
350
APP
APP
300

FIG. 3

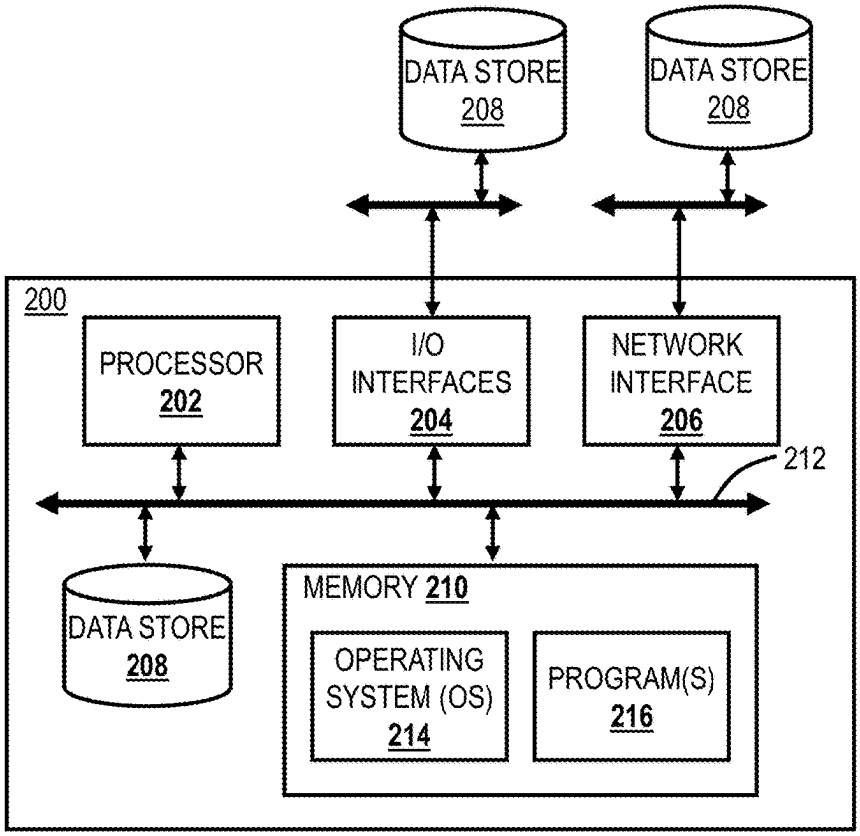

DATA STORE
208
DATA STORE
208
200
PROCESSOR
202
I/O
INTERFACES
204
NETWORK
INTERFACE
206
212
DATA STORE
208
MEMORY 210
OPERATING
SYSTEM (OS)
214
PROGRAM(S)
216

FIG. 4

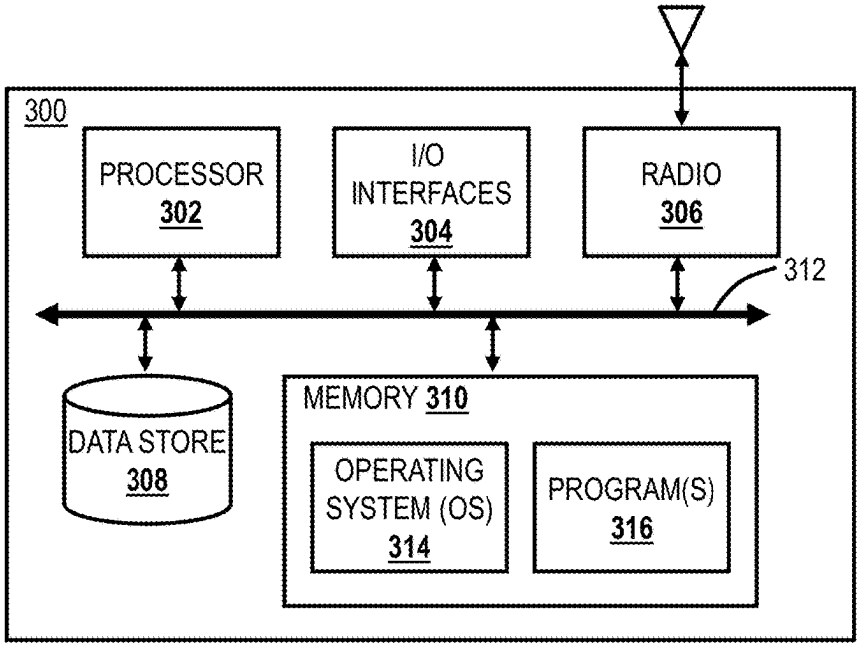

300
PROCESSOR
302
I/O
INTERFACES
304
RADIO
306
312
DATA STORE
308
MEMORY 310
OPERATING
SYSTEM (OS)
314
PROGRAM(S)
316

FIG. 5

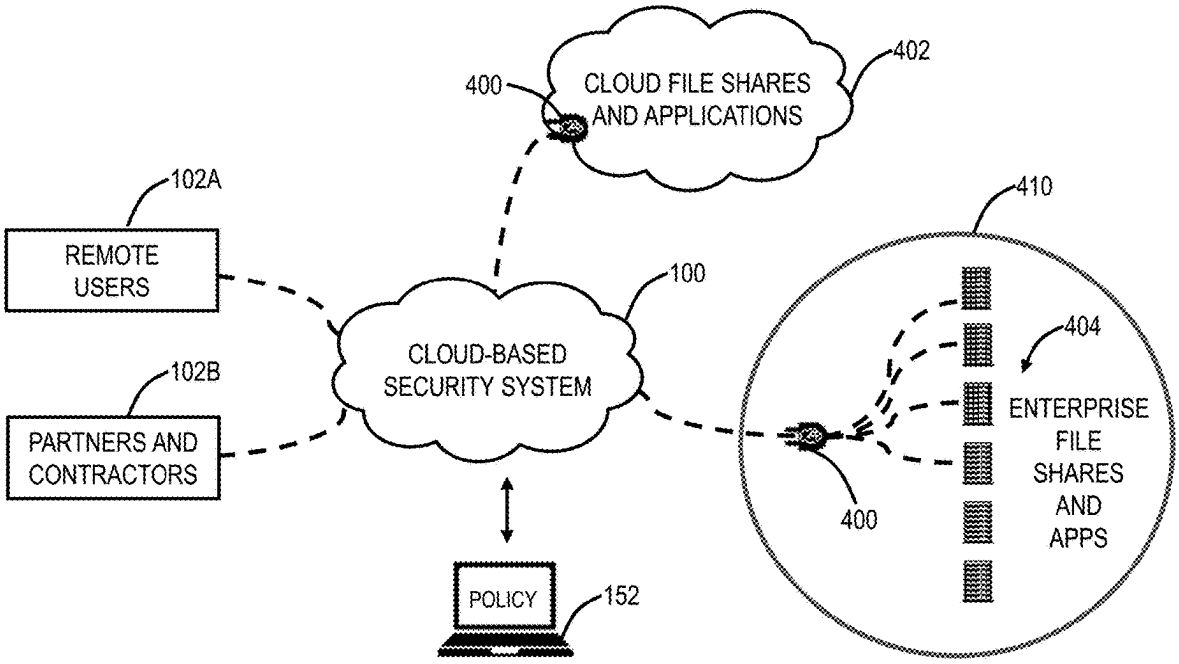

400
CLOUD FILE SHARES AND APPLICATIONS
402
102A
REMOTE USERS
410
100
CLOUD-BASED SECURITY SYSTEM
404
102B
PARTNERS AND CONTRACTORS
ENTERPRISE FILE SHARES AND APPS
400
POLICY
152

FIG. 6

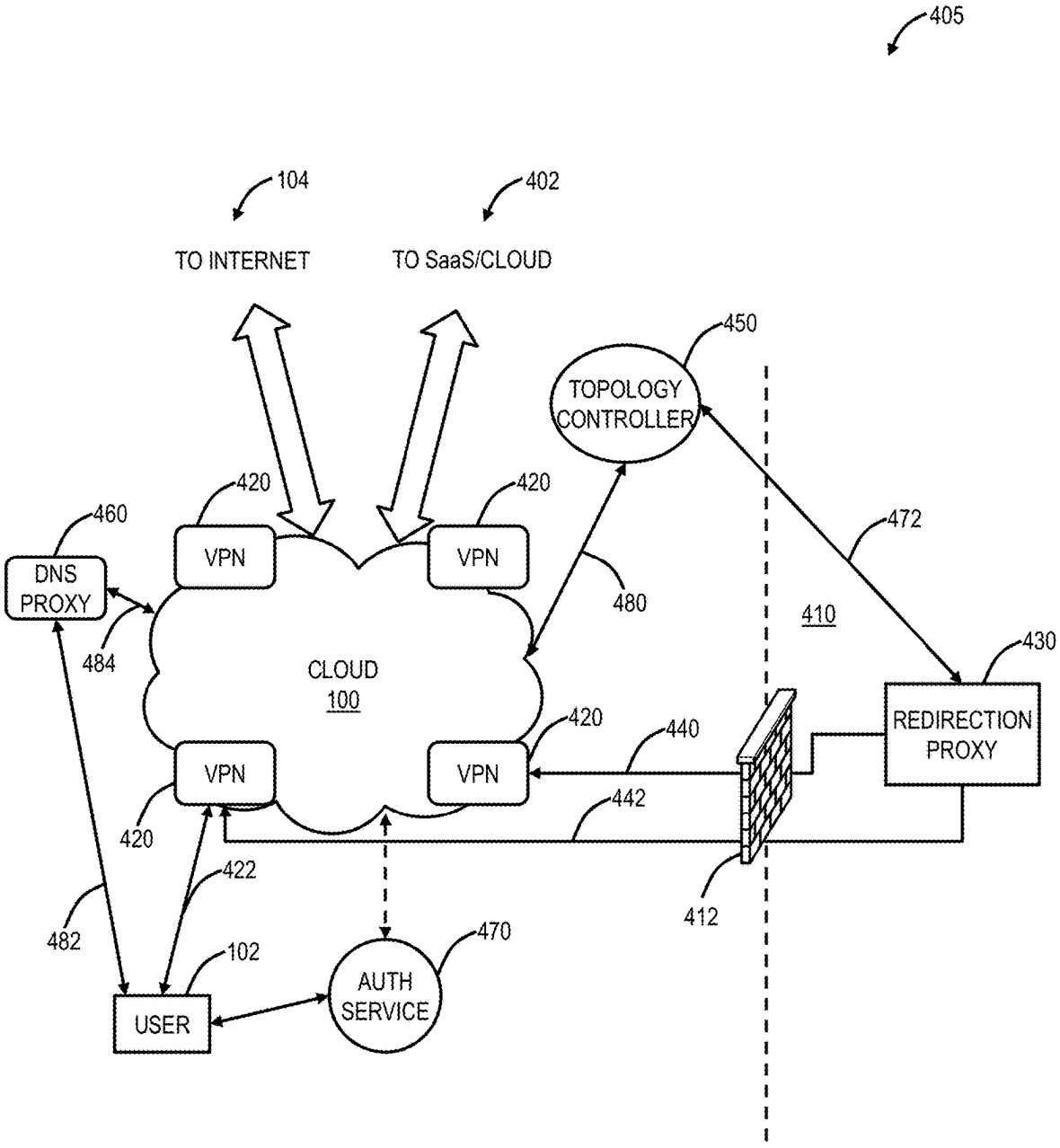

405
104
TO INTERNET
402
TO SaaS/CLOUD
450
TOPOLOGY CONTROLLER
420
420
460
VPN
VPN
DNS PROXY
472
480
484
CLOUD
100
410
430
REDIRECTION PROXY
420
440
VPN
VPN
420
442
422
412
482
102
470
AUTH SERVICE
USER

FIG. 7

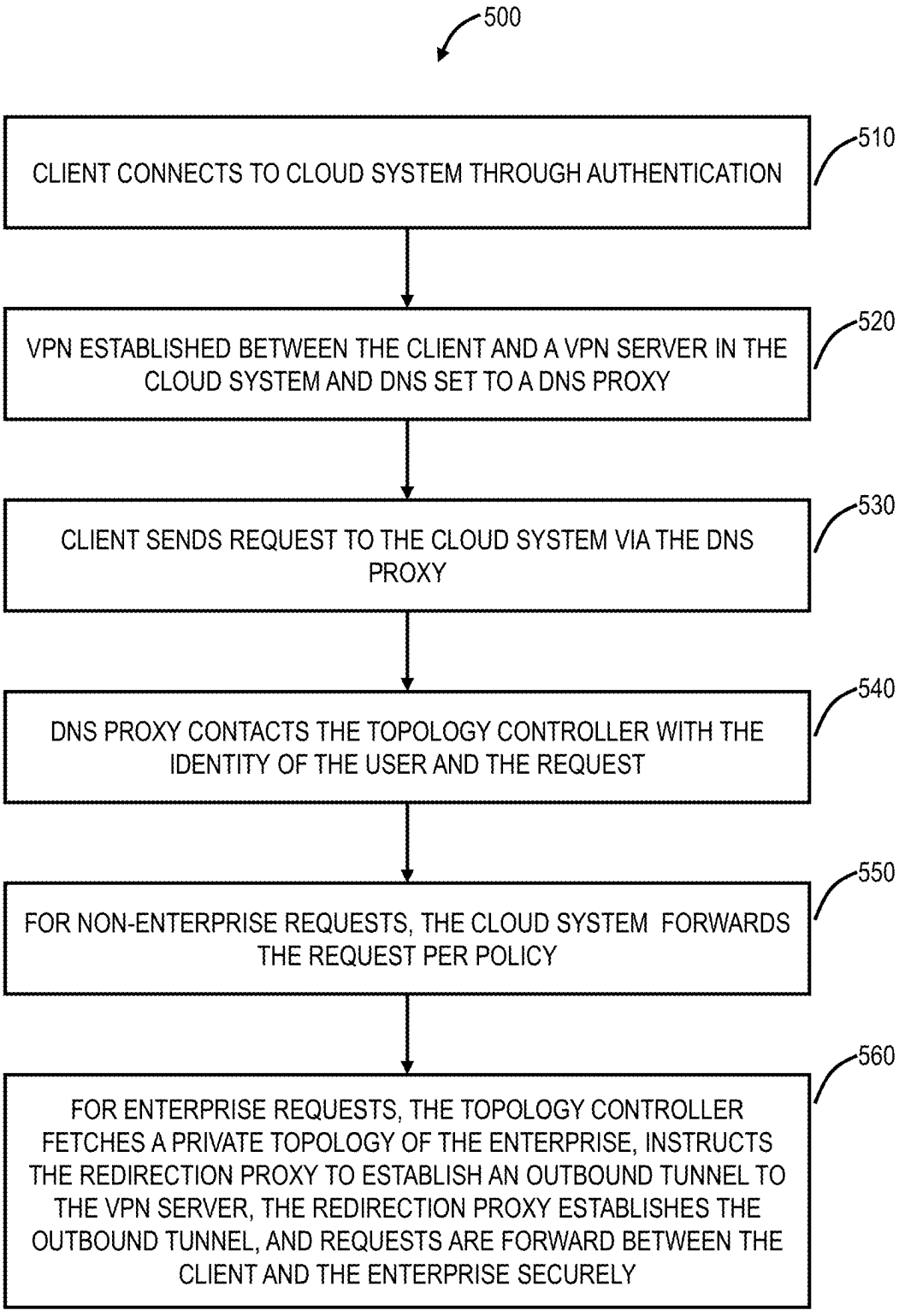
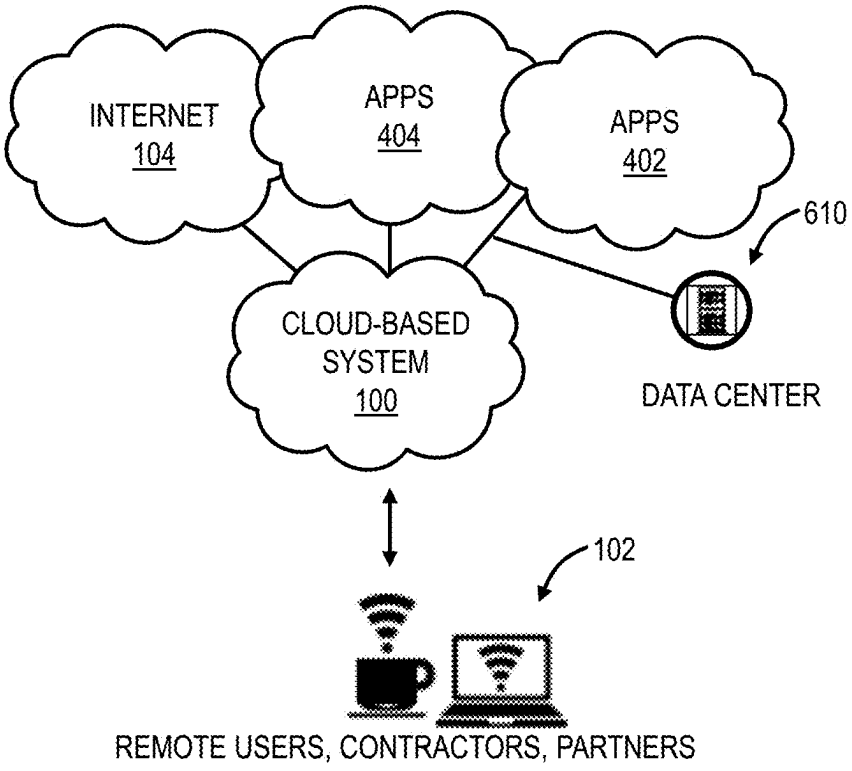
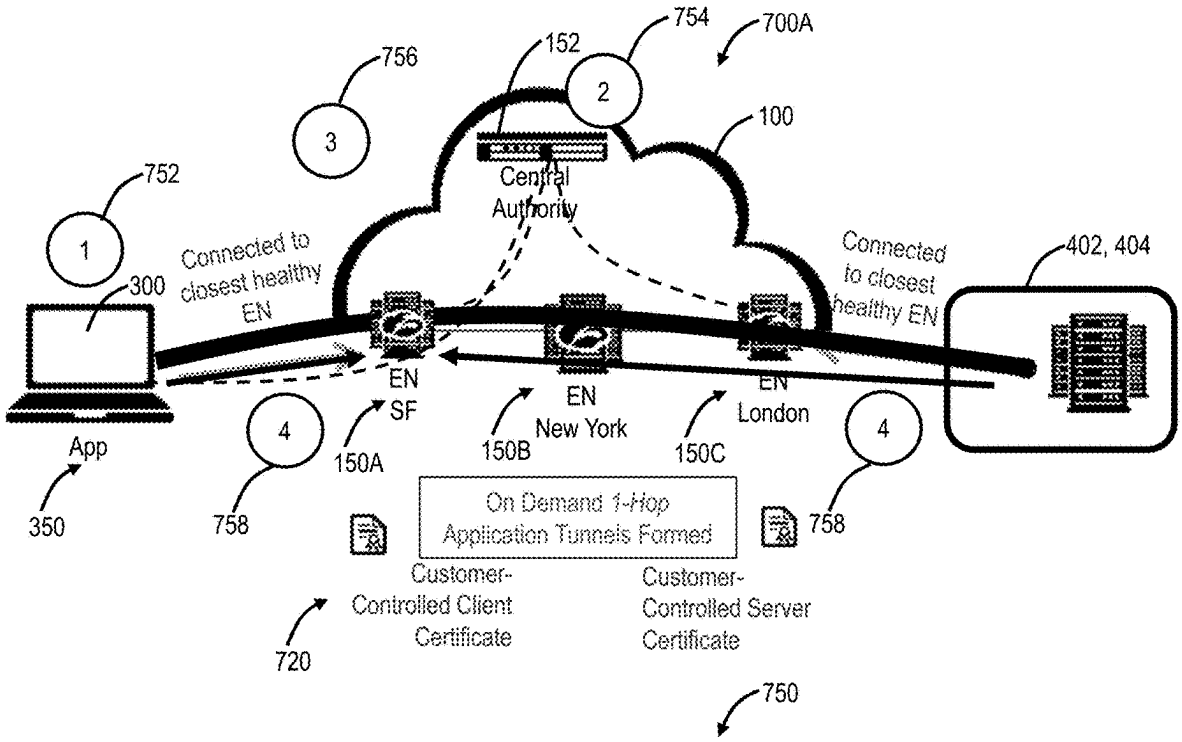

500
CLIENT CONNECTS TO CLOUD SYSTEM THROUGH AUTHENTICATION
510
VPN ESTABLISHED BETWEEN THE CLIENT AND A VPN SERVER IN THE CLOUD SYSTEM AND DNS SET TO A DNS PROXY
520
CLIENT SENDS REQUEST TO THE CLOUD SYSTEM VIA THE DNS PROXY
530
DNS PROXY CONTACTS THE TOPOLOGY CONTROLLER WITH THE IDENTITY OF THE USER AND THE REQUEST
540
FOR NON-ENTERPRISE REQUESTS, THE CLOUD SYSTEM FORWARDS THE REQUEST PER POLICY
550
FOR ENTERPRISE REQUESTS, THE TOPOLOGY CONTROLLER FETCHES A PRIVATE TOPOLOGY OF THE ENTERPRISE, INSTRUCTS THE REDIRECTION PROXY TO ESTABLISH AN OUTBOUND TUNNEL TO THE VPN SERVER, THE REDIRECTION PROXY ESTABLISHES THE OUTBOUND TUNNEL, AND REQUESTS ARE FORWARD BETWEEN THE CLIENT AND THE ENTERPRISE SECURELY
560

FIG. 8

FIG. 9


INTERNET
104
APPS
404
APPS
402
610
CLOUD-BASED
SYSTEM
100
DATA CENTER
102
REMOTE USERS, CONTRACTORS, PARTNERS


754
700A
152
756
2
3
100
752
Central Authority
1
Connected to closest healthy EN
Connected to closest healthy EN
402, 404
300
EN SF
EN New York
EN London
4
4
App
150A
150B
150C
350
758
On Demand 1-Hop Application Tunnels Formed
758
Customer-Controlled Client Certificate
Customer-Controlled Server Certificate
720
750

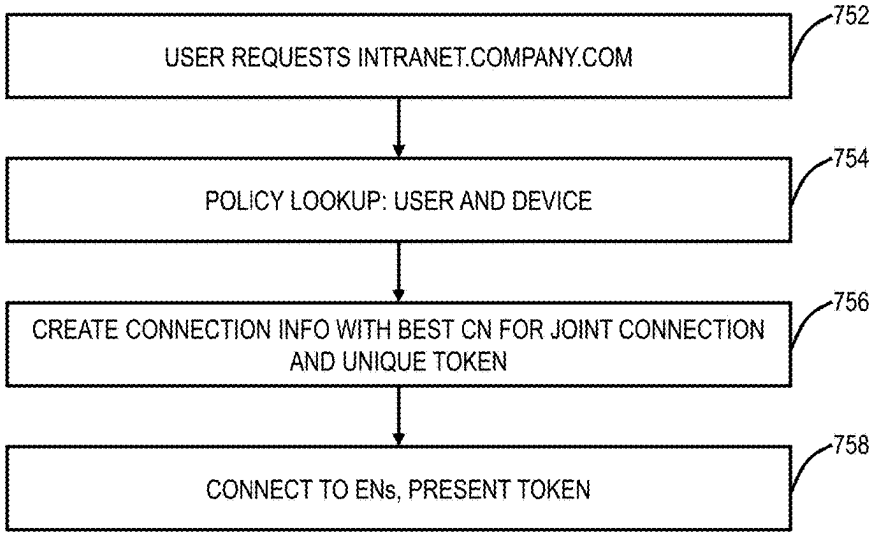

USER REQUESTS INTRANET.COMPANY.COM
752
POLICY LOOKUP: USER AND DEVICE
754
CREATE CONNECTION INFO WITH BEST CN FOR JOINT CONNECTION AND UNIQUE TOKEN
756
CONNECT TO ENs, PRESENT TOKEN
758

FIG. 10

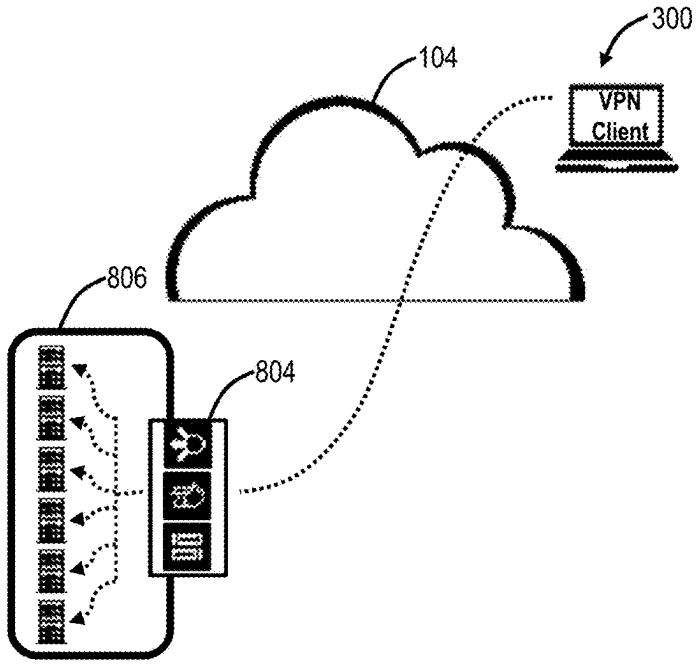


FIG. 11
(Prior Art)

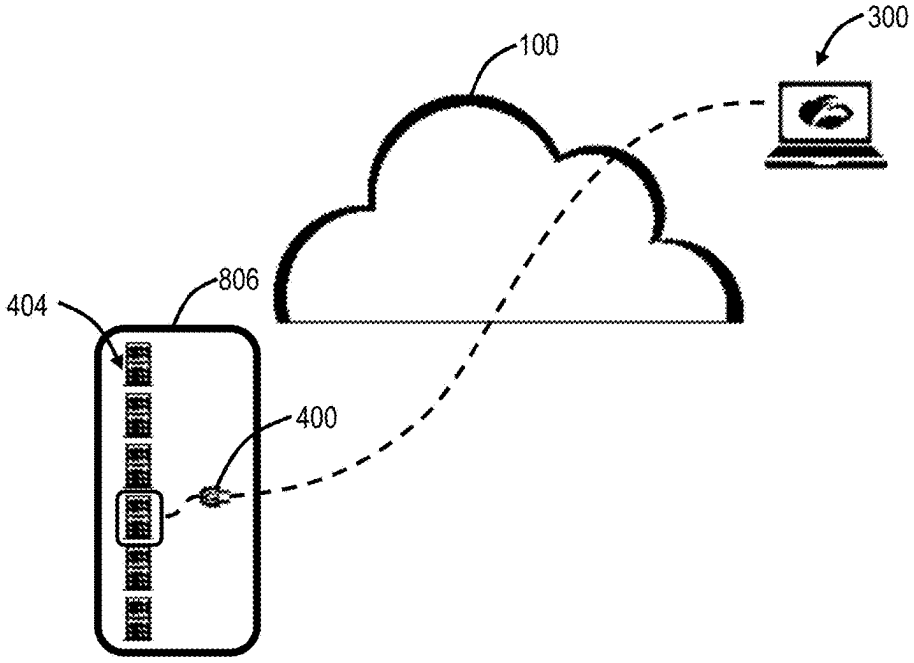


FIG. 12

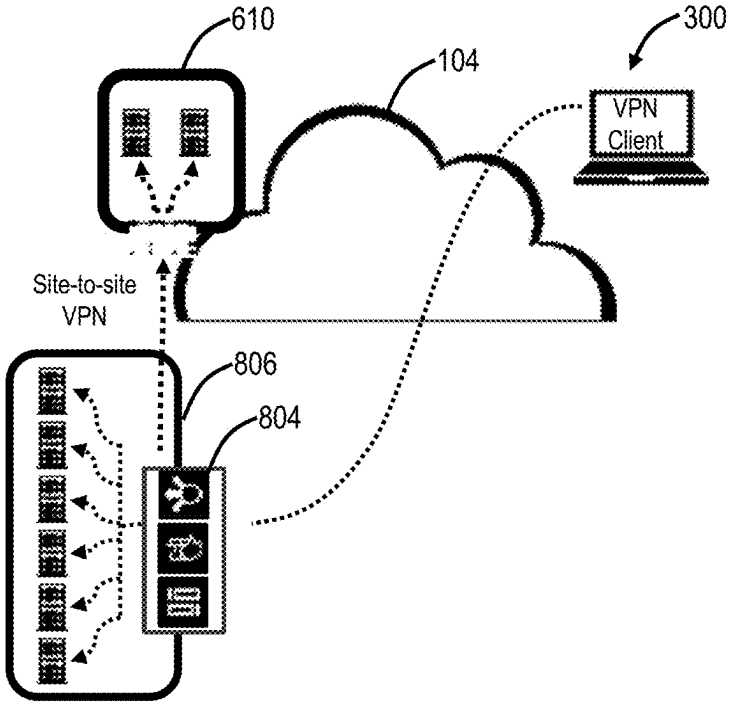


FIG. 13
(Prior Art)

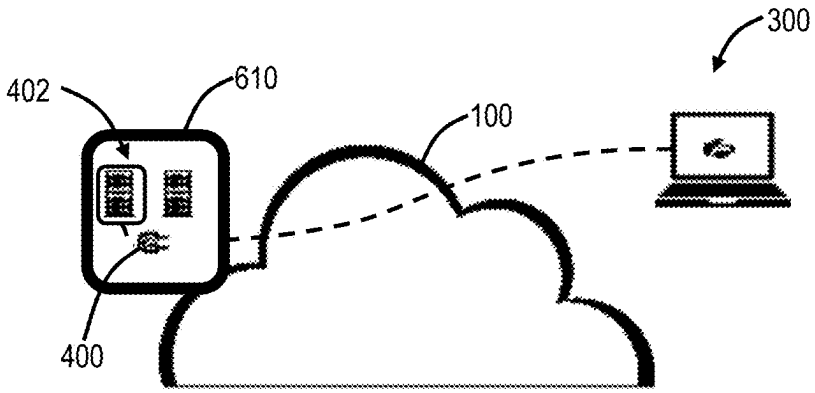


FIG. 14

FIG. 15
(Prior Art)

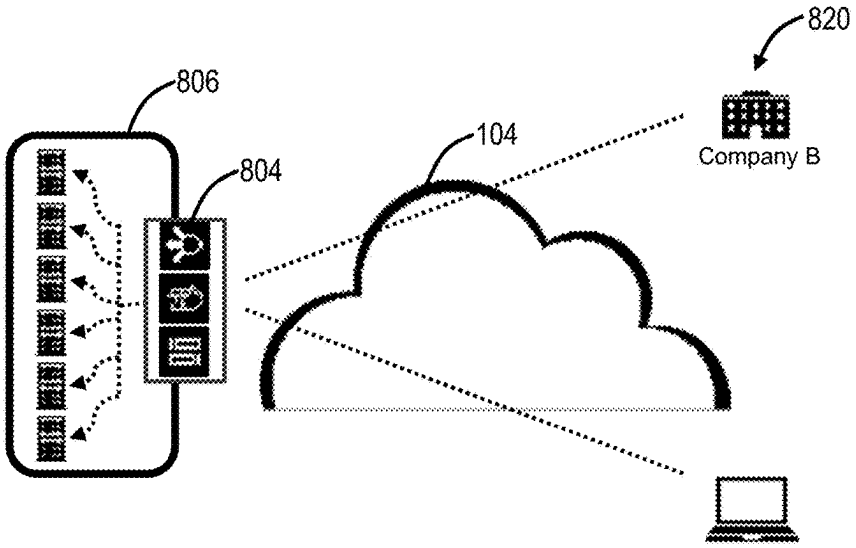


FIG. 16

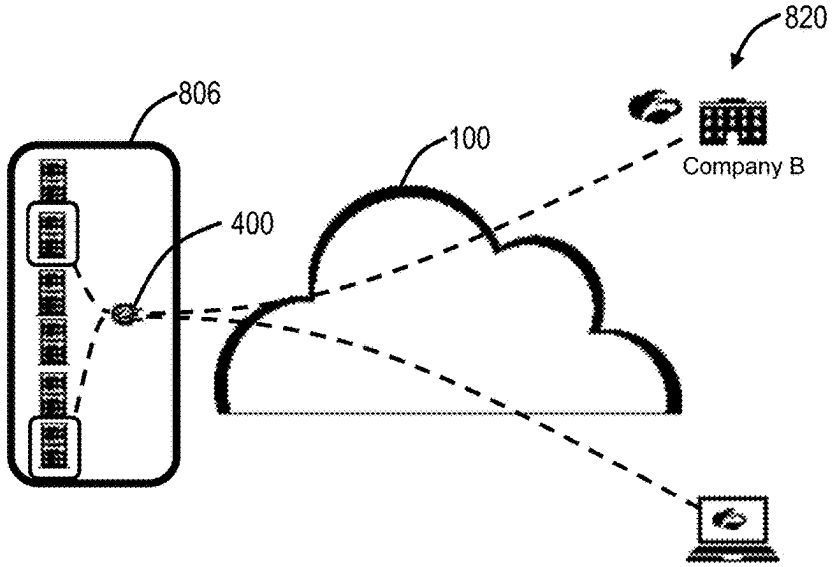


FIG. 17
(Prior Art)

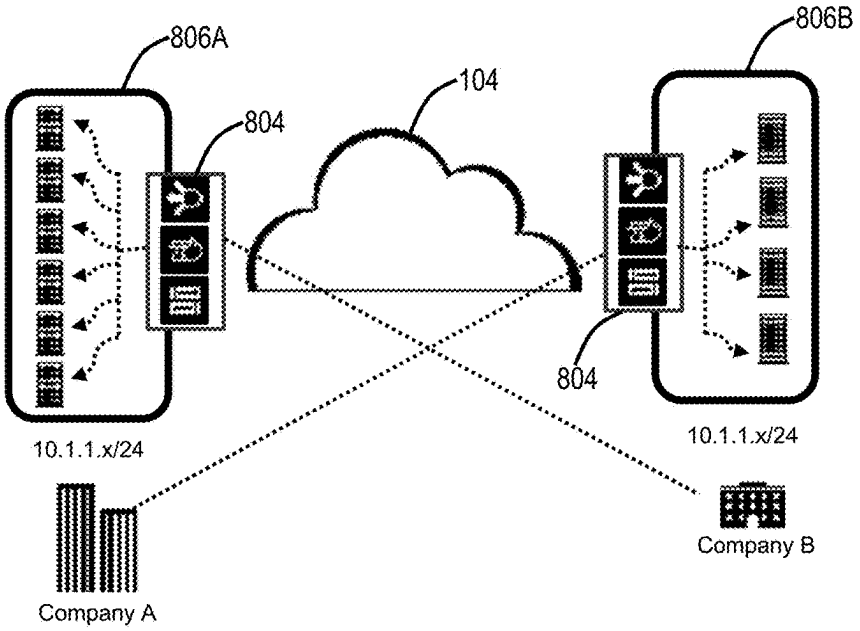
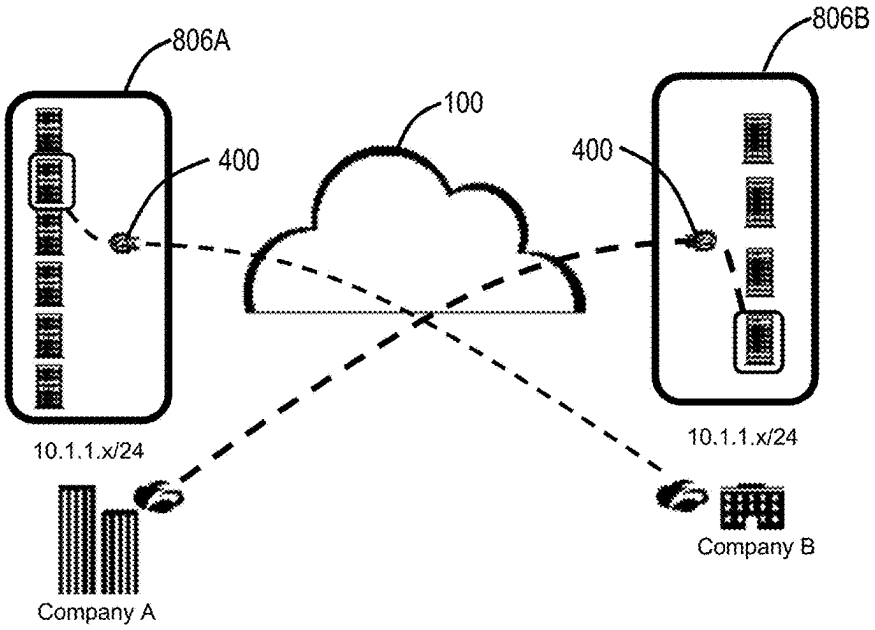


FIG. 18



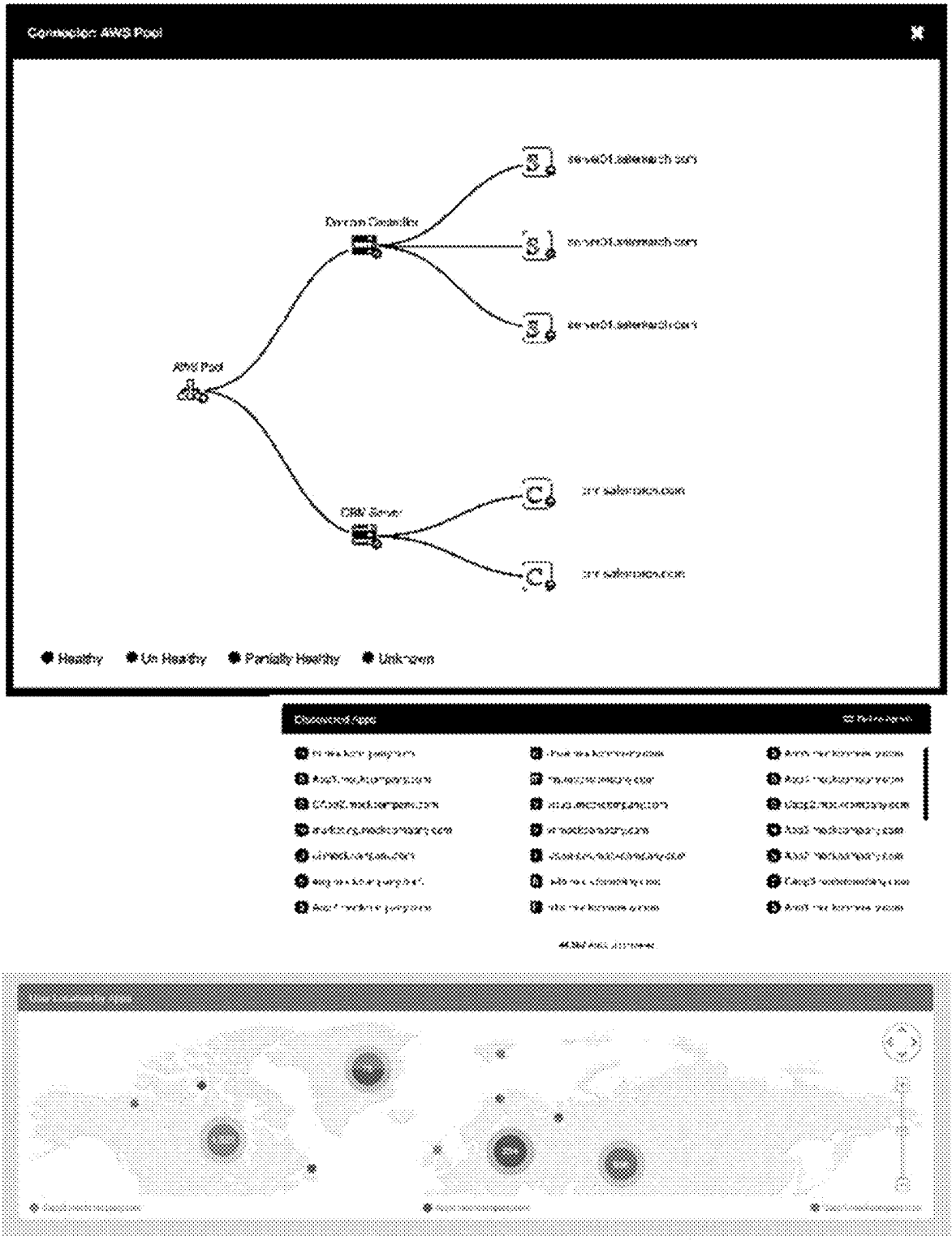


FIG. 19

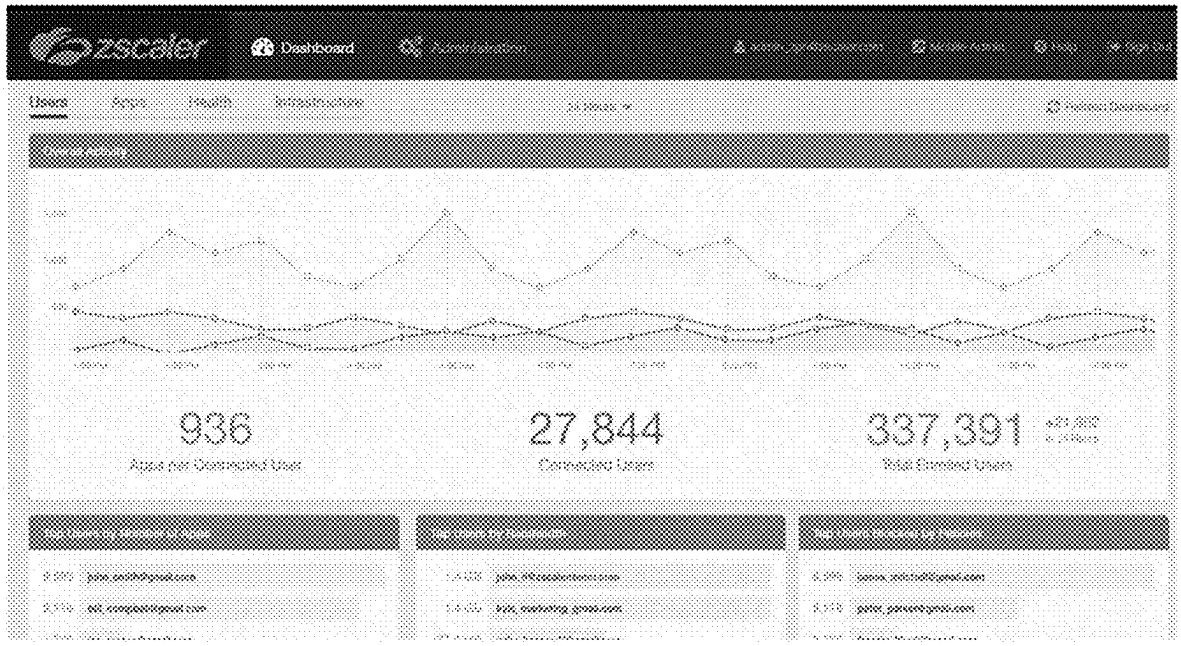

zscaler
Dashboard
Users
Apps
Health
Infrastructure
936
27,844
337,391

FIG. 20

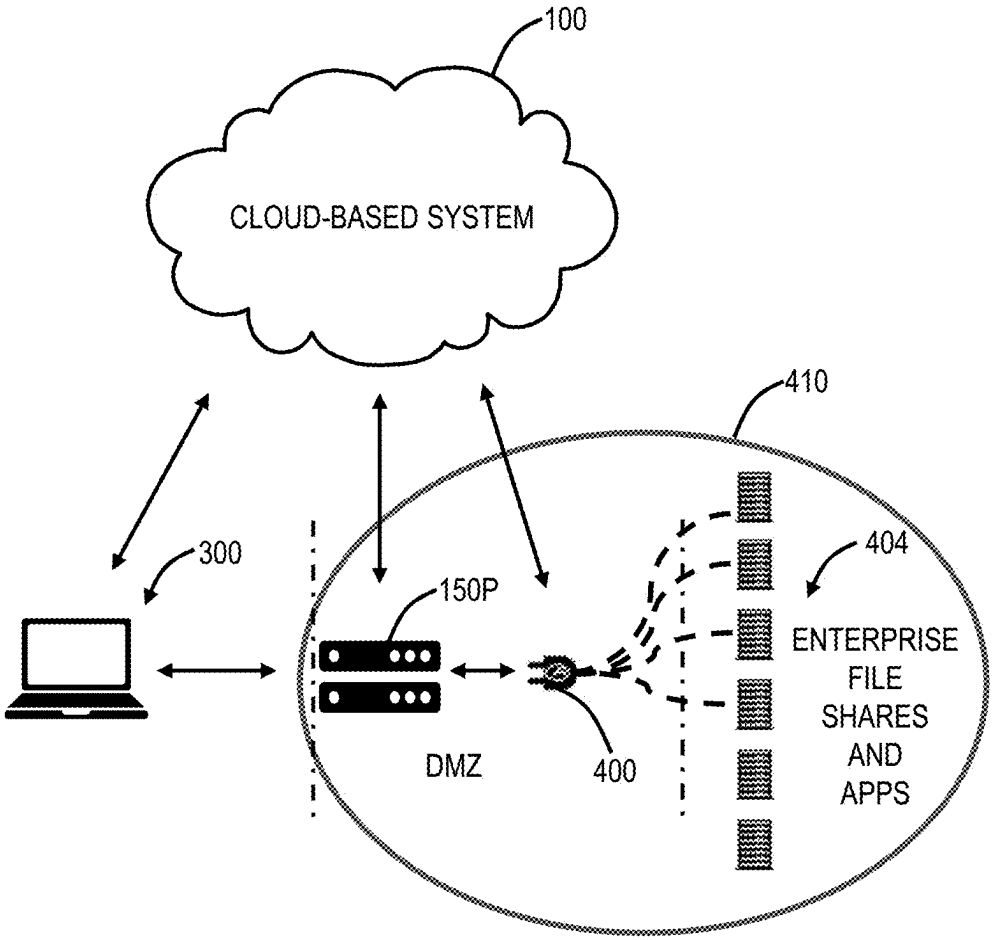

100
CLOUD-BASED SYSTEM
410
300
150P
404
ENTERPRISE
FILE
SHARES
AND
APPS
DMZ
400

FIG. 21

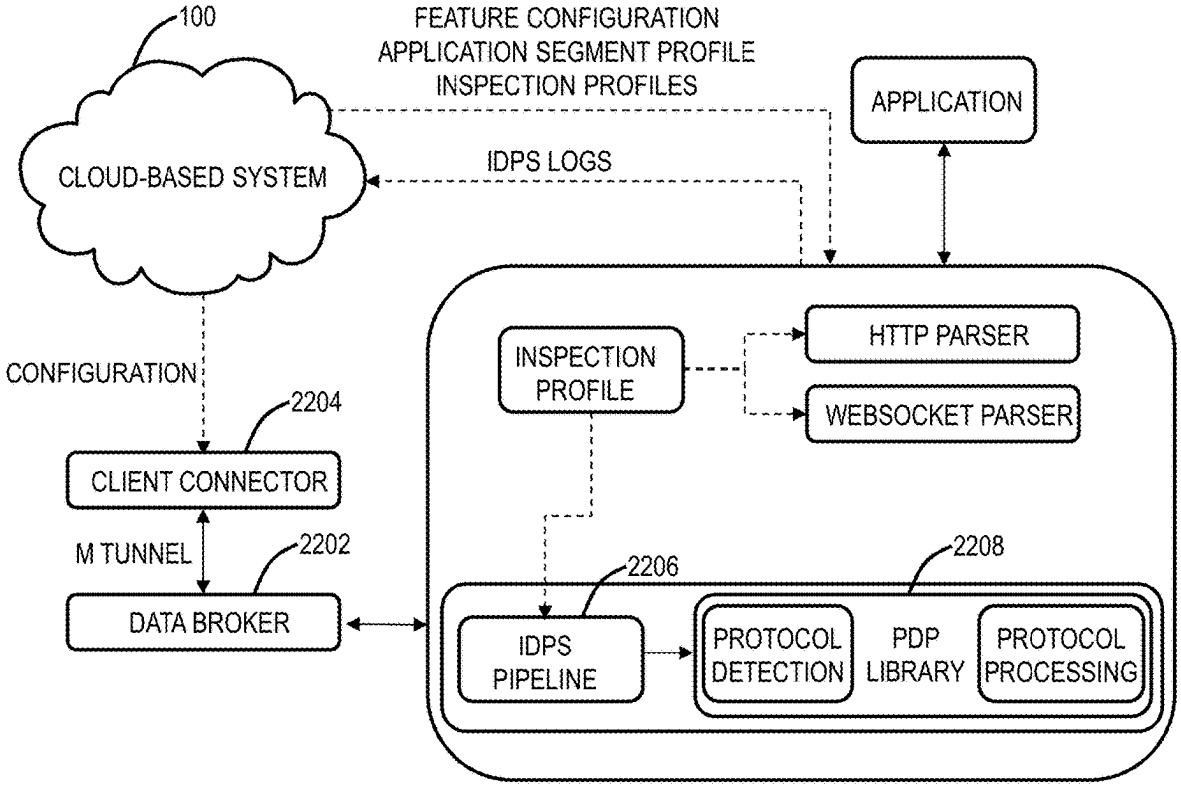

100
FEATURE CONFIGURATION
APPLICATION SEGMENT PROFILE
INSPECTION PROFILES
APPLICATION
CLOUD-BASED SYSTEM
IDPS LOGS
CONFIGURATION
2204
CLIENT CONNECTOR
M TUNNEL
2202
DATA BROKER
INSPECTION PROFILE
HTTP PARSER
WEBSOCKET PARSER
2206
2208
IDPS PIPELINE
PROTOCOL DETECTION
PDP LIBRARY
PROTOCOL PROCESSING

FIG. 22

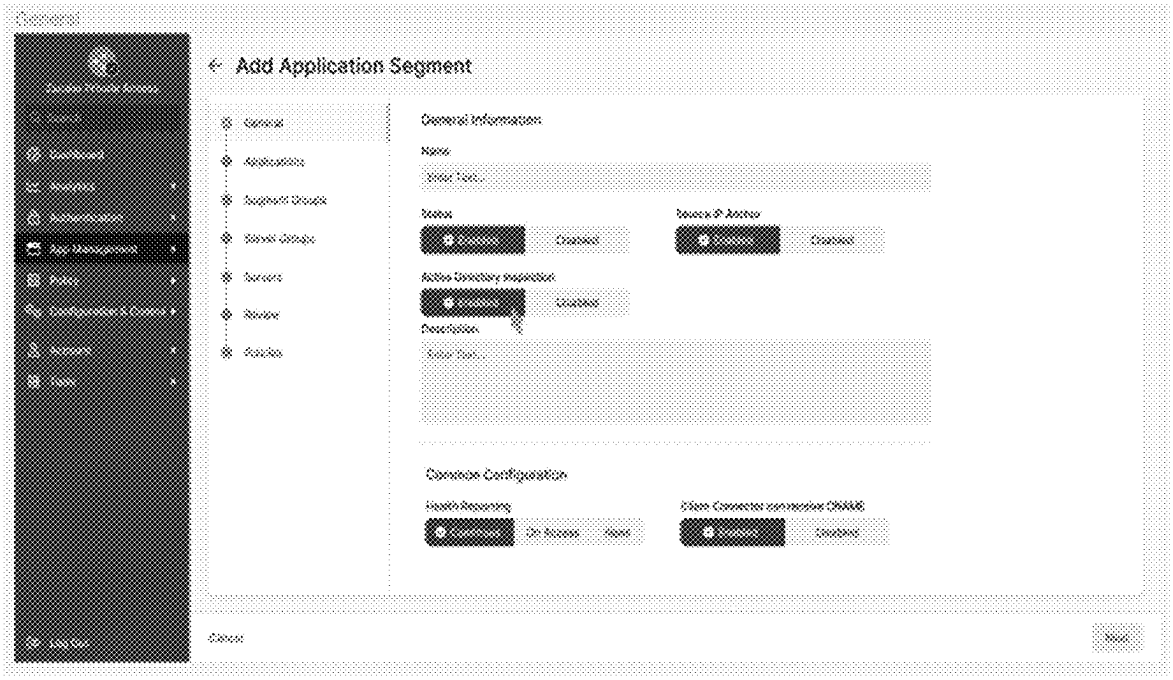

Add Application Segment
General Information
Name
Status
Source IP Anchor
Active Directory Acquisition
Description
Connector Configuration
Cancel
Next

FIG. 23

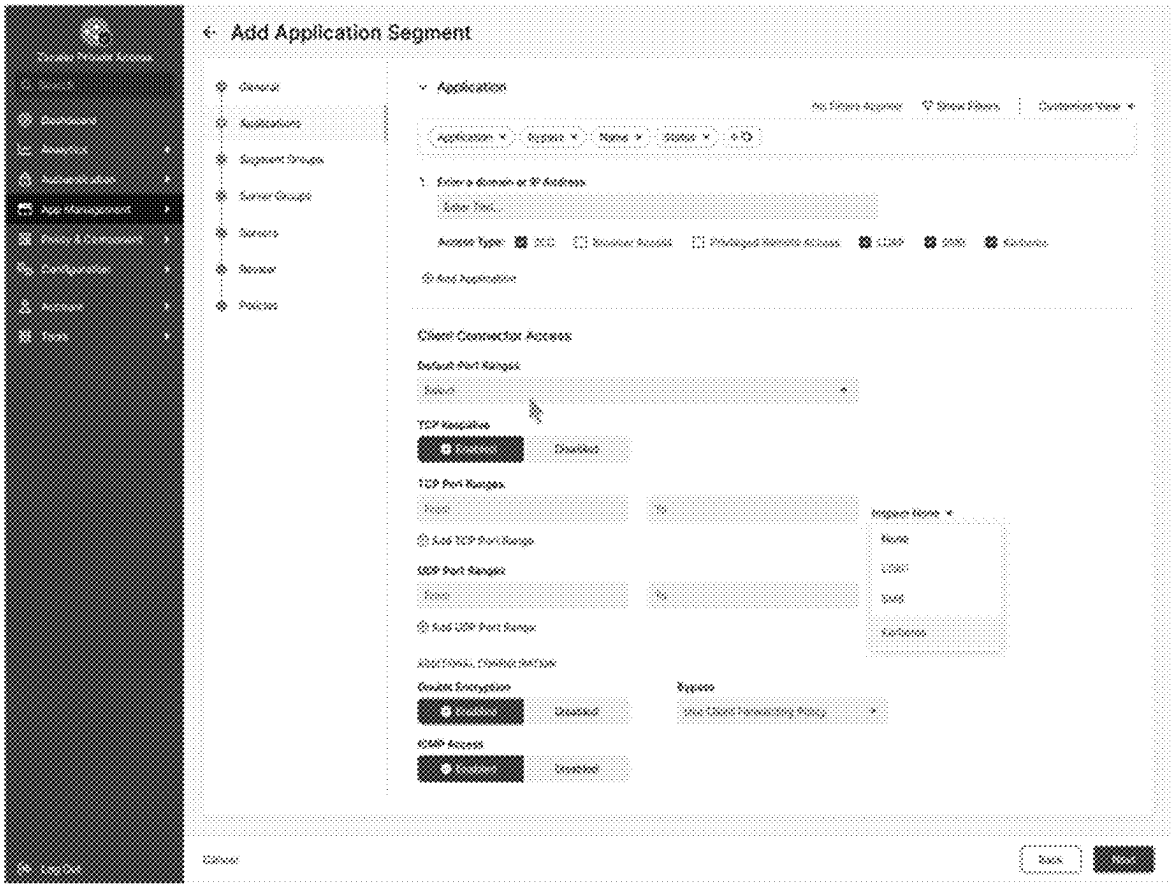


FIG. 24

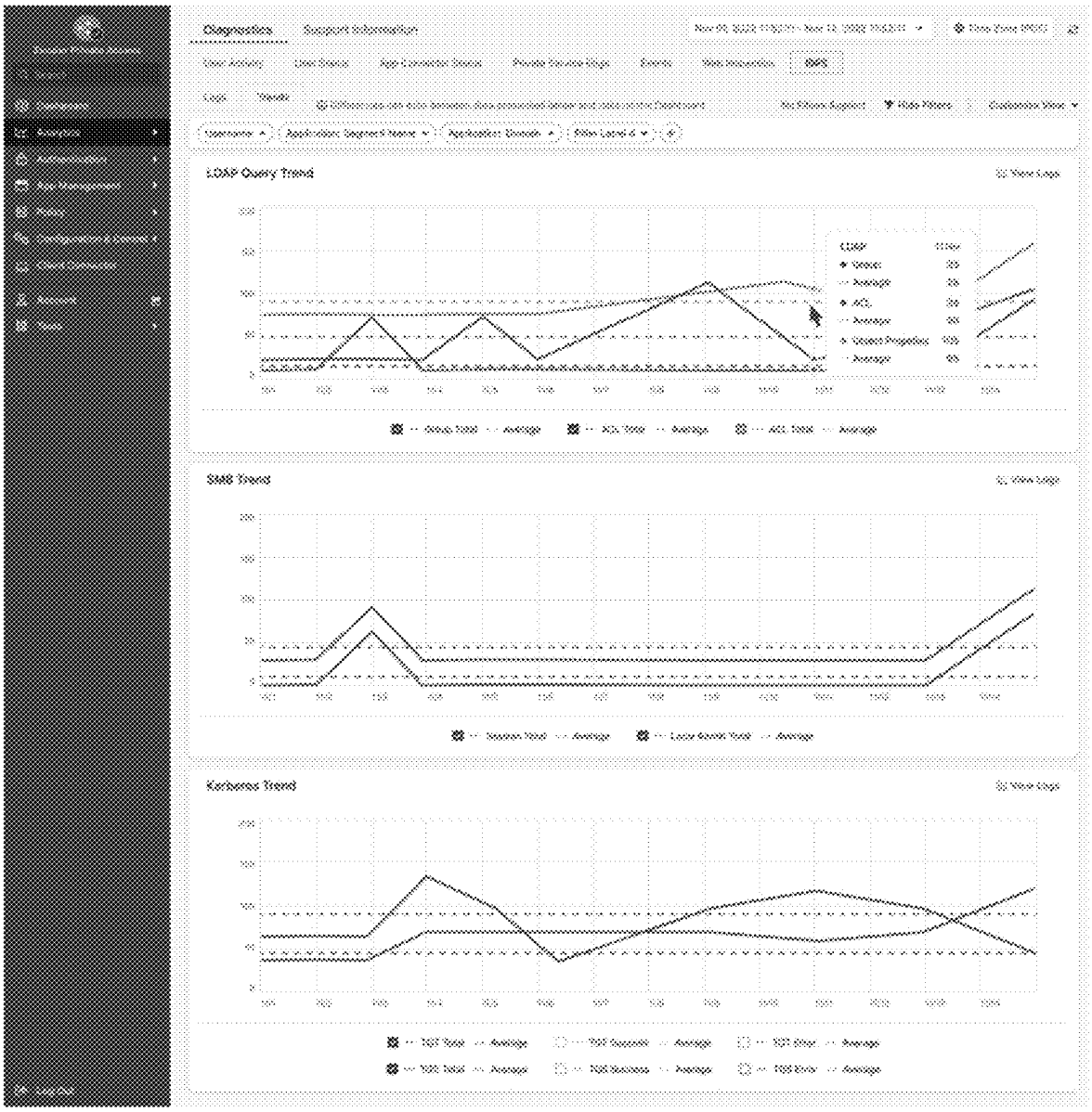


FIG. 25

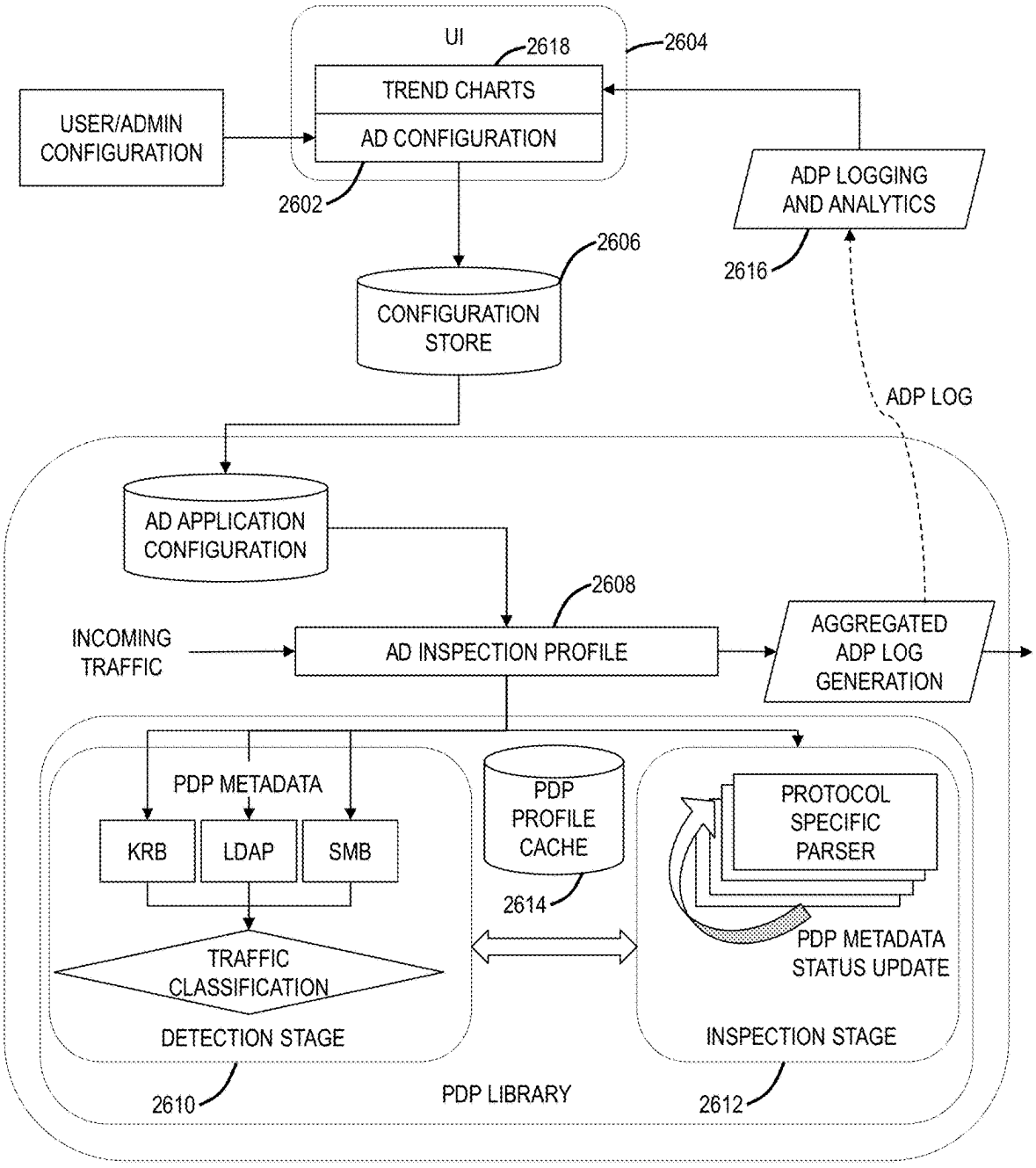

UI
2618
2604
TREND CHARTS
AD CONFIGURATION
USER/ADMIN CONFIGURATION
2602
ADP LOGGING AND ANALYTICS
2616
2606
CONFIGURATION STORE
ADP LOG
AD APPLICATION CONFIGURATION
2608
INCOMING TRAFFIC
AD INSPECTION PROFILE
AGGREGATED ADP LOG GENERATION
PDP METADATA
KRB
LDAP
SMB
TRAFFIC CLASSIFICATION
DETECTION STAGE
PDP PROFILE CACHE
2614
PROTOCOL SPECIFIC PARSER
PDP METADATA STATUS UPDATE
INSPECTION STAGE
2610
PDP LIBRARY
2612

FIG. 26

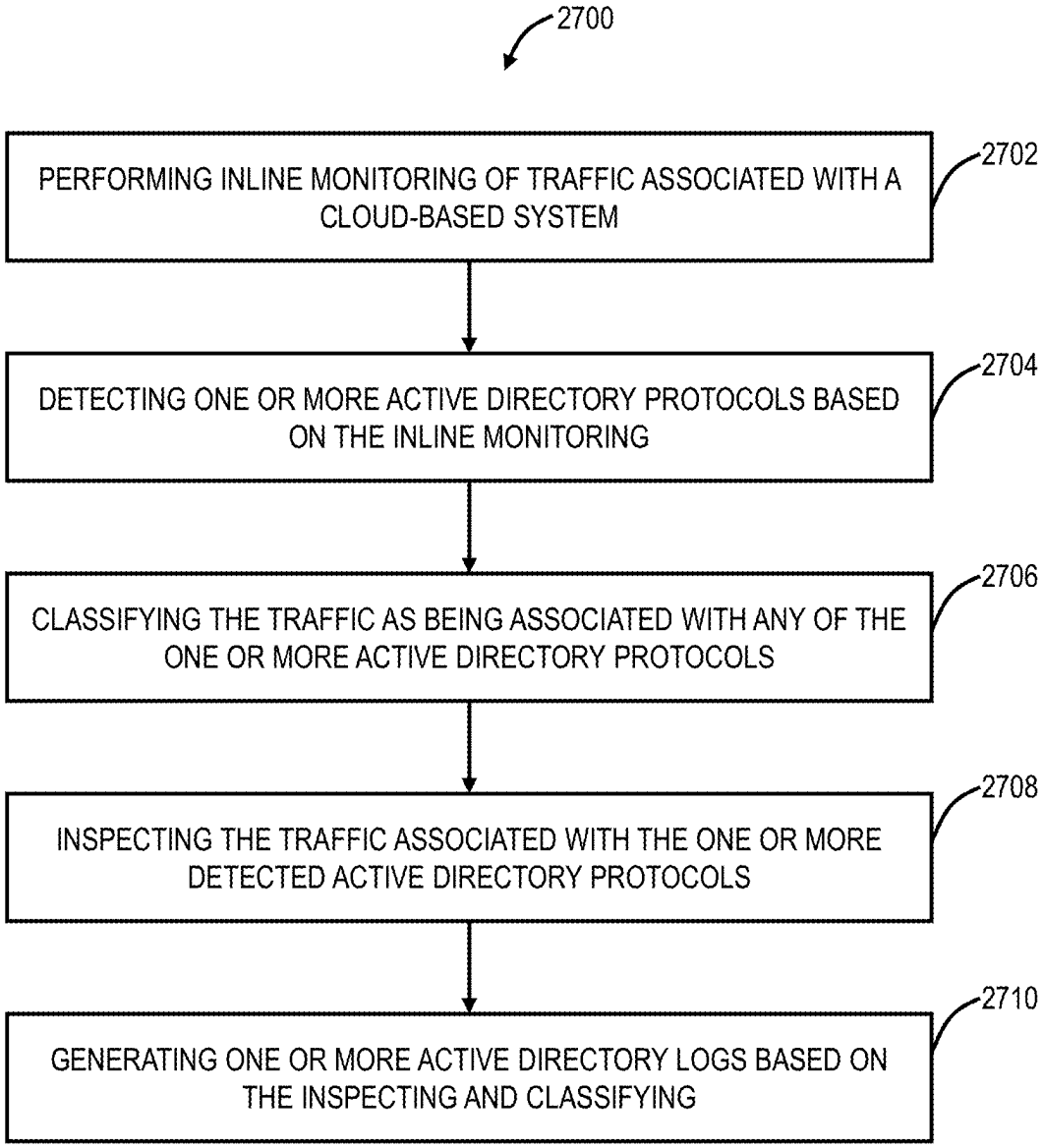

2700
PERFORMING INLINE MONITORING OF TRAFFIC ASSOCIATED WITH A CLOUD-BASED SYSTEM
2702
DETECTING ONE OR MORE ACTIVE DIRECTORY PROTOCOLS BASED ON THE INLINE MONITORING
2704
CLASSIFYING THE TRAFFIC AS BEING ASSOCIATED WITH ANY OF THE ONE OR MORE ACTIVE DIRECTORY PROTOCOLS
2706
INSPECTING THE TRAFFIC ASSOCIATED WITH THE ONE OR MORE DETECTED ACTIVE DIRECTORY PROTOCOLS
2708
GENERATING ONE OR MORE ACTIVE DIRECTORY LOGS BASED ON THE INSPECTING AND CLASSIFYING
2710

FIG. 27

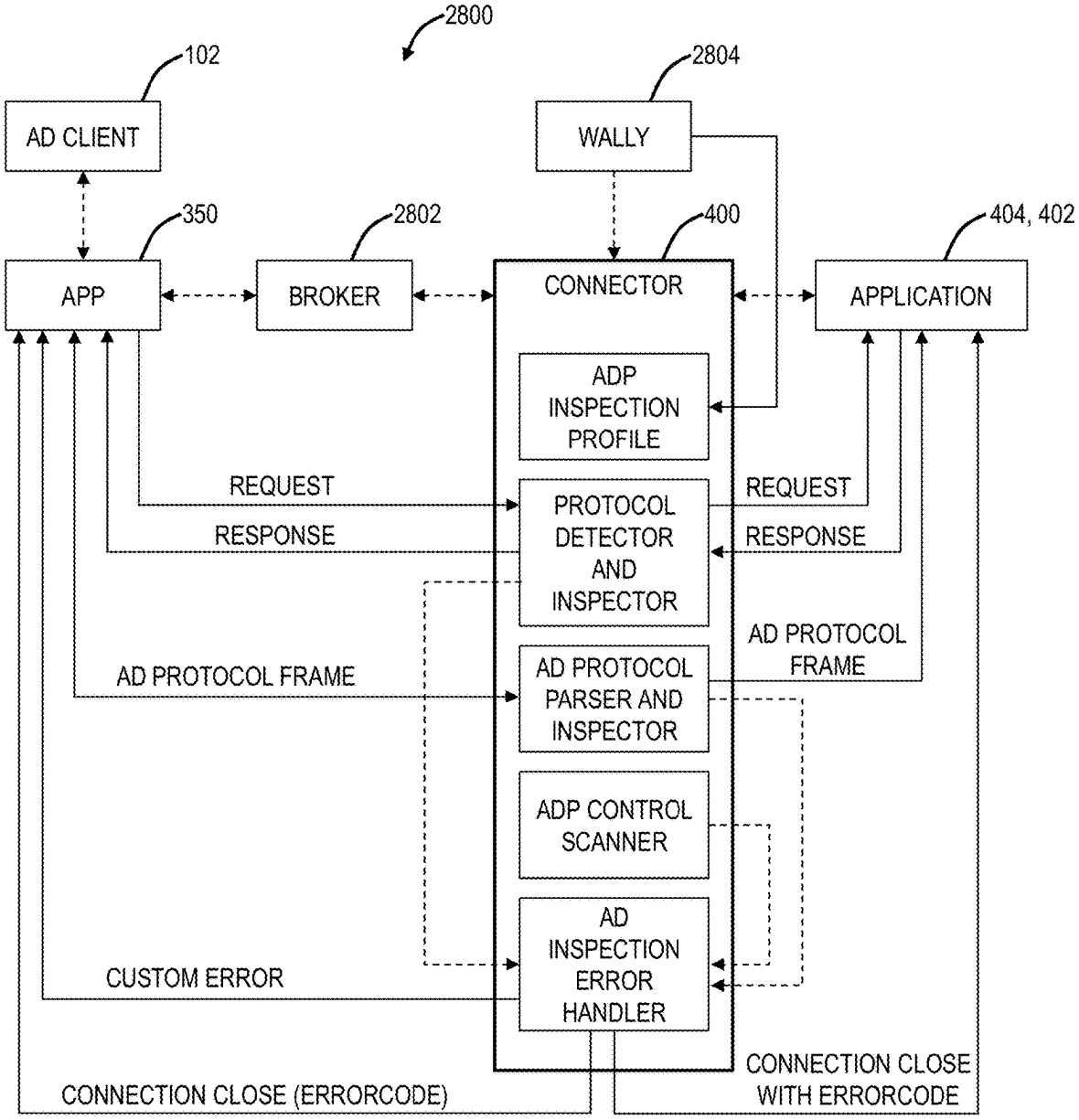

2800
102
AD CLIENT
2804
WALLY
350
2802
400
404, 402
APP
BROKER
CONNECTOR
APPLICATION
ADP INSPECTION PROFILE
REQUEST
RESPONSE
PROTOCOL DETECTOR AND INSPECTOR
REQUEST
RESPONSE
AD PROTOCOL FRAME
AD PROTOCOL PARSER AND INSPECTOR
AD PROTOCOL FRAME
ADP CONTROL SCANNER
AD INSPECTION ERROR HANDLER
CUSTOM ERROR
CONNECTION CLOSE (ERRORCODE)
CONNECTION CLOSE WITH ERRORCODE

FIG. 28

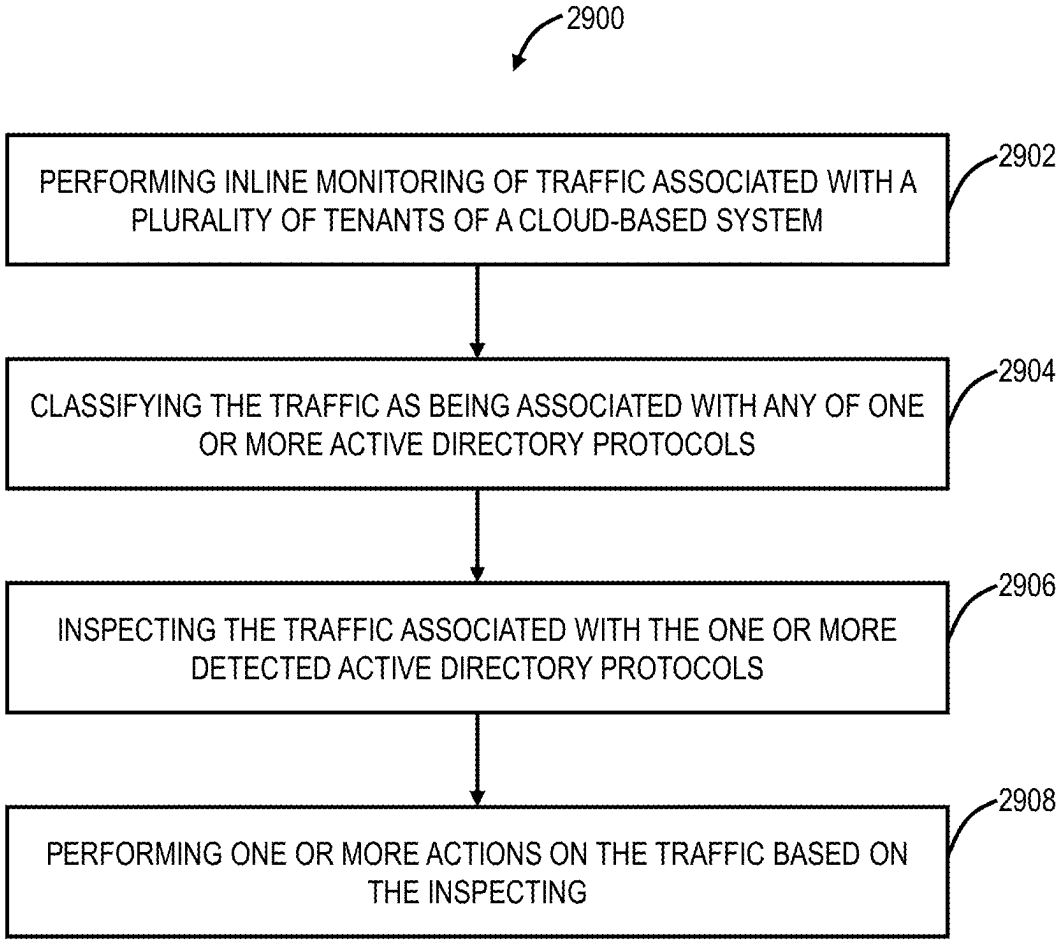

2900
PERFORMING INLINE MONITORING OF TRAFFIC ASSOCIATED WITH A PLURALITY OF TENANTS OF A CLOUD-BASED SYSTEM
2902
CLASSIFYING THE TRAFFIC AS BEING ASSOCIATED WITH ANY OF ONE OR MORE ACTIVE DIRECTORY PROTOCOLS
2904
INSPECTING THE TRAFFIC ASSOCIATED WITH THE ONE OR MORE DETECTED ACTIVE DIRECTORY PROTOCOLS
2906
PERFORMING ONE OR MORE ACTIONS ON THE TRAFFIC BASED ON THE INSPECTING
2908

FIG. 29

ACTIVE DIRECTORY SECURITY ENFORCEMENT AND THREAT INSIGHTS ON ZERO TRUST NETWORKS

CROSS-REFERENCE TO RELATED APPLICATION(S)

[0001] The present disclosure is a continuation-in-part of U.S. patent application Ser. No. 18/621,258, filed Mar. 29, 2024, entitled "Systems and Methods for Active Directory Protection in Zero Trust Networks," the contents of which are incorporated by reference in their entirety.

FIELD OF THE DISCLOSURE

[0002] The present disclosure generally relates to computer networking systems and methods. More particularly, the present disclosure relates to active directory security enforcement and threat insights on zero trust networks.

BACKGROUND OF THE DISCLOSURE

[0003] Active Directory (AD) can be prone to various types of attacks which can even bring down domain controllers and expose sensitive information. A plurality of types of protocol traffic can be seen on domain services, mainly Light-weight Directory Access Protocol (LDAP), Server Message Block (SMB), and Kerberos (KRB). The AD infrastructure is built as a centralized system that controls the entire Information Technology (IT) infrastructure, with access to applications, software, sensitive files, and confidential data. This makes AD a popular location for an attacker to breach security and the network. Currently, there is no solution for AD protection within zero trust networks. Traditional AD threat hunting is based on identifying different events related to activities on AD servers and clients using Antimalware Scan Interface (AMSI) tools. The present disclosure provides a feature for Active Directory Protection (ADP) inside zero trust networks by auto detecting AD protocols (i.e., KRB, LDAP, SMB) and inspecting the protocol messages to log necessary information for possible threat hunting.

BRIEF SUMMARY OF THE DISCLOSURE

[0004] The present disclosure relates to active directory security enforcement and threat insights on zero trust networks. In an embodiment, steps include performing inline monitoring of traffic associated with a plurality of tenants of the cloud-based system; classifying the traffic as being associated with any of one or more active directory protocols; inspecting the traffic associated with the one or more detected active directory protocols; and performing one or more actions on the traffic based on the inspecting.

[0005] The steps can further include wherein the inline monitoring includes real-time, live inspection of traffic associated with the one or more tenants. The inline monitoring can include real-time, live inspection of Kerberos, Light-weight Directory Access Protocol (LDAP), Server Message Block (SMB), and Distributed Computing Environment/Remote Procedure Calls (DCERPC) traffic. The inspecting can include determining one or more attack signatures based on the traffic. The one or more actions can include alerting users of a tenant of the cloud-based system responsive to detecting one or more attack signatures. The alerting can include providing remediation steps for mitigating a potential attack. The one or more actions can include blocking access to an active directory domain responsive to detecting one or more attack signatures. The inspecting can be performed for each of the plurality of tenants based on an inspection profile of each of the plurality of tenants. The steps can include receiving, from each of the plurality of tenants, an inspection profile for performing the inspecting. The inspecting can be performed at an application connector of the cloud-based system.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] The present disclosure is illustrated and described herein with reference to the various drawings, in which like reference numbers are used to denote like system components/method steps, as appropriate, and in which:

[0007] FIG. **1**A is a network diagram of a cloud-based system offering security as a service.

[0008] FIG. **1**B is a logical diagram of the cloud-based system operating as a zero-trust platform.

[0009] FIG. **1**C is a logical diagram illustrating zero trust policies with the cloud-based system and a comparison with the conventional firewall-based approach.

[0010] FIG. **2** is a network diagram of an example implementation of the cloud-based system.

[0011] FIG. **3** is a network diagram of the cloud-based system illustrating an application on the user devices with users configured to operate through the cloud-based system.

[0012] FIG. **4** is a block diagram of a server, which may be used in the cloud-based system, in other systems, or standalone.

[0013] FIG. **5** is a block diagram of a user device, which may be used with the cloud-based system or the like.

[0014] FIG. **6** is a network diagram of a Zero Trust Network Access (ZTNA) application utilizing the cloud-based system.

[0015] FIG. **7** is a network diagram of a VPN architecture for an intelligent, cloud-based global VPN.

[0016] FIG. **8** is a flowchart of a VPN process for an intelligent, cloud-based global VPN.

[0017] FIG. **9** is a network diagram illustrating the cloud-based system with private applications and data centers connected thereto to provide virtual private access through the cloud-based system.

[0018] FIG. **10** is a network diagram of a virtual private access network and a flowchart of a virtual private access process implemented thereon.

[0019] FIGS. **11** and **12** are network diagrams of a VPN configuration (FIG. **11**) compared to virtual private access (FIG. **12**) illustrating the differences therein.

[0020] FIGS. **13** and **14** are network diagrams of conventional private application access in the public cloud (FIG. **13**) compared to private applications in the public cloud with virtual private access (FIG. **14**).

[0021] FIGS. **15** and **16** are network diagrams of conventional contractor/partner access (FIG. **15**) of applications in the enterprise network compared to contractor/partner access (FIG. **16**) of the applications with virtual private access.

[0022] FIGS. **17** and **18** are network diagrams of a conventional network setup to share data between two companies (FIG. **17**) such as for Merger and Acquisition (M&A) purposes or the like compared to a network setup using virtual private access (FIG. **18**).

[0023] FIGS. **19** and **20** are screenshots of Graphical User Interfaces (GUIs) for administrator access to the virtual

private access with FIG. **19** illustrating a GUI of network auto-discovery and FIG. **20** illustrating a GUI for reporting.

[0024] FIG. **21** is a network diagram of the cloud-based system with a private service edge node in an enterprise network.

[0025] FIG. **22** is a flow diagram of main connector or data plane components for IDPS AD inspection.

[0026] FIGS. **23-25** are screenshots of a UI for configuring the present ADP system and reviewing trends.

[0027] FIG. **26** is a flow diagram of the present ADP feature.

[0028] FIG. **27** includes a flowchart of a process for active directory protection.

[0029] FIG. **28** is a flow diagram of the present Active Directory Protection (ADP) system.

[0030] FIG. **29** includes a flowchart of a process for active directory security enforcement.

DETAILED DESCRIPTION OF THE DISCLOSURE

[0031] The traditional view of an enterprise network (i.e., corporate, private, etc.) included a well-defined perimeter defended by various appliances (e.g., firewalls, intrusion prevention, advanced threat detection, etc.). In this traditional view, mobile users utilize a Virtual Private Network (VPN), etc. and have their traffic backhauled into the well-defined perimeter. This worked when mobile users represented a small fraction of the users, i.e., most users were within the well-defined perimeter. However, this is no longer the case—the definition of the workplace is no longer confined to within the well-defined perimeter, and with applications moving to the cloud, the perimeter has extended to the Internet. This results in an increased risk for the enterprise data residing on unsecured and unmanaged devices as well as the security risks in access to the Internet. Cloud-based security solutions have emerged, such as Zscaler Internet Access (ZIA) and Zscaler Private Access (ZPA), available from Zscaler, Inc., the applicant and assignee of the present application.

[0032] ZPA is a cloud service that provides seamless, zero trust access to private applications running on the public cloud, within the data center, within an enterprise network, etc. As described herein, ZPA is referred to as zero trust access to private applications or simply a zero trust access service. Here, applications are never exposed to the Internet, making them completely invisible to unauthorized users. The service enables the applications to connect to users via inside-out connectivity versus extending the network to them. Users are never placed on the network. This Zero Trust Network Access (ZTNA) approach supports both managed and unmanaged devices and any private application (not just web apps).

[0033] This Zero Trust Network Access (ZTNA) approach provides significant security in avoiding direct exposure of applications to the Internet. Rather, this ZTNA approach dials out from a connector. However, enterprise applications contain critical resources, and it is critical that any device accessing such applications, even though a ZTNA approach, are monitored.

[0034] The paradigm of the virtual private access systems and methods is to give users network access to get to an application, not to the entire network. If a user is not authorized to get the application, the user should not be able to even see that it exists, much less access it. The virtual private access systems and methods provide a new approach to deliver secure access by decoupling applications from the network, instead providing access with a lightweight software connector, in front of the applications, an application on the user device, a central authority to push policy, and a cloud to stitch the applications and the software connectors together, on a per-user, per-application basis.

[0035] With the virtual private access, users can only see the specific applications allowed by policy. Everything else is "invisible" or "dark" to them. Because the virtual private access separates the application from the network, the physical location of the application becomes irrelevant—if applications are located in more than one place, the user is automatically directed to the instance that will give them the best performance. The virtual private access also dramatically reduces configuration complexity, such as policies/firewalls in the data centers. Enterprises can, for example, move applications to Amazon Web Services or Microsoft Azure, and take advantage of the elasticity of the cloud, making private, internal applications behave just like the marketing leading enterprise applications. Advantageously, there is no hardware to buy or deploy because the virtual private access is a service offering to users and enterprises.

Example Cloud-Based System Architecture

[0036] FIG. **1**A is a network diagram of a cloud-based system **100** offering security as a service. Specifically, the cloud-based system **100** can offer a Secure Internet and Web Gateway as a service to various users **102**, as well as other cloud services. In this manner, the cloud-based system **100** is located between the users **102** and the Internet as well as any cloud services **106** (or applications) accessed by the users **102**. As such, the cloud-based system **100** provides inline monitoring inspecting traffic between the users **102**, the Internet **104**, and the cloud services **106**, including Secure Sockets Layer (SSL) traffic. The cloud-based system **100** can offer access control, threat prevention, data protection, etc. The access control can include a cloud-based firewall, cloud-based intrusion detection, Uniform Resource Locator (URL) filtering, bandwidth control, Domain Name System (DNS) filtering, etc. The threat prevention can include cloud-based intrusion prevention, protection against advanced threats (malware, spam, Cross-Site Scripting (XSS), phishing, etc.), cloud-based sandbox, antivirus, DNS security, etc. The data protection can include Data Loss Prevention (DLP), cloud application security such as via a Cloud Access Security Broker (CASB), file type control, etc.

[0037] The cloud-based firewall can provide Deep Packet Inspection (DPI) and access controls across various ports and protocols as well as being application and user aware. The URL filtering can block, allow, or limit website access based on policy for a user, group of users, or entire organization, including specific destinations or categories of URLs (e.g., gambling, social media, etc.). The bandwidth control can enforce bandwidth policies and prioritize critical applications such as relative to recreational traffic. DNS filtering can control and block DNS requests against known and malicious destinations.

[0038] The cloud-based intrusion prevention and advanced threat protection can deliver full threat protection against malicious content such as browser exploits, scripts, identified botnets and malware callbacks, etc. The cloud-based sandbox can block zero-day exploits (just identified) by analyzing unknown files for malicious behavior. Advan-

tageously, the cloud-based system **100** is multi-tenant and can service a large volume of the users **102**. As such, newly discovered threats can be promulgated throughout the cloud-based system **100** for all tenants practically instantaneously. The antivirus protection can include antivirus, antispyware, antimalware, etc. protection for the users **102**, using signatures sourced and constantly updated. The DNS security can identify and route command-and-control connections to threat detection engines for full content inspection.

[0039] The DLP can use standard and/or custom dictionaries to continuously monitor the users **102**, including compressed and/or SSL-encrypted traffic. Again, being in a cloud implementation, the cloud-based system **100** can scale this monitoring with near-zero latency on the users **102**. The cloud application security can include CASB functionality to discover and control user access to known and unknown cloud services **106**. The file type controls enable true file type control by the user, location, destination, etc. to determine which files are allowed or not.

[0040] For illustration purposes, the users **102** of the cloud-based system **100** can include a mobile device **110**, a headquarters (HQ) **112** which can include or connect to a data center (DC) **114**, Internet of Things (IOT) devices **116**, a branch office/remote location **118**, etc., and each includes one or more user devices (an example user device **300** is illustrated in FIG. **5**). The devices **110**, **116**, and the locations **112**, **114**, **118** are shown for illustrative purposes, and those skilled in the art will recognize there are various access scenarios and other users **102** for the cloud-based system **100**, all of which are contemplated herein. The users **102** can be associated with a tenant, which may include an enterprise, a corporation, an organization, etc. That is, a tenant is a group of users who share a common access with specific privileges to the cloud-based system **100**, a cloud service, etc. In an embodiment, the headquarters **112** can include an enterprise's network with resources in the data center **114**. The mobile device **110** can be a so-called road warrior, i.e., users that are off-site, on-the-road, etc. Those skilled in the art will recognize a user **102** has to use a corresponding user device **300** for accessing the cloud-based system **100** and the like, and the description herein may use the user **102** and/or the user device **300** interchangeably.

[0041] Further, the cloud-based system **100** can be multi-tenant, with each tenant having its own users **102** and configuration, policy, rules, etc. One advantage of the multi-tenancy and a large volume of users is the zero-day/zero-hour protection in that a new vulnerability can be detected and then instantly remediated across the entire cloud-based system **100**. The same applies to policy, rule, configuration, etc. changes-they are instantly remediated across the entire cloud-based system **100**. As well, new features in the cloud-based system **100** can also be rolled up simultaneously across the user base, as opposed to selective and time-consuming upgrades on every device at the locations **112**, **114**, **118**, and the devices **110**, **116**.

[0042] Logically, the cloud-based system **100** can be viewed as an overlay network between users (at the locations **112**, **114**, **118**, and the devices **110**, **116**) and the Internet **104** and the cloud services **106**. Previously, the IT deployment model included enterprise resources and applications stored within the data center **114** (i.e., physical devices) behind a firewall (perimeter), accessible by employees, partners, contractors, etc. on-site or remote via Virtual Private Networks (VPNs), etc. The cloud-based system **100** is replacing the conventional deployment model. The cloud-based system **100** can be used to implement these services in the cloud without requiring the physical devices and management thereof by enterprise IT administrators. As an ever-present overlay network, the cloud-based system **100** can provide the same functions as the physical devices and/or appliances regardless of geography or location of the users **102**, as well as independent of platform, operating system, network access technique, network access provider, etc.

[0043] There are various techniques to forward traffic between the users **102** at the locations **112**, **114**, **118**, and via the devices **110**, **116**, and the cloud-based system **100**. Typically, the locations **112**, **114**, **118** can use tunneling where all traffic is forward through the cloud-based system **100**. For example, various tunneling protocols are contemplated, such as Generic Routing Encapsulation (GRE), Layer Two Tunneling Protocol (L2TP), Internet Protocol (IP) Security (IPsec), customized tunneling protocols, etc. The devices **110**, **116**, when not at one of the locations **112**, **114**, **118** can use a local application that forwards traffic, a proxy such as via a Proxy Auto-Config (PAC) file, and the like. An application of the local application is the application **350** described in detail herein as a connector application. A key aspect of the cloud-based system **100** is all traffic between the users **102** and the Internet **104** or the cloud services **106** is via the cloud-based system **100**. As such, the cloud-based system **100** has visibility to enable various functions, all of which are performed off the user device in the cloud.

[0044] The cloud-based system **100** can also include a management system **120** for tenant access to provide global policy and configuration as well as real-time analytics. This enables IT administrators to have a unified view of user activity, threat intelligence, application usage, etc. For example, IT administrators can drill-down to a per-user level to understand events and correlate threats, to identify compromised devices, to have application visibility, and the like. The cloud-based system **100** can further include connectivity to an Identity Provider (IDP) **122** for authentication of the users **102** and to a Security Information and Event Management (SIEM) system **124** for event logging. The system **124** can provide alert and activity logs on a per-user **102** basis.

Zero Trust

[0045] FIG. **1**B is a logical diagram of the cloud-based system **100** operating as a zero-trust platform. Zero trust is a framework for securing organizations in the cloud and mobile world that asserts that no user or application should be trusted by default. Following a key zero trust principle, least-privileged access, trust is established based on context (e.g., user identity and location, the security posture of the endpoint, the app or service being requested) with policy checks at each step, via the cloud-based system **100**. Zero trust is a cybersecurity strategy wherein security policy is applied based on context established through least-privileged access controls and strict user authentication—not assumed trust. A well-tuned zero trust architecture leads to simpler network infrastructure, a better user experience, and improved cyberthreat defense.

[0046] Establishing a zero trust architecture requires visibility and control over the environment's users and traffic, including that which is encrypted; monitoring and verification of traffic between parts of the environment; and strong

multifactor authentication (MFA) methods beyond passwords, such as biometrics or one-time codes. This is performed via the cloud-based system **100**. Critically, in a zero trust architecture, a resource's network location is not the biggest factor in its security posture anymore. Instead of rigid network segmentation, your data, workflows, services, and such are protected by software-defined microsegmentation, enabling you to keep them secure anywhere, whether in your data center or in distributed hybrid and multicloud environments.

[0047] The core concept of zero trust is simple: assume everything is hostile by default. It is a major departure from the network security model built on the centralized data center and secure network perimeter. These network architectures rely on approved IP addresses, ports, and protocols to establish access controls and validate what's trusted inside the network, generally including anybody connecting via remote access VPN. In contrast, a zero trust approach treats all traffic, even if it is already inside the perimeter, as hostile. For example, workloads are blocked from communicating until they are validated by a set of attributes, such as a fingerprint or identity. Identity-based validation policies result in stronger security that travels with the workload wherever it communicates—in a public cloud, a hybrid environment, a container, or an on-premises network architecture.

[0048] Because protection is environment-agnostic, zero trust secures applications and services even if they communicate across network environments, requiring no architectural changes or policy updates. Zero trust securely connects users, devices, and applications using business policies over any network, enabling safe digital transformation. Zero trust is about more than user identity, segmentation, and secure access. It is a strategy upon which to build a cybersecurity ecosystem.

[0049] At its core are three tenets:

[0050] Terminate every connection: Technologies like firewalls use a "passthrough" approach, inspecting files as they are delivered. If a malicious file is detected, alerts are often too late. An effective zero trust solution terminates every connection to allow an inline proxy architecture to inspect all traffic, including encrypted traffic, in real time—before it reaches its destination—to prevent ransomware, malware, and more.

[0051] Protect data using granular context-based policies: Zero trust policies verify access requests and rights based on context, including user identity, device, location, type of content, and the application being requested. Policies are adaptive, so user access privileges are continually reassessed as context changes.

[0052] Reduce risk by eliminating the attack surface: With a zero trust approach, users connect directly to the apps and resources they need, never to networks (see ZTNA). Direct user-to-app and app-to-app connections eliminate the risk of lateral movement and prevent compromised devices from infecting other resources. Plus, users and apps are invisible to the internet, so they cannot be discovered or attacked.

[0053] FIG. **1**C is a logical diagram illustrating zero trust policies with the cloud-based system **100** and a comparison with the conventional firewall-based approach. Zero trust with the cloud-based system **100** allows per session policy decisions and enforcement regardless of the user **102** location. Unlike the conventional firewall-based approach, this eliminates attack surfaces, there are no inbound connections; prevents lateral movement, the user is not on the network; prevents compromise, allowing encrypted inspection; and prevents data loss with inline inspection.

Example Implementation of the Cloud-Based System

[0054] FIG. **2** is a network diagram of an example implementation of the cloud-based system **100**. In an embodiment, the cloud-based system **100** includes a plurality of enforcement nodes (EN) **150**, labeled as enforcement nodes **150-1**, **150-2**, **150**-N, interconnected to one another and interconnected to a central authority (CA) **152**. The nodes **150** and the central authority **152**, while described as nodes, can include one or more servers, including physical servers, virtual machines (VM) executed on physical hardware, etc. An example of a server is illustrated in FIG. **4**. The cloud-based system **100** further includes a log router **154** that connects to a storage cluster **156** for supporting log maintenance from the enforcement nodes **150**. The central authority **152** provide centralized policy, real-time threat updates, etc. and coordinates the distribution of this data between the enforcement nodes **150**. The enforcement nodes **150** provide an onramp to the users **102** and are configured to execute policy, based on the central authority **152**, for each user **102**. The enforcement nodes **150** can be geographically distributed, and the policy for each user **102** follows that user **102** as he or she connects to the nearest (or other criteria) enforcement node **150**.

[0055] Of note, the cloud-based system **100** is an external system meaning it is separate from tenant's private networks (enterprise networks) as well as from networks associated with the devices **110**, **116**, and locations **112**, **118**. Also, of note, the present disclosure describes a private enforcement node **150**P that is both part of the cloud-based system **100** and part of a private network. Further, of note, the enforcement node described herein may simply be referred to as a node or cloud node. Also, the terminology enforcement node **150** is used in the context of the cloud-based system **100** providing cloud-based security. In the context of secure, private application access, the enforcement node **150** can also be referred to as a service edge or service edge node. Also, a service edge node **150** can be a public service edge node (part of the cloud-based system **100**) separate from an enterprise network or a private service edge node (still part of the cloud-based system **100**) but hosted either within an enterprise network, in a data center **114**, in a branch office **118**, etc. Further, the term nodes as used herein with respect to the cloud-based system **100** (including enforcement nodes, service edge nodes, etc.) can be one or more servers, including physical servers, virtual machines (VM) executed on physical hardware, etc., as described above. The service edge node **150** can also be a Secure Access Service Edge (SASE).

[0056] The enforcement nodes **150** are full-featured secure internet gateways that provide integrated internet security. They inspect all web traffic bi-directionally for malware and enforce security, compliance, and firewall policies, as described herein, as well as various additional functionality. In an embodiment, each enforcement node **150** has two main modules for inspecting traffic and applying policies: a web module and a firewall module. The enforcement nodes **150** are deployed around the world and can handle hundreds of thousands of concurrent users with millions of concurrent sessions. Because of this, regardless of where the users **102** are, they can access the Internet **104**

from any device, and the enforcement nodes **150** protect the traffic and apply corporate policies. The enforcement nodes **150** can implement various inspection engines therein, and optionally, send sandboxing to another system. The enforcement nodes **150** include significant fault tolerance capabilities, such as deployment in active-active mode to ensure availability and redundancy as well as continuous monitoring.

[0057] In an embodiment, customer traffic is not passed to any other component within the cloud-based system **100**, and the enforcement nodes **150** can be configured never to store any data to disk. Packet data is held in memory for inspection and then, based on policy, is either forwarded or dropped. Log data generated for every transaction is compressed, tokenized, and exported over secure Transport Layer Security (TLS) connections to the log routers **154** that direct the logs to the storage cluster **156**, hosted in the appropriate geographical region, for each organization. In an embodiment, all data destined for or received from the Internet is processed through one of the enforcement nodes **150**. In another embodiment, specific data specified by each tenant, e.g., only email, only executable files, etc., is processed through one of the enforcement nodes **150**.

[0058] Each of the enforcement nodes **150** may generate a decision vector D=[d1, d2, . . . , dn] for a content item of one or more parts C=[c1, c2, . . . , cm]. Each decision vector may identify a threat classification, e.g., clean, spyware, malware, undesirable content, innocuous, spam email, unknown, etc. For example, the output of each element of the decision vector D may be based on the output of one or more data inspection engines. In an embodiment, the threat classification may be reduced to a subset of categories, e.g., violating, non-violating, neutral, unknown. Based on the subset classification, the enforcement node **150** may allow the distribution of the content item, preclude distribution of the content item, allow distribution of the content item after a cleaning process, or perform threat detection on the content item. In an embodiment, the actions taken by one of the enforcement nodes **150** may be determinative on the threat classification of the content item and on a security policy of the tenant to which the content item is being sent from or from which the content item is being requested by. A content item is violating if, for any part C=[c1, c2, . . . , cm] of the content item, at any of the enforcement nodes **150**, any one of the data inspection engines generates an output that results in a classification of "violating."

[0059] The central authority **152** hosts all customer (tenant) policy and configuration settings. It monitors the cloud and provides a central location for software and database updates and threat intelligence. Given the multi-tenant architecture, the central authority **152** is redundant and backed up in multiple different data centers. The enforcement nodes **150** establish persistent connections to the central authority **152** to download all policy configurations. When a new user connects to an enforcement node **150**, a policy request is sent to the central authority **152** through this connection. The central authority **152** then calculates the policies that apply to that user **102** and sends the policy to the enforcement node **150** as a highly compressed bitmap.

[0060] The policy can be tenant-specific and can include access privileges for users, websites and/or content that is disallowed, restricted domains, DLP dictionaries, etc. Once downloaded, a tenant's policy is cached until a policy change is made in the management system **120**. The policy can be tenant-specific and can include access privileges for users, websites and/or content that is disallowed, restricted domains, DLP dictionaries, etc. When this happens, all of the cached policies are purged, and the enforcement nodes **150** request the new policy when the user **102** next makes a request. In an embodiment, the enforcement node **150** exchange "heartbeats" periodically, so all enforcement nodes **150** are informed when there is a policy change. Any enforcement node **150** can then pull the change in policy when it sees a new request.

[0061] The cloud-based system **100** can be a private cloud, a public cloud, a combination of a private cloud and a public cloud (hybrid cloud), or the like. Cloud computing systems and methods abstract away physical servers, storage, networking, etc., and instead offer these as on-demand and elastic resources. The National Institute of Standards and Technology (NIST) provides a concise and specific definition which states cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Cloud computing differs from the classic client-server model by providing applications from a server that are executed and managed by a client's web browser or the like, with no installed client version of an application required. Centralization gives cloud service providers complete control over the versions of the browser-based and other applications provided to clients, which removes the need for version upgrades or license management on individual client computing devices. The phrase "Software as a Service" (SaaS) is sometimes used to describe application programs offered through cloud computing. A common shorthand for a provided cloud computing service (or even an aggregation of all existing cloud services) is "the cloud." The cloud-based system **100** is illustrated herein as an example embodiment of a cloud-based system, and other implementations are also contemplated.

[0062] As described herein, the terms cloud services and cloud applications may be used interchangeably. The cloud service **106** is any service made available to users on-demand via the Internet, as opposed to being provided from a company's on-premises servers. A cloud application, or cloud app, is a software program where cloud-based and local components work together. The cloud-based system **100** can be utilized to provide example cloud services, including Zscaler Internet Access (ZIA), Zscaler Private Access (ZPA), and Zscaler Digital Experience (ZDX), all from Zscaler, Inc. (the assignee and applicant of the present application). Also, there can be multiple different cloud-based systems **100**, including ones with different architectures and multiple cloud services. The ZIA service can provide the access control, threat prevention, and data protection described above with reference to the cloud-based system **100**. ZPA can include access control, microservice segmentation, etc. The ZDX service can provide monitoring of user experience, e.g., Quality of Experience (QoE), Quality of Service (QoS), etc., in a manner that can gain insights based on continuous, inline monitoring. For example, the ZIA service can provide a user with Internet Access, and the ZPA service can provide a user with access to enterprise resources instead of traditional Virtual Private Networks (VPNs), namely ZPA provides Zero Trust Net-

work Access (ZTNA). Those of ordinary skill in the art will recognize various other types of cloud services **106** are also contemplated. Also, other types of cloud architectures are also contemplated, with the cloud-based system **100** presented for illustration purposes.

User Device Application for Traffic Forwarding and Monitoring

[0063] FIG. **3** is a network diagram of the cloud-based system **100** illustrating an application **350** on user devices **300** with users **102** configured to operate through the cloud-based system **100**. Different types of user devices **300** are proliferating, including Bring Your Own Device (BYOD) as well as IT-managed devices. The conventional approach for a user device **300** to operate with the cloud-based system **100** as well as for accessing enterprise resources includes complex policies, VPNs, poor user experience, etc. The application **350** can automatically forward user traffic with the cloud-based system **100** as well as ensuring that security and access policies are enforced, regardless of device, location, operating system, or application. The application **350** automatically determines if a user **102** is looking to access the open Internet **104**, a SaaS app, or an internal app running in public, private, or the datacenter and routes mobile traffic through the cloud-based system **100**. The application **350** can support various cloud services, including ZIA, ZPA, ZDX, etc., allowing the best in class security with zero trust access to internal apps. As described herein, the application **350** can also be referred to as a connector application.

[0064] The application **350** is configured to auto-route traffic for seamless user experience. This can be protocol as well as application-specific, and the application **350** can route traffic with a nearest or best fit enforcement node **150**. Further, the application **350** can detect trusted networks, allowed applications, etc. and support secure network access. The application **350** can also support the enrollment of the user device **300** prior to accessing applications. The application **350** can uniquely detect the users **102** based on fingerprinting the user device **300**, using criteria like device model, platform, operating system, etc. The application **350** can support Mobile Device Management (MDM) functions, allowing IT personnel to deploy and manage the user devices **300** seamlessly. This can also include the automatic installation of client and SSL certificates during enrollment. Finally, the application **350** provides visibility into device and app usage of the user **102** of the user device **300**.

[0065] The application **350** supports a secure, lightweight tunnel between the user device **300** and the cloud-based system **100**. For example, the lightweight tunnel can be HTTP-based. With the application **350**, there is no requirement for PAC files, an IPSec VPN, authentication cookies, or user **102** setup.

Example Server Architecture

[0066] FIG. **4** is a block diagram of a server **200**, which may be used in the cloud-based system **100**, in other systems, or standalone. For example, the enforcement nodes **150** and the central authority **152** may be formed as one or more of the servers **200**. The server **200** may be a digital computer that, in terms of hardware architecture, generally includes a processor **202**, input/output (I/O) interfaces **204**, a network interface **206**, a data store **208**, and memory **210**. It should be appreciated by those of ordinary skill in the art that FIG. **4** depicts the server **200** in an oversimplified manner, and a practical embodiment may include additional components and suitably configured processing logic to support known or conventional operating features that are not described in detail herein. The components (**202**, **204**, **206**, **208**, and **210**) are communicatively coupled via a local interface **212**. The local interface **212** may be, for example, but not limited to, one or more buses or other wired or wireless connections, as is known in the art. The local interface **212** may have additional elements, which are omitted for simplicity, such as controllers, buffers (caches), drivers, repeaters, and receivers, among many others, to enable communications. Further, the local interface **212** may include address, control, and/or data connections to enable appropriate communications among the aforementioned components.

[0067] The processor **202** is a hardware device for executing software instructions. The processor **202** may be any custom made or commercially available processor, a Central Processing Unit (CPU), an auxiliary processor among several processors associated with the server **200**, a semiconductor-based microprocessor (in the form of a microchip or chipset), or generally any device for executing software instructions. When the server **200** is in operation, the processor **202** is configured to execute software stored within the memory **210**, to communicate data to and from the memory **210**, and to generally control operations of the server **200** pursuant to the software instructions. The I/O interfaces **204** may be used to receive user input from and/or for providing system output to one or more devices or components.

[0068] The network interface **206** may be used to enable the server **200** to communicate on a network, such as the Internet **104**. The network interface **206** may include, for example, an Ethernet card or adapter or a Wireless Local Area Network (WLAN) card or adapter. The network interface **206** may include address, control, and/or data connections to enable appropriate communications on the network. A data store **208** may be used to store data. The data store **208** may include any of volatile memory elements (e.g., random access memory (RAM, such as DRAM, SRAM, SDRAM, and the like)), nonvolatile memory elements (e.g., ROM, hard drive, tape, CDROM, and the like), and combinations thereof.

[0069] Moreover, the data store **208** may incorporate electronic, magnetic, optical, and/or other types of storage media. In one example, the data store **208** may be located internal to the server **200**, such as, for example, an internal hard drive connected to the local interface **212** in the server **200**. Additionally, in another embodiment, the data store **208** may be located external to the server **200** such as, for example, an external hard drive connected to the I/O interfaces **204** (e.g., SCSI or USB connection). In a further embodiment, the data store **208** may be connected to the server **200** through a network, such as, for example, a network-attached file server.

[0070] The memory **210** may include any of volatile memory elements (e.g., random access memory (RAM, such as DRAM, SRAM, SDRAM, etc.)), nonvolatile memory elements (e.g., ROM, hard drive, tape, CDROM, etc.), and combinations thereof. Moreover, the memory **210** may incorporate electronic, magnetic, optical, and/or other types of storage media. Note that the memory **210** may have a

distributed architecture, where various components are situated remotely from one another but can be accessed by the processor **202**. The software in memory **210** may include one or more software programs, each of which includes an ordered listing of executable instructions for implementing logical functions. The software in the memory **210** includes a suitable Operating System (O/S) **214** and one or more programs **216**. The operating system **214** essentially controls the execution of other computer programs, such as the one or more programs **216**, and provides scheduling, input-output control, file and data management, memory management, and communication control and related services. The one or more programs **216** may be configured to implement the various processes, algorithms, methods, techniques, etc. described herein.

Example User Device Architecture

[0071] FIG. **5** is a block diagram of a user device **300**, which may be used with the cloud-based system **100** or the like. Specifically, the user device **300** can form a device used by one of the users **102**, and this may include common devices such as laptops, smartphones, tablets, netbooks, personal digital assistants, MP3 players, cell phones, e-book readers, IoT devices, servers, desktops, printers, televisions, streaming media devices, and the like. The user device **300** can be a digital device that, in terms of hardware architecture, generally includes a processor **302**, I/O interfaces **304**, a network interface **306**, a data store **308**, and memory **310**. It should be appreciated by those of ordinary skill in the art that FIG. **5** depicts the user device **300** in an oversimplified manner, and a practical embodiment may include additional components and suitably configured processing logic to support known or conventional operating features that are not described in detail herein. The components (**302**, **304**, **306**, **308**, and **302**) are communicatively coupled via a local interface **312**. The local interface **312** can be, for example, but not limited to, one or more buses or other wired or wireless connections, as is known in the art. The local interface **312** can have additional elements, which are omitted for simplicity, such as controllers, buffers (caches), drivers, repeaters, and receivers, among many others, to enable communications. Further, the local interface **312** may include address, control, and/or data connections to enable appropriate communications among the aforementioned components.

[0072] The processor **302** is a hardware device for executing software instructions. The processor **302** can be any custom made or commercially available processor, a CPU, an auxiliary processor among several processors associated with the user device **300**, a semiconductor-based microprocessor (in the form of a microchip or chipset), or generally any device for executing software instructions. When the user device **300** is in operation, the processor **302** is configured to execute software stored within the memory **310**, to communicate data to and from the memory **310**, and to generally control operations of the user device **300** pursuant to the software instructions. In an embodiment, the processor **302** may include a mobile optimized processor such as optimized for power consumption and mobile applications. The I/O interfaces **304** can be used to receive user input from and/or for providing system output. User input can be provided via, for example, a keypad, a touch screen, a scroll ball, a scroll bar, buttons, a barcode scanner, and the like. System output can be provided via a display device such as a Liquid Crystal Display (LCD), touch screen, and the like.

[0073] The network interface **306** enables wireless communication to an external access device or network. Any number of suitable wireless data communication protocols, techniques, or methodologies can be supported by the network interface **306**, including any protocols for wireless communication. The data store **308** may be used to store data. The data store **308** may include any of volatile memory elements (e.g., random access memory (RAM, such as DRAM, SRAM, SDRAM, and the like)), nonvolatile memory elements (e.g., ROM, hard drive, tape, CDROM, and the like), and combinations thereof. Moreover, the data store **308** may incorporate electronic, magnetic, optical, and/or other types of storage media.

[0074] The memory **310** may include any of volatile memory elements (e.g., random access memory (RAM, such as DRAM, SRAM, SDRAM, etc.)), nonvolatile memory elements (e.g., ROM, hard drive, etc.), and combinations thereof. Moreover, the memory **310** may incorporate electronic, magnetic, optical, and/or other types of storage media. Note that the memory **310** may have a distributed architecture, where various components are situated remotely from one another but can be accessed by the processor **302**. The software in memory **310** can include one or more software programs, each of which includes an ordered listing of executable instructions for implementing logical functions. In the example of FIG. **3**, the software in the memory **310** includes a suitable operating system **314** and programs **316**. The operating system **314** essentially controls the execution of other computer programs and provides scheduling, input-output control, file and data management, memory management, and communication control and related services. The programs **316** may include various applications, add-ons, etc. configured to provide end user functionality with the user device **300**. For example, example programs **316** may include, but not limited to, a web browser, social networking applications, streaming media applications, games, mapping and location applications, electronic mail applications, financial applications, and the like. In a typical example, the end-user typically uses one or more of the programs **316** along with a network such as the cloud-based system **100**.

Zero Trust Network Access Using the Cloud-Based System

[0075] FIG. **6** is a network diagram of a Zero Trust Network Access (ZTNA) application utilizing the cloud-based system **100**. For ZTNA, the cloud-based system **100** can dynamically create a connection through a secure tunnel between an endpoint (e.g., users **102**A, **102**B) that are remote and an on-premises connector **400** that is either located in cloud file shares and applications **402** and/or in an enterprise network **410** that includes enterprise file shares and applications **404**. The connection between the cloud-based system **100** and on-premises connector **400** is dynamic, on-demand, and orchestrated by the cloud-based system **100**. A key feature is its security at the edge—there is no need to punch any holes in the existing on-premises firewall. The connector **400** inside the enterprise (on-premises) "dials out" and connects to the cloud-based system **100** as if too were an endpoint. This on-demand dial-out capability and tunneling authenticated traffic back to the enterprise is a key differentiator for ZTNA. Also, this functionality can be implemented in part by the application **350** on

the user device **300**. Also, the applications **402**, **404** can include B2B applications. Note, the difference between the applications **402**, **404** is the applications **402** are hosted in the cloud, whereas the applications **404** are hosted on the enterprise network **410**. The B2B service described herein contemplates use with either or both of the applications **402**, **404**.

[0076] The paradigm of virtual private access systems and methods is to give users network access to get to an application and/or file share, not to the entire network. If a user is not authorized to get the application, the user should not be able even to see that it exists, much less access it. The virtual private access systems and methods provide an approach to deliver secure access by decoupling applications **402**, **404** from the network, instead of providing access with a connector **400**, in front of the applications **402**, **404**, an application on the user device **300**, a central authority **152** to push policy, and the cloud-based system **100** to stitch the applications **402**, **404** and the software connectors **400** together, on a per-user, per-application basis.

[0077] With the virtual private access, users can only see the specific applications **402**, **404** allowed by the central authority **152**. Everything else is "invisible" or "dark" to them. Because the virtual private access separates the application from the network, the physical location of the application **402**, **404** becomes irrelevant-if applications **402**, **404** are located in more than one place, the user is automatically directed to the instance that will give them the best performance. The virtual private access also dramatically reduces configuration complexity, such as policies/firewalls in the data centers. Enterprises can, for example, move applications to Amazon Web Services or Microsoft Azure, and take advantage of the elasticity of the cloud, making private, internal applications behave just like the marketing leading enterprise applications. Advantageously, there is no hardware to buy or deploy because the virtual private access is a service offering to end-users and enterprises.

VPN Architecture

[0078] FIG. **7** is a network diagram of a VPN architecture **405** for an intelligent, cloud-based global VPN. For illustration purposes, the VPN architecture **405** includes the cloud-based system **100**, the Internet **104**, the applications **402** in SaaS/public cloud systems, and the enterprise network **410**. The VPN architecture **405** also includes a user **102**, which can include any computing device/platform connecting to the cloud-based system **100**, the Internet **104**, the applications **402**, and the enterprise network **410**. The VPN architecture **405** includes a single user **102** for illustration purposes, but those of ordinary skill in the art will recognize that the VPN architecture **405** contemplates a plurality of users **102**. The user **102** can be a nomadic user, a regional/branch office, etc. That is, the user **102** can be any user of the enterprise network **410** that is physically located outside a firewall **412** associated with the enterprise network **410**. The SaaS/public cloud systems can include any systems containing computing and data assets in the cloud such as, for example, Microsoft OneDrive, Google Drive, Dropbox, Apple iCloud, Customer Relationship Management (CRM) systems, SCM, Sales management systems, etc. The enterprise network **410** includes local computing and data assets behind the firewall **412** for additional security on highly confidential assets or legacy assets not yet migrated to the cloud.

[0079] The user **102** needs to access the Internet **104**, the SaaS/public cloud systems for the applications **402**, and the enterprise network **410**. Again, conventionally, the solution for secure communication, the user **102** has a VPN connection through the firewall **412** where all data is sent to the enterprise network **410**, including data destined for the Internet **104** or the SaaS/public cloud systems for the applications **402**. Furthermore, this VPN connection dials into the enterprise network **410**. The systems and methods described herein provide the VPN architecture **405**, which provides a secure connection to the enterprise network **410** without bringing all traffic, e.g., traffic for the Internet **104** or the SaaS/public cloud systems, into the enterprise network **410** as well as removing the requirement for the user **102** to dial into the enterprise network **410**.

[0080] Instead of the user **102** creating a secure connection through the firewall **412**, the user **102** connects securely to a VPN device **420** located in the cloud-based system **100** through a secure connection **422**. Note, the cloud-based system **100** can include a plurality of VPN devices **420**. The VPN architecture **405** dynamically routes traffic between the user **102** and the Internet **104**, the SaaS/public cloud systems for the applications **402**, and securely with the enterprise network **410**. For secure access to the enterprise network **410**, the VPN architecture **405** includes dynamically creating connections through secure tunnels between three entities: the VPN device **420**, the cloud, and an on-premises redirection proxy **430**. The connection between the cloud-based system **100** and the on-premises redirection proxy **430** is dynamic, on-demand and orchestrated by the cloud-based system **100**. A key feature of the systems and methods is its security at the edge of the cloud-based system **100**—there is no need to punch any holes in the existing on-premises firewall **412**. The on-premises redirection proxy **430** inside the enterprise network **410** "dials out" and connects to the cloud-based system **100** as if too were an end-point via secure connections **440**, **442**. This on-demand dial-out capability and tunneling authenticated traffic back to the enterprise network **410** is a key differentiator.

[0081] The VPN architecture **405** includes the VPN devices **420**, the on-premises redirection proxy **430**, a topology controller **450**, and an intelligent DNS proxy **460**. The VPN devices **420** can be Traffic (VPN) distribution servers and can be part of the cloud-based system **100**. In an embodiment, the cloud-based system **100** can be a security cloud such as available from Zscaler, Inc. (www.zscaler.com) performing functions on behalf of every client that connects to it: a) allowing/denying access to specific Internet sites/apps—based on security policy and absence/presence of malware in those sites, and b) set policies on specific SaaS apps and allowing/denying access to specific employees or groups.

[0082] The on-premises redirection proxy **430** is located inside a perimeter of the enterprise network **410** (inside the private cloud or inside the corporate data center—depending on the deployment topology). It is connected to a local network and acts as a "bridge" between the users **102** outside the perimeter and apps that are inside the perimeter through the secure connections **440**, **442**. But, this "bridge" is always closed—it is only open to the users **102** that pass two criteria: a) they must be authenticated by an enterprise authentication service **470**, and b) the security policy in effect allows them access to "cross the bridge."

[0083] When the on-premises redirection proxy **430** starts, it establishes a persistent, long-lived connection **472** to the topology controller **450**. The topology controller **450** connects to the on-premises redirection proxy **430** through a secure connection **472** and to the cloud-based system **100** through a secure connection **480**. The on-premises redirection proxy **430** waits for instruction from the topology controller **450** to establish tunnels to specific VPN termination nodes, i.e., the VPN devices **420**, in the cloud-based system **100**. The on-premises redirection proxy **430** is most expediently realized as custom software running inside a virtual machine (VM). The topology controller **450**, as part of the non-volatile data for each enterprise, stores the network topology of a private network of the enterprise network **410**, including, but not limited to, the internal domain name(s), subnet(s) and other routing information.

[0084] The DNS proxy **460** handles all domain names to Internet Protocol (IP) Address resolution on behalf of endpoints (clients). These endpoints are user computing devices—such as mobile devices, laptops, tablets, etc. The DNS proxy **460** consults the topology controller **450** to discern packets that must be sent to the Internet **104**, the SaaS/public cloud systems, vs. the enterprise network **410** private network. This decision is made by consulting the topology controller **450** for information about a company's private network and domains. The DNS proxy **460** is connected to the user **102** through a connection **482** and to the cloud-based system **100** through a connection **484**.

[0085] The VPN device **420** is located in the cloud-based system **100** and can have multiple points-of-presence around the world. If the cloud-based system **100** is a distributed security cloud, the VPN device **420** can be located with enforcement nodes **150**. In general, the VPN device **420** can be implemented as software instances on the enforcement nodes **150**, as a separate virtual machine on the same physical hardware as the enforcement nodes **150**, or a separate hardware device such as the server **200**, but part of the cloud-based system **100**. The VPN device **420** is the first point of entry for any client wishing to connect to the Internet **104**, SaaS apps, or the enterprise private network. In addition to doing traditional functions of a VPN server, the VPN device **420** works in concert with the topology controller **450** to establish on-demand routes to the on-premises redirection proxy **430**. These routes are set up for each user on demand. When the VPN device **420** determines that a packet from the user **102** is destined for the enterprise private network, it encapsulates the packet and sends it via a tunnel between the VPN device **420** and the on-premises redirection proxy **430**. For packets meant for the Internet **104** or SaaS clouds, the VPN device **420** can forwards it to the enforcement nodes **150**—to continue processing as before or send directly to the Internet **104** or SaaS clouds.

VPN Process

[0086] FIG. **8** is a flowchart of a VPN process **500** for an intelligent, cloud-based global VPN. The VPN process **500** can be implemented through the VPN architecture **405**. The VPN process **500** includes the user **102** connecting to the cloud-based system **100** through authentication (step **510**). Once the authentication is complete, a VPN is established between the user **102** and a VPN server in the cloud-based system **100** and DNS for the user **102** is set to a DNS proxy **460** (step **520**). Now, the user **102** has a secure VPN connection to the cloud-based system **100**. Subsequently, the user **102** sends a request to the cloud-based system **100** via the DNS proxy **460** (step **530**). Here, the request can be anything—request for the enterprise network **410**, the Internet **104**, the applications **402** in the SaaS/public cloud systems, the applications **404** in the enterprise network **410**, etc. The DNS proxy **460** contacts the topology controller **450** with the identity of the user and the request (step **540**). That is, whenever the user **102** wishes to reach a destination (Internet, Intranet, SaaS, etc.), it will consult the DNS proxy **460** to obtain the address of the destination.

[0087] For non-enterprise requests, the cloud-based system **100** forwards the request per policy (step **550**). Here, the cloud-based system **100** can forward the request based on the policy associated with the enterprise network **410** and the user **102**. With the identity of the user and the enterprise they belong to, the VPN server will contact the topology controller **450** and pre-fetch the enterprise private topology. For enterprise requests, the topology controller **450** fetches a private topology of the enterprise network **410**, instructs the redirection proxy **430** to establish an outbound tunnel to the VPN server, the redirection proxy **430** establishes the outbound tunnel, and requests are forward between the user **102** and the enterprise network **410** securely (step **560**). Here, the DNS proxy **460** works with the topology controller **450** to determine the local access in the enterprise network **410**, and the topology controller **450** works with the redirection proxy **430** to dial out a secure connection to the VPN server. The redirection proxy **430** establishes an on-demand tunnel to the specific VPN server so that it can receive packets meant for its internal network.

Global VPN Applications

[0088] Advantageously, the systems and methods avoid the conventional requirement of VPN tunneling all data into the enterprise network **410** and hair-pinning non-enterprise data back out. The systems and methods also allow the enterprise network **410** to have remote offices, etc. without requiring large hardware infrastructures—the cloud-based system **100** bridges the users **102**, remote offices, etc. to the enterprise network **410** in a seamless manner while removing the requirement to bring non-enterprise data through the enterprise network **410**. This recognizes the shift to mobility in enterprise applications. Also, the VPN tunnel on the user **102** can leverage and use existing VPN clients available on the user devices **300**. The cloud-based system **100**, through the VPN architecture **405**, determines how to route traffic for the user **102** efficiently—only enterprise traffic is routed securely to the enterprise network **410**. Additionally, the VPN architecture **405** removes the conventional requirement of tunneling into the enterprise network **410**, which can be an opportunity for security vulnerabilities. Instead, the redirection proxy **430** dials out of the enterprise network **410**.

[0089] The systems and methods provide, to the user (enterprise user), a single, seamless way to connect to Public and Private clouds—with no special steps needed to access one vs. the other. To the IT Admin, the systems and methods provide a single point of control and access for all users—security policies and rules are enforced at a single global cloud chokepoint—without impacting user convenience/performance or weakening security.

Virtual Private Access Via the Cloud

[0090] FIG. **9** is a network diagram illustrating the cloud-based system **100** with private applications **402**, **404** and

data centers **610** connected thereto to provide virtual private access through the cloud-based system **100**. In an aspect, the virtual private access described herein leverages the cloud-based system **100** to enable various users **102** including remote users, contractors, partners, business customers, etc., i.e., anyone who needs access to the private applications **402**, **404** and the data centers **610** access, without granting unfettered access to the internal network, without requiring hardware or appliances, and in a seamless manner from the users' **102** perspective. The private applications **402**, **404** include applications dealing with financial data, personal data, medical data, intellectual property, records, etc., that is the private applications **404** can be available on the enterprise network **410**, but not available remotely except conventionally via VPN access. Examples of the private applications **402**, **404** can include Customer Relationship Management (CRM), sales automation, financial applications, time management, document management, etc. Also, the applications **402**, **404** can be B2B applications or services as described herein.

[0091] The virtual private access is a new technique for the users **102** to access the file shares and applications **402**, **404**, without the cost, hassle or security risk of VPNs, which extend network access to deliver app access. The virtual private access decouples private internal applications from the physical network to enable authorized user access to the file shares and applications **402**, **404**, without the security risk or complexity of VPNs. That is, virtual private access takes the "Network" out of VPNs.

[0092] In the virtual private access, the users **102**, the file shares and applications **402**, **404**, are communicatively coupled to the cloud-based system **100**, such as via the Internet **104** or the like. On the client-side, at the users **102**, the applications **402**, **404** provision both secure remote access and optionally accessibility to the cloud-based system **100**. The application **402**, **404** establishes a connection to the closest enforcement node **150** in the cloud-based system **100** at startup and may not accept incoming requests.

[0093] At the file shares and applications **402**, **404**, the lightweight connectors **400** sit in front of the applications **402**, **404**. The lightweight connectors **400** become the path to the file shares and applications **402**, **404** behind it, and connect only to the cloud-based system **100**. The lightweight connectors **400** can be lightweight, ephemeral binary, such as deployed as a virtual machine, to establish a connection between the file shares and applications **402**, **404** and the cloud-based system **100**, such as via the closest enforcement node **150**. The lightweight connectors **400** do not accept inbound connections of any kind, dramatically reducing the overall threat surface. The lightweight connectors **400** can be enabled on a standard VMware platform; additional lightweight connectors **400** can be created in less than 5 seconds to handle additional application instances. By not accepting inbound connections, the lightweight connectors **400** make the file shares and applications **402**, **404** "dark," removing a significant threat vector.

[0094] The policy can be established and pushed by policy engines in the central authority **152**, such as via a distributed cluster of multi-tenant policy engines that provide a single interface for all policy creation. Also, no data of any kind transits the policy engines. The enforcement nodes **150** in the security cloud stitch connections together, between the users **102** and the file shares and applications **402**, **404**, without processing traffic of any kind. When the user **102** requests an application in the file shares and applications **402**, **404**, the policy engine delivers connection information to the application **350** and app-side enforcement nodes **150**, which includes the location of a single enforcement nodes **150** to provision the client/app connection. The connection is established through the enforcement nodes **150**, and is encrypted with a combination of the customer's client and server-side certificates. While the enforcement nodes **150** provision the connection, they do not participate in the key exchange, nor do they have visibility into the traffic flows.

[0095] Advantageously, the virtual private access provides increased security in that the file shares and applications **402**, **404** are visible only to the users **102** that are authorized to access them; unauthorized users are not able to even see them. Because application access is provisioned through the cloud-based system **100**, rather than via a network connection, the virtual private access makes it impossible to route back to applications. The virtual private access is enabled using the application **350**, without the need to launch or exit VPN clients. The application access just works in the background enabling application-specific access to individual contractors, business partners or other companies, i.e., the users **102**.

[0096] The virtual private access provides capital expense (CAPEX) and operating expense (OPEX) reductions as there is no hardware to deploy, configure, or maintain. Legacy VPNs can be phased out. Internal IT can be devoted to enabling business strategy, rather than maintaining network "plumbing." Enterprises can move apps to the cloud on their schedule, without the need to re-architect, set up site-to-site VPNs or deliver a substandard user experience.

[0097] The virtual private access provides easy deployment, i.e., put lightweight connectors **400** in front of the file shares and applications **402**, **404**, wherever they are. The virtual private access will automatically route to the location that delivers the best performance. Wildcard app deployment will discover applications upon request, regardless of their location, then build granular user access policies around them. There is no need for complex firewall rules, Network Address Translation issues or policy juggling to deliver application access. Further, the virtual private access provides seamless integration with existing Single Sign-On (SSO) infrastructure.

[0098] FIG. **10** is a network diagram of a virtual private access network **700**A and a flowchart of a virtual private access process **750** implemented thereon. The cloud-based system **100** includes three enforcement nodes **150**A, **150**B, **150**C, assume for illustration purposes in San Francisco, New York, and London, respectively. The user **102** has the application **350** executing on the user device **300**, which is communicatively coupled to the enforcement node **150**A. The enterprise file share and application **402**, **404** is communicatively coupled to the enforcement node **150**C. Note, there can be direct connectivity between the enforcement nodes **150**A, **150**C, the enforcement nodes **150**A, **150**C can connect through the enforcement node **150**B, or both the user **102** and the enterprise file share and application **402**, **404** can be connected to the same node **150**. That is, the architecture of the cloud-based system **100** can include various implementations.

[0099] The virtual private access process **750** is described with reference to both the user **102**, the cloud-based system **100**, and the enterprise file share and application **402**, **404**. First, the user **102** is executing the application **350** on the

user device **300**, in the background. The user **102** launches the application **350** and can be redirected to an enterprise ID provider or the like to sign on, i.e., a single sign on, without setting up new accounts. Once authenticated, Public Key Infrastructure (PKI) certificate **720** enrollment occurs, between the user **102** and the enforcement node **150**A. With the application **350** executing on the user device, the user **102** makes a request to the enterprise file share and application **402**, **404**, e.g., intranet.company.com, crm.company.com, etc. (step **752**). Note, the request is not limited to web applications and can include anything such as a remote desktop or anything handling any static Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) applications.

[0100] This request is intercepted by the enforcement node **150**A and redirected to the central authority **152**, which performs a policy lookup for the user **102** and the user device **300** (step **754**), transparent to the user **102**. The central authority **152** determines if the user **102** and the user device **300** are authorized for the enterprise file share and application **402**, **404**. Once authorization is determined, the central authority **152** provides information to the enforcement nodes **150**A, **150**B, **150**C, the application **350**, and the lightweight connectors **400** at the enterprise file share and application **402**, **404**, and the information can include the certificates **720** and other details necessary to stitch secure connections between the various devices. Specifically, the central authority **152** can create connection information with the best enforcement nodes **150** for joint connections, from the user **102** to the enterprise file share and application **402**, **404**, and the unique tokens (step **756**). With the connection information, the enforcement node **150**A connects to the user **102**, presenting a token, and the enforcement node **150**C connects to the lightweight connector **400**, presenting a token (step **758**). Now, a connection is stitched between the user **102** to the enterprise file share and application **402**, **404**, through the application **350**, the enforcement nodes **150**A, **150**B, **150**C, and the lightweight connector **400**.

Comparison—VPN with Virtual Private Access

[0101] FIGS. **11** and **12** are network diagrams of a VPN configuration (FIG. **11**) compared to virtual private access (FIG. **12**) illustrating the differences therein. In FIG. **11**, a user device **300** connects to a VPN termination device **804** associated with an enterprise network **806** via the Internet **104**, such that the user device **300** is on the enterprise network **806**, where associated applications reside. Of course, any malware on the user device **300** or anyone that steals the user device **300** is also on the enterprise network **806**. The VPN termination device **804** creates a Distributed Denial-of-Service (DDoS) attack surface, adds infrastructure cost and creates network complexity as applications grow. Conversely, in FIG. **12**, the user device **300** uses the virtual private access via the cloud-based system **100** to connect to the lightweight connector **400** associated with a specific application **404**. The virtual private access provides granular access by the user device **300** and the application, and the user device **300** is not on the enterprise network **806**. Thus, the application is never directly exposed to the user device **300**, the security cloud handles provisioning, and the traffic remains completely private.

Comparison—Private Applications in the Public Cloud

[0102] FIGS. **13** and **14** are network diagrams of conventional private application access in the public cloud (FIG. **13**) compared to private application in the public cloud with virtual private access (FIG. **14**). In FIG. **13**, the user device **300** still has to connect to the enterprise network **806** via the VPN termination device **804** as in FIG. **11**, and the cloud applications, such as in the data center **610**, are accessible via the enterprise network **806** via a site-to-site VPN between the enterprise network **806** and the data center **610**. Disadvantageously, the user experience is eroded for the user device **300** and agility is hampered for the enterprise by networking concerns and capability. In FIG. **14**, the virtual private access abstracts the application **402**, in the data center **610**, from the IP address, so location is irrelevant. The enterprise can move private applications to the cloud securely, as needed.

Comparison—Contractor/Private Application Access

[0103] FIGS. **15** and **16** are network diagrams of conventional contractor/partner access (FIG. **15**) of applications in the enterprise network **806** compared to contractor/partner access (FIG. **16**) of the applications with virtual private access. Contractor/partner access includes providing third parties access to applications on the enterprise network **806**, for a variety of purposes. In FIG. **15**, similar to FIGS. **11** and **13**, contractor/partner access includes VPN connections to the VPN termination device **804**, providing contractor/partners **820** full access to the enterprise network **806**, not just the specific application or asset that they require. Unfortunately, stolen credentials can allow hackers to get into networks or to map assets for later assault. In FIG. **16**, the virtual private access, using the cloud-based system **100**, allows access specific to applications or assets as needed by the contractor/partners **820**, via the lightweight connector **400**. Thus, the contractor/partners **820** do not have full network access, the access is specific to each user, and the connections are provisioned dynamically, avoiding a direct network connection that can be misused or exploited.

Comparison—Example Application—M&A Data Access

[0104] FIGS. **17** and **18** are network diagrams of a conventional network setup to share data between two companies (FIG. **17**) such as for Merger and Acquisition (M&A) purposes or the like, compared to a network setup using virtual private access (FIG. **18**). Conventionally, the two companies provide VPN connections between their associated enterprise networks **806**A, **806**B to one another. Each company gets "all or nothing"—no per-application granularity. Disadvantageously, creating Access Control Lists (ACLs)/firewall rules and NATting through each companies' respective firewalls is very complex, particularly with overlapping internal IP addressing. In FIG. **18**, the virtual private access allows connections provisioned by the user and device to the application by name, not by IP address, authorized users can access only specific applications, not an entire network, and firewall complexities disappear.

Administrative View of Virtual Private Access

[0105] FIGS. **19** and **20** are screenshots of Graphical User Interfaces (GUIs) for administrator access to the virtual private access. FIG. **19** illustrates a GUI of network auto-discovery and FIG. **20** illustrates a GUI for reporting. For network and application discovery, the virtual private access can use wildcard application discovery where a Domain/name-based query to the lightweight connector **400** will

show company applications behind them. This allows the discovery of internal applications as users request them using "*.company.com" to find applications. Then, the granular policy can be built around the applications to dramatically simply startup. Further, the virtual private access can show the location of users that are accessing private/internal applications, including identifying anomalous access patterns to assist in stopping possible data leakage or compliance violation.

Virtual Private Access

[0106] In an embodiment, a virtual private access method implemented by a cloud-based system, includes receiving a request to access resources from a user device, wherein the resources are located in one of a public cloud and an enterprise network and the user device is remote therefrom on the Internet; forwarding the request to a central authority for a policy look up and for a determination of connection information to make an associated secure connection through the cloud-based system to the resources; receiving the connection information from the central authority responsive to an authorized policy look up; and creating secure tunnels between the user device and the resources based on the connection information. Prior to the receiving, a user executes an application on the user device, provides authentication, and provides the request with the application operating on the user device. The application can be configured to connect the user device to the cloud-based system, via an optimized cloud node based on a location of the user device. The resources can be communicatively coupled to a lightweight connector operating on a computer and communicatively coupled between the resources and the cloud-based system. The virtual private access method can further include detecting the resources based on a query to the lightweight connector. The lightweight connector can be prevented from accepting inbound connections, thereby preventing access of the resources external from the public cloud or the enterprise network. The creating secure tunnels can include creating connections between one or more cloud nodes in the cloud-based system, wherein the one or more cloud nodes do not participate in a key exchange, and the one or more cloud nodes do not have data access to traffic on the secure tunnels. The creating secure tunnels can include creating connections between one or more cloud nodes in the cloud-based system, wherein the one or more cloud nodes create the secure tunnels based on a combination of a client-side certificate and a server-side certificate. The secure tunnels can be created through software on the user device, the cloud-based system, and a lightweight connector operating on a computer associated with the resources, thereby eliminating dedicated hardware for virtual private network connections.

[0107] In another embodiment, a cloud-based system adapted to implement virtual private access includes one or more cloud nodes communicatively coupled to one another; wherein each of the one or more cloud nodes includes one or more processors and memory storing instructions that, when executed, cause the one or more processors to receive a request to access resources from a user device, wherein the resources are located in one of a public cloud and an enterprise network and the user device is remote therefrom on the Internet; forward the request to a central authority for a policy look up and for a determination of connection information to make an associated secure connection through the cloud-based system to the resources; receive the connection information from the central authority responsive to an authorized policy look up; and create secure tunnels between the user device and the resources based on the connection information. Prior to reception of the request, a user executes an application on the user device, provides authentication, and provides the request with the application operating on the user device. The application can be configured to connect the user device to the cloud-based system, via an optimized cloud node based on a location of the user device. The resources can be communicatively coupled to a lightweight connector operating on a computer and communicatively coupled between the resources and the cloud-based system. The memory storing instructions that, when executed, can further cause the one or more processors to detect the resources based on a query to the lightweight connector. The lightweight connector can be prevented from accepting inbound connections, thereby preventing access of the resources external from the public cloud or the enterprise network. The secure tunnels can be created through connections between one or more cloud nodes in the cloud-based system, wherein the one or more cloud nodes do not participate in a key exchange, and the one or more cloud nodes do not have data access to traffic on the secure tunnels. The secure tunnels can be created through connections between one or more cloud nodes in the cloud-based system, wherein the one or more cloud nodes create the secure tunnels based on a combination of a client-side certificate and a server-side certificate. The secure tunnels can be created through software on the user device, the cloud-based system, and a lightweight connector operating on a computer associated with the resources, thereby eliminating dedicated hardware for virtual private network connections.

[0108] Software stored in a non-transitory computer readable medium including instructions executable by a system, which in response to such execution causes the system to perform operations including receiving a request to access resources from a user device, wherein the resources are located in one of a public cloud and an enterprise network and the user device is remote therefrom on the Internet; forwarding the request to a central authority for a policy look up and for a determination of connection information to make an associated secure connection through the cloud-based system to the resources; receiving the connection information from the central authority responsive to an authorized policy look up; and creating secure tunnels between the user device and the resources based on the connection information. The resources can be communicatively coupled to a lightweight connector operating on a computer and communicatively coupled between the resources and the cloud-based system, and wherein the instructions executable by the system, which in response to such execution can further cause the system to perform operations including detecting the resources based on a query to the lightweight connector.

VPN in the Cloud

[0109] In an embodiment, a method includes connecting to a client at a Virtual Private Network (VPN) device in a cloud-based system; forwarding requests from the client for the Internet or public clouds accordingly; and for requests for an enterprise associated with the client, contacting a topology controller to fetch a topology of the enterprise, causing a tunnel to be established from the enterprise to the

VPN device, and forwarding the requests for the enterprise through the tunnel to the cloud-based system for proactive monitoring; and providing a secure connection from the cloud-based system back to the enterprise, including internal domain and subnets associated with the enterprise. The method can further include authenticating, via an authentication server, the client prior to the connecting and associated the client with the enterprise. The method can further include, subsequent to the connecting, setting a Domain Name Server (DNS) associated with the cloud-based system to provide DNS lookups for the client. The method can further include utilizing the DNS to determine a destination of the requests; and, for the requests for the enterprise, contacting the topology controller to pre-fetch the topology of the enterprise. The method can further include operating an on-premises redirection proxy within the enterprise, wherein the on-premises redirection proxy is configured to establish the tunnel from the enterprise to the VPN device. Secure tunnels to the enterprise are dialed out from the enterprise by the on-premises redirection proxy. The on-premises redirection proxy is a virtual machine operating behind a firewall associated with the enterprise. The on-premises redirection proxy is configured as a bridge between the client and applications inside the enterprise. The VPN device operates on a cloud node in the cloud-based system, and wherein the cloud-based system includes a distributed security cloud. The VPN device can include one of a software instance on a cloud node or a virtual machine on the cloud node. The topology controller includes a network topology of the enterprise, including internal domain names and subnets.

[0110] In another embodiment, a cloud-based system includes one or more Virtual Private Network (VPN) servers, wherein one or more clients connect securely to the one or more VPN servers; a topology controller communicatively coupled to the one or more VPN servers; a Domain Name Server (DNS) communicatively coupled to the topology controller and the one or more VPN servers; and a redirection proxy located in a private network and communicatively coupled to the one or more VPN servers and the topology controller; wherein requests from the one or more clients to the private network cause on demand secure connections being established by the redirection proxy to associated VPN servers in a cloud-based system, wherein the on demand secure connections provide connectivity to the private network including internal domain and subnets associated with the private network, and wherein the cloud-based system performs proactive monitoring. Requests from the one or more clients outside of the private network are forwarded without traversing the private network. The redirection proxy maintains a persistent connection to the topology controller and establishes secure tunnels to the one or more VPN servers based on direction from the topology controller. The topology controller includes a network topology of the private network, including internal domain names and subnets. The VPN servers operate on cloud nodes in a distributed security cloud.

[0111] In yet another embodiment, a VPN system includes a network interface, a data store, and a processor, each communicatively coupled together; and memory storing instructions that, when executed, cause the processor to establish a secure tunnel with a client; forward requests from the client to the Internet accordingly; and for requests to an enterprise, contact a topology controller to fetch a topology of the enterprise, cause a tunnel to be established from the enterprise to the VPN system, and forwarding the requests for the enterprise through the tunnel and the secure tunnel, wherein the secure tunnel is achieved by using an on-demand dial-out and tunneling traffic authentication. The memory storing instructions that, when executed, further cause the processor to cause the tunnel to be established from the enterprise to the VPN system through an on premises redirection proxy located within the enterprise.

Browser Isolation

[0112] Browser (web) isolation is a technique where a user's browser or apps are physically isolated away from the user device, the local network, etc. thereby removing the risks of malicious code, malware, cyberattacks, etc. This has been shown to be an effective technique for enterprises to reduce attacks. Techniques for browser isolation are described in commonly-assigned U.S. patent application Ser. No. 16/702,889, filed Dec. 4, 2019, and entitled "Cloud-based web content processing system providing client threat isolation and data integrity," the contents of which are incorporated by reference herein. Traditionally browser isolation was focused on removing the risks of malicious code, malware, cyberattacks, etc. U.S. patent application Ser. No. 16/702,889 describes an additional use case of preventing data exfiltration. That is, because no data is delivered to the local system (e.g., to be processed by web content through the local web browser), none of the confidential or otherwise sensitive data can be retained on the local system.

[0113] The secure access can interoperate with browser isolation through the cloud-based system **100**, to prevent data exfiltration, which is extremely critical as this is customer-facing data which adds to the sensitivity and liability, and also accessible to external users (customers). This functionality forces customers to interact with the B2B applications via an isolated, contained environment.

Private Service Edge in a Cloud-Based System

[0114] FIG. **21** is a network diagram of the cloud-based system **100** with a private service edge node **150**P in the enterprise network **410**. The private service edge node **150**P is similar to the enforcement nodes **150** (i.e., public service edge nodes) except located in the enterprise network **410**. For private application access, the service edge node **150**P can be a broker that is hosted by the enterprise, but managed with the cloud-based system **100**. As described herein, a broker is configured to create the tunnels between the user device **300** and the connector **400**, and the broker is an intermediate device. The service edge node **150**P is designed as a single-tenant (per customer) instance, is configured to operate with the cloud-based system **100** including downloading policies and configuration, is configured to broker connections between the connector application **350** and the connector **400**, is configured to enforce policies and cache path selection decisions, etc.

[0115] When a user **102** with the user device **300** is located on the enterprise network **410**, the traffic between the user **102** and the applications **404** stay on the enterprise network **410** and consistent policies are applied for on-premise and remote. The private service edge node **150**P can be located in a branch office, in a central office with tunnels to branch offices, etc. Of note, the private service edge node **150**P is

located with the applications **404** and the connector **400** and this proximity reduces latency.

[0116] The private service edge node **150**P can be hosted in a public cloud, on-site as a Virtual Machine (VM), in a container, on physical servers, etc. The private service edge node **150**P is publicly accessible such as via an IP address; the connector **400** is not publicly accessible—it dials out. The private service edge node **150**P can include listen IP addresses and publish IP addresses or domains. The listen IP addresses are a set of IP addresses that the private service edge node **150**P uses for accepting incoming connections, and this can be specified or all IP addresses. The publish IP addresses or domains, if specified, are required for connection to the private service edge node **150**P. If these are specified, one of the entries is provided to the applications **350**, e.g., randomly selected.

Active Directory Protection

[0117] The present disclosure provides a feature for Active Directory Protection (ADP) inside zero trust networks by auto-detecting Active Directory (AD) protocols (i.e., KRB, LDAP, SMB, etc.) and inspecting the protocol messages to log necessary information for possible threat hunting. A generic inspection pipeline for both TCP and UDP connections, along with auto-detection and inspection of protocol traffic is contemplated for visualization and analytics for threat hunting is provided by the present ADP feature.

[0118] AD protocols can be prone to various types of attacks which can bring down domain controllers. A plurality of types of protocol traffic can be seen on domain services, mainly Light-weight Directory Access Protocol (LDAP), Server Message Block (SMB), and Kerberos (KRB). The AD infrastructure is built as a centralized system that controls the entire Information Technology (IT) infrastructure, with access to applications, software, sensitive files, and confidential data which is located in the AD. This makes the AD a popular location for an attacker to breach security and the network. Connectivity to the domain controller is via an inside out connection initiated by a service that is also capable of decrypting and decoding all traffic destined to the domain controller. The ADP feature described herein detects and inspects the protocol traffic to generate logs which can be used for threat hunting, isolating attackers, and early detection and mitigation.

[0119] Currently, there is no solution for ADP within zero trust networks. Traditional AD threat hunting is based on identifying different events related to activities on AD servers and clients using Antimalware Scan Interface (AMSI) tools.

[0120] The virtual private access systems described herein support HTTP/S and WebSocket inspection. Customers have shown interest in active directory traffic inspection to prevent threats to identity and active directory infrastructure, DNS infrastructure, etc. The ADP inspection feature is designed to support this requirement. The present disclosure describes the architecture and design of the Intrusion Detection and Prevention System (IDPS) pipeline for AD which includes Light-weight Directory Access Protocol (LDAP), Server Message Block (SMB), and Kerberos (KRB) traffic inspection. These protocols are commonly manipulated by attackers to perform network enumeration and exploitation.

[0121] With the introduction of the ADP Feature, the private access systems have support for inspecting AD protocols, mainly LDAP, SMB, and KRB traffic. The feature is enabled globally for a customer or tenant using an AD protection feature flag in a configuration portal. With the AD Protection feature enabled for a tenant, ZPA administrators have provision to configure application segments with AD inspection. The ADP feature can be selectively configured for each tenant of the cloud-based system on a per-tenant basis, and selectively configured within tenant environments, such as for individual users or groups. For customer ease of configuration, the ADP feature can be enabled globally within an application segment, on which all domains configured will have inspection enabled by default for LDAP, SMB, and KRB on various ports.

[0122] The private access systems can achieve AD protection with IDSP infrastructure which provides respective protocol pipelines to detect various protocols and process them to collect required information. The information can be processed by elastic search to identify possible threats in AD domains.

[0123] A Protocol Auto Detection and Processing (PDP) Library provides an infrastructure for retrieving required protocol information for analytics to an inference engine or threat hunting engine to show trend charts for alerting administrators. The PDP maintains profiles/context for optimized auto detection by saving profiles in memory for the lifetime of the system. The diagnostic logs for analytics and creating trend charts are rendered based on ADP logs which are generated uniquely. The graph plotted for each protocol can be filtered based on users, protocol errors, message types, LDAP queries, SMB enumerations, etc. for a specific time interval. For identifying threats, the systems can uncover suspicious behaviors and activities via unusual counters, unused or invalid data, unusual number of requests with more errors than success with respect to normalized behavior in the domain services, etc.

[0124] FIG. **22** is a flow diagram of main connector or data plane components for IDPS AD inspection. On receiving a request from the data broker **2202**, the client connector **2204** creates an Mtunnel with the IDPS pipeline **2206** for LDAP, SMB, or KRB inspection based on domain and server port configured in the AD protection enabled application segment. The IDPS pipeline **2206** detects and inspects the traffic from client to server and vice versa using a PDP library **2208** which processes different protocol messages. This information gathered by the PDP library **2208** is updated to the cloud-based system **100** via IDPS Logs. The traffic on the IDPS pipeline **2206** can be categorized as LDAP, SMB, KRB, or Unclassified.

[0125] FIGS. **23-25** are screenshots of a UI for configuring the present ADP feature and reviewing trends. In various embodiments, a User Interface (UI) can include a provision for enabling or disabling ADP per application segment. With ADP enabled, all configured domains will have the inspection enabled for LDAP, SMB, and KRB. In an embodiment, regular HTTP/S and WebSocket inspection is only supported on application segments where ADP is disabled. Additional default ports can be added using a drop down "default port ranges" list which enables the respective inspection for designated protocols. Custom ports can be configured for inspection by adding ports and configuring required inspection protocols. Custom domains can disable one or more AD protocol inspections by deselecting the LDAP, SMB, or KRB selector. In an embodiment, for inspection profiles and rules, an empty inspection profile is created for ADP. All anomalies detected and protocol sta-

tistics can be plotted as timeline trends on the UI for administrators to analyze further. Such anomalies can include unusual counters, unused or invalid data, unusual number of requests with more errors than success with respect to normalized behavior in the domain services, etc. An inspection policy is applied for an AD application segment or segment group. If any non-AD application segment or segment group is added to the same inspection policy, an error is presented indicating that the inspection policy will not apply for AD traffic. The UI is adapted to provide a dashboard which includes IDPS logs instead of inspection logs generated for each (c_uid, domain, port) tuple in 15 second periodic intervals. It will be appreciated that the 15 second interval is configurable by administrators to be any length of time. Logging and analytics of IDPS logs logged by different connectors for each protocol (i.e., LDAP, SMB, and KRB) with respect to {c_uid, domain, port} tuple are used to draw IDPS trends or timeline graphs which plot relevant fields indicating a possible anomaly. A sample log is shown in FIG. **25**.

[0126] FIG. **26** is a flow diagram of the present ADP feature. In various embodiments, users/administrators are able to configure AD application and inspection configurations **2602**, i.e., via the UI **2604**. Such configurations can include ADP feature flag configuration, AD application segment configuration, multi-port multi-protocol AD domain inspection configuration, and inspection policy/access policy configuration. These configurations can be stored in a configuration store **2606**. Incoming traffic is fed through the ADP feature based on the AD inspection profile **2608**. Different PDP metadata is processed in a detection stage **2610** to detect and classify the traffic as any of LDAP, DMSB, KRB, or any other AD traffic known to one of skill in the art. The detection stage can provide intermediate messages indicating process failure, classification in progress, or the detected traffic classification. An inspection stage **2612** includes a protocol specific parser to provide status updates associated with the PDP metadata. The systems can further include a PDP profile cache **2614** which takes domain and port as inputs, and provides the traffic classification, detected protocol, and protocol inspection details as outputs. These outputs are processed and aggregated to create an ADP log which is provided to an ADP logging and analytics engine **2616**. These analytics are then presented in the form of various graphical representations in the UI **2604** for users/administrators to further inspect. These graphical representations can include trend charts **2618** for identifying suspicious activity and the like.

[0127] Various embodiments of the present ADP feature are adapted to provide global configuration to enable the ADP feature. The detection and inspection stages can be adapted to support generic infrastructure for any protocol. The multi-stage inspection pipeline can include protocol detection, protocol message processing, and/or deep packet inspection. Smart inspection and analysis of AD traffic can detect and protect AD networks from threats, security vulnerabilities, uncover unusual user activities such as Service Principal Name (SPN) scanning, user hunting, password spraying, credential dumping, etc. The process is adapted to generate various AD trend visualizations, such as graphical timeline charts, indicating anomalies in various streams. The present feature provides support for default protocol ports including the following.

Protocol	Ports
KRB	88(tcp/udp), 464(tcp/udp)
LDAP	389(tcp/udp)
SMB	445(tcp/udp)

[0128] As stated, logs are generated for a tuple at preconfigured time intervals when traffic is detected for the tuple. This is streamed and analyzed for diagnostics trends, timeline chart plotting, and thereby threat intelligence. In an embodiment, if traffic detection is not successful within a configured number of attempts, the inspection stage is bypassed, and an unclassified log is generated. Each log contains information specific to protocols, relevant to identifying attackers, malicious user activity, etc.

[0129] Again, the PDP library is a module for detecting and inspecting multiple protocols. The PDP library maintains the profile for every {domain, port} tuple. An entry is created when a stream is processed for a specific {domain, port} tuple for the first time. Protocol classification is saved in the profile for optimization in case of auto-detection of the protocol. Each stream maintains PDP metadata which includes configuration data, operational data, and status of the protocol processed. Configuration data can include respective domain, server port, transport (TCP/UDP), and pipeline protocol configuration. Operation data includes profile protocol which is the previously detected protocol classification for the {domain, port} tuple, traffic classification, and protocol classification. A status can include the default/well-known protocol port, sub-protocol categories, detection attempts, expected requested length, expected response length, and protocol specific status information. The PDP contains a detection stage and a processing/inspection stage. The detection stage can include the capability to auto-detect supported protocols based on well-known/common ports and payload format inspection. The inspection stage inspects the detected protocol payloads to collect details such as success, errors, types of errors, and protocol information for generating the logs.

[0130] Logs are generated from the information gathered in 15 second intervals, or other preconfigures time interval. This is consumed to plot the diagnostics trends timeline graph for the various protocols, thereby deducing possible attack threats.

Process for Active Directory Protection

[0131] FIG. **27** includes a flowchart of a process **2700** for active directory protection. The process **2700** includes performing inline monitoring of traffic associated with a cloud-based system (step **2702**); detecting one or more active directory protocols based on the inline monitoring (step **2704**); classifying the traffic as being associated with any of the one or more active directory protocols (step **2706**); inspecting the traffic associated with the one or more detected active directory protocols (step **2708**); and generating one or more active directory logs based on the inspecting and classifying (step **2710**).

[0132] The process **2700** further includes wherein the generating includes generating a plurality of active directory logs, wherein each of the plurality of active directory logs is associated with one of the one or more active directory protocols. The steps can further include generating one or more active directory trend visualizations based on the

active directory logs. The steps can further include providing the one or more active directory trend visualizations via a User Interface (UI). The one or more active directory trend visualizations can include a trend timeline chart. The one or more active directory trend visualizations can include a trend timeline chart associated with each of the one or more active directory protocols. The detecting can include automatically detecting active directory protocols based on ports and payload format inspection. The inspecting can include uncovering anomalies associated with the traffic associated with the one or more active directory protocols. The steps can further include receiving one or more active directory protection configurations prior to the monitoring, wherein the one or more active directory protection configurations specify one or more application segments or segment groups for performing the steps thereon. The one or more active directory protocols can include Light-weight Directory Access Protocol (LDAP), Server Message Block (SMB), and Kerberos (KRB).

Active Directory Security Enforcement

[0133] Active Directory (AD) serves as a cornerstone technology for many global enterprises, providing a centralized framework for managing users, devices, applications, and other organizational objects. It is especially valuable for organizations operating across multiple geographical locations, as it ensures seamless integration and governance of resources. ADs critical role in identity and access management makes it a prime target for cyber adversaries seeking to exploit its vulnerabilities. In recent years, the AD threat landscape has evolved dramatically, with attackers employing increasingly sophisticated methods to compromise individual identities and steal credentials. These tactics enable unauthorized access to sensitive applications, systems, and critical data, often causing significant operational and reputational damage.

[0134] As organizations recognize these risks, a growing number of AD security solutions have emerged to safeguard these environments. Leading endpoint security tools, including CrowdStrike Falcon, AD AUDIT, ManageEngine, and Antimalware Scan Interface (AMSI) tools such as Windows Defender, McAfee, and Kaspersky, are designed to detect, prevent, and respond to a range of AD threats. These tools offer advanced monitoring, alerting, and mitigation capabilities that help protect against common attack vectors targeting AD systems.

[0135] Some of the most frequently observed attacks include Kerberoasting, where attackers exploit weak service account configurations to extract and crack service account credentials; AS-REP Roasting, which targets accounts not requiring pre-authentication to obtain credential hashes; and account enumeration, where attackers probe AD for valid user accounts. Additionally, legacy protocols such as Server Message Block version 1 (SMBv1) traffic and improper Lightweight Directory Access Protocol (LDAP) querying are often used by attackers to identify misconfigurations and vulnerabilities within an organization's AD setup.

[0136] The effectiveness of these attacks underscores the importance of implementing robust endpoint security measures. When AD-specific security solutions are deployed, they can effectively monitor attack signatures, block malicious activities in real-time, and alert administrators to potential threats. For example, Kerberoasting and AS-REP Roasting attempts can be detected through monitoring unusual authentication patterns, while tools can flag inappropriate LDAP queries or SMBv1 traffic. By integrating these tools into their security strategy, organizations can proactively reduce the attack surface of their AD environments and ensure the integrity and confidentiality of their critical assets.

[0137] As the sophistication of cyber threats continues to grow, the need for a proactive and layered security approach becomes paramount. Organizations must continuously evaluate their AD configurations, patch vulnerabilities, and implement security best practices alongside these tools to stay ahead of potential adversaries. This not only fortifies their defenses but also ensures compliance with regulatory standards and protects their business operations from the far-reaching impacts of a security breach.

[0138] Currently, the market lacks a dedicated solution for protecting AD within a zero trust network framework. While zero trust principles emphasize robust identity verification and continuous monitoring, most available solutions focus exclusively on securing endpoints, such as laptops, desktops, and mobile devices. These endpoint-focused tools, including antivirus software, Endpoint Detection and Response (EDR) systems, and other security measures, are designed to prevent threats at the user and device level. However, they do not address the need for proactive security measures that directly protect AD environments at the network level.

[0139] The present invention enables administrators to proactively configure AD security controls and mitigate potential threats within a zero trust environment. By enabling the blocking of suspicious activities originating from AD domain users, this solution prevents malicious or unauthorized actions from progressing to critical infrastructure components such as AD servers and domain controllers.

[0140] The zero trust network model assumes that threats can originate from both inside and outside the organizational perimeter, emphasizing continuous validation of user identities and device trust. This invention aligns with these principles by acting as a protective intermediary, scrutinizing and intercepting dubious activities before they pose a risk to the AD environment. This preemptive approach significantly enhances organizational security by minimizing the likelihood of compromise to the central AD infrastructure, which is often targeted for its critical role in identity and access management.

[0141] By leveraging advanced security controls and real-time monitoring, this invention supports administrators in enforcing stringent policies and safeguarding sensitive organizational resources. It complements the broader zero trust strategy by ensuring that no action, user, or device is implicitly trusted, thereby fortifying the AD ecosystem against emerging cyber threats.

[0142] In various embodiments, this invention, i.e., the Active Directory Protection (ADP) system is composed of two primary functions. The first function provides security enforcement by utilizing AD controls to manage and mitigate vulnerabilities associated with specific protocol messages and versions. It achieves this by blocking or alerting on various security threats. For instance, it can block or alert when a Kerberos Ticket Granting Service (TGS) request uses weak cipher encryption, which is a significant security risk. Additionally, it can detect and respond to highly vulnerable Server Message Block version 1 (SMBv1) traffic, thereby preventing potential exploitation through this out-

dated protocol. Another critical feature is its ability to block or alert on Lightweight Directory Access Protocol (LDAP) requests that lack an authentication blob, which can indicate unauthorized access attempts.

[0143] The second function of the invention focuses on providing threat insights related to AD attacks by identifying specific attack signatures. It monitors for signs of Kerberos account harvesting or enumeration activities, which are often precursors to more sophisticated attacks. It also detects Kerberoasting, a technique used by attackers to extract service account credentials from Kerberos tickets. Additionally, the system can identify AS-REP roasting attempts, where attackers exploit accounts not requiring pre-authentication to obtain password hashes. Beyond Kerberos-specific threats, the invention also scrutinizes suspicious LDAP activities, which could indicate unauthorized directory access or modifications. It looks for anomalous behaviors in Server Message Block (SMB) traffic and Distributed Computing Environment/Remote Procedure Calls (DCERPC) over SMB, both of which could signify attempts to exploit these protocols for malicious purposes. Through these comprehensive threat detection capabilities, the invention significantly enhances the security posture of systems relying on AD.

[0144] The current ADP system operates within a zero trust network framework of the cloud-based system **100**, which significantly enhances its scalability and adaptability for any AD environment, regardless of its geographic distribution. This approach is particularly beneficial for global AD environments, as it allows the inspection and protection mechanisms to be uniformly and effectively applied across diverse and widely dispersed network segments. The scalability of this solution means it can seamlessly accommodate the expansion of AD infrastructures, ensuring robust security measures are consistently maintained as the organization grows and evolves. Consequently, this system provides a resilient and flexible security posture that is well-suited for contemporary, geographically distributed AD environments.

[0145] In various embodiments, this invention conducts real-time, live inspection of Kerberos, LDAP, SMB, and DCERPC traffic, meticulously analyzing network traffic to identify and derive information pertinent to potential attack signatures. The system continuously monitors these protocols, enabling it to detect threats and suspicious activities as they occur. When such activities are identified, administrators are promptly alerted through intuitive dashboards provided by the cloud-based system **100**, which also includes diagnostic tools to facilitate a deep dive into the alerts.

[0146] These dashboards not only notify admins of potential security issues but also offer actionable insights, allowing them to respond swiftly. That is, the system can provide remediation steps to mitigate a potential attack based on detecting one or more attack signatures in real-time. Administrators, as well as the system, have the capability to block suspicious activities in real-time through predefined AD security controls integrated within the system. This proactive approach ensures that potential threats are mitigated immediately, preventing further exploitation.

[0147] Moreover, the invention extends its functionality by supporting the streaming of inspection logs for Kerberos, LDAP, and SMB protocols to Security Information and Event Management (SIEM) monitoring servers within customer environments. This feature allows for comprehensive analytics and analysis beyond the immediate detection and response capabilities of the system. By integrating with SIEM solutions, the invention enables organizations to leverage advanced analytical tools for deeper investigation, long-term monitoring, and correlation of security events, thereby enhancing their overall security posture.

[0148] FIG. **28** is a flow diagram of the present Active Directory Protection (ADP) system. The application **350** plays a crucial role in extending security and connectivity features to end users, ensuring that their interactions with the network are secure and compliant with corporate policies. The application **350** ensures that users can securely access applications, including those that are part of the AD domain. By routing traffic through the zero trust exchange, i.e., the cloud-based system **100**, the application **350** enforces security policies and inspections on the traffic, protecting against threats and ensuring compliance with security protocols. The application **350** helps in directing user traffic to ADP-enabled application segments where it can be inspected for security threats. This includes inspecting Kerberos, LDAP, SMB, and DCERPC over SMB protocols to detect and mitigate potential attacks. The application **350** ensures that all traffic is subject to the necessary security controls and policies defined in the ADP system **2800**.

[0149] The application **350** facilitates secure user authentication and authorization processes by integrating with the AD. This ensures that only authenticated and authorized users can access sensitive resources, thereby enhancing the security posture of the AD environment. By deploying the application **350** on user devices, the cloud-based system **100** can enforce security policies directly at the endpoints. This includes applying ADP policies, which helps in preventing unauthorized access and mitigating risks posed by compromised devices or malicious activities originating from user endpoints.

[0150] The application **350** contributes to real-time threat detection and response by ensuring that traffic from user devices is continuously monitored and inspected. Any detected threats are immediately acted upon as per the predefined ADP security controls, such as blocking, alerting, or logging the suspicious activities. The application **350** integrates seamlessly with the cloud-based system, leveraging its cloud-native architecture to provide scalable and reliable security services. This integration ensures that ADP functionalities are consistently applied, regardless of user location, whether they are on-premises or remote. While providing robust security, the application **350** is designed to minimize impact on user experience and performance. It ensures that security inspections and policy enforcement do not degrade the performance of applications or the overall user experience.

[0151] The broker **2802** plays a crucial role in the ADP system **2800** by evaluating the inspection policies and profiles applied to ADP-enabled application segments. It transmits the inspection policy and profile information to the connector **400** via a zpn_broker_request. The connector **400** then retrieves the inspection applications and profiles from Wally **2804** and integrates them into the inspection pipeline. The broker **2802** is also responsible for collecting logs generated by the connector **400**, which includes ADP logs, and forwarding them to the respective customer logging zone's for streaming or further processing.

[0152] In various embodiments, "Wally" **2804** is a key component responsible for managing and distributing the

security controls and inspection profiles necessary for the ADP process. Wally **2804** retrieves ADP controls from inspection profiles and control tables stored in a Relational Database Service (RDS). These controls define the specific rules and configurations used to inspect and protect AD traffic. It ensures that both brokers **2802** and connectors **400** are updated with the latest configuration changes. By doing so, it maintains the accuracy and effectiveness of the inspection policies and security controls applied across the network. It also plays a vital role in managing the configurations related to ADP. It ensures that any updates or changes to the inspection profiles and security controls are promptly communicated to the relevant components (brokers **2802** and connectors **400**). This continuous update process helps in adapting to new threats and maintaining a robust security posture.

[0153] By acting as a central repository and distribution point for ADP controls, Wally **2804** simplifies the management of security policies. It ensures that all components involved in the ADP process are synchronized and operating with the most current and comprehensive set of rules. Its ability to efficiently manage and distribute security controls contributes to the scalability and reliability of the ADP system **2800**. It ensures that even as the network expands or changes, the security measures remain consistent and effective.

[0154] Within the ADP system **2800**, the connector **400** plays a fundamental role in implementing and enforcing security measures to protect AD environments. Its functions are multi-faceted and integral to the overall security architecture. The following outlines key functions of the connector **400** in the ADP system.

[0155] Downloading inspection profiles: The connector **400** begins by downloading the AD inspection profiles of a tenant from Wally, or any other service of the like. These profiles contain predefined security rules and configurations that are essential for monitoring and protecting AD traffic.

[0156] Building inspection profiles: Once the profiles are downloaded, the connector **400** compiles them into a comprehensive ADP inspection profile. This profile includes one or more AD security rules tailored to address specific threats and vulnerabilities.

[0157] Traffic inspection and control application: The connector **400** inspects traffic for protocols such as Kerberos (KRB), LDAP, and SMB. Using the protocol detection and inspection library, it performs Deep Packet Inspection (DPI) to identify the protocol by examining the byte stream. This allows the connector **400** to detect and decode packets accurately, extracting relevant information to identify potential AD attacks. Based on the inspection, controls are applied according to the criteria specified in the security rules, determining whether to allow or block the traffic.

[0158] Real-time threat detection: The connector **400** operates in real-time, continuously monitoring network traffic to detect and respond to threats. When suspicious or malicious activities are identified, the connector **400** can take immediate action to block the traffic, preventing potential breaches or exploits.

[0159] Log generation and reporting: For every inspected tunnel, the connector **400** generates an ADP inspection log. These logs provide detailed records of the inspection process and any actions taken, such as blocking traffic. This information is crucial for security auditing and analysis.

[0160] Synthetic response generation: When the connector **400** blocks traffic, it can generate synthetic responses to inform the requesting entity of the action. For example: For Kerberos, it creates a KRB_ERROR message with a custom reason.

[0161] For LDAP, it produces an LDAP message with either protocol error or operations error.

[0162] For SMB, it generates an SMB access denied message with a custom notification.

[0163] Integration with logging systems: The logs generated by the connector **400** are forwarded to control brokers, which then stream or forward these logs to producers in the respective customer logging zones. This ensures that all inspection activities and security events are documented and can be further analyzed using Security Information and Event Management (SIEM) systems or other monitoring tools.

[0164] In summary, the connector **400** is a critical component in the ADP system **2800**, responsible for downloading and building inspection profiles, performing real-time traffic inspection, applying security controls, generating logs, and integrating with logging systems. Its role ensures that AD environments are continuously monitored and protected from a wide range of security threats.

[0165] In an example use case, the process begins when a user authenticates with AD domain services using the Kerberos (KRB) protocol, obtaining a valid Kerberos Ticket Granting Ticket (KRB-TGT). With this ticket, the user then requests access to services hosted by the AD domain, utilizing a Kerberos service ticket received through a Kerberos Ticket Granting Service (KRB-TGS) request. Once the user has both the Kerberos user ticket and service ticket, they can proceed to perform various operations such as running LDAP queries, accessing SMB files, and executing other service-related tasks.

[0166] All this AD traffic is managed through Zscaler Private Access (ZPA) application segment configurations, which specify the TCP/UDP port ranges and the necessary access control settings. When the ADP feature is enabled, the connectors **400** inspect the Kerberos, LDAP, and SMB traffic to extract information relevant to potential threat signatures. These connectors then scan the extracted information against the ADP security profile applied to the application segment, determining whether to allow or block the traffic based on the analysis.

[0167] As part of this process, an ADP log is generated, detailing the inspection controls triggered by the AD tunnel. This log is then propagated from the connector through a broker to the logging zone using ARGO serialization, ensuring that comprehensive records of the traffic inspection and security actions are maintained for further analysis and auditing. This systematic approach ensures robust security enforcement and detailed monitoring of AD traffic, enhancing the overall protection of the AD environment.

[0168] As described, the ADP system **2800** offers administrators the ability to configure specific AD controls within security profiles. These security profiles are comprehensive collections of rules and settings designed to protect AD environments from various threats. Once configured, these security profiles can be associated with specific security policies, which dictate how the rules are enforced across the network. Furthermore, the security profiles are linked to application segments, which define the particular applications and services within the network that the security

controls will apply to. This seamless integration ensures that the appropriate security measures are consistently applied to the relevant areas of the network, enhancing the overall security posture and providing administrators with granular control over their AD security configurations.

Active Directory Security Enforcement Process

[0169] FIG. **29** includes a flowchart of a process **2900** for active directory security enforcement. The process **2900** includes performing inline monitoring of traffic associated with a plurality of tenants of the cloud-based system (step **2902**); classifying the traffic as being associated with any of one or more active directory protocols (step **2904**); inspecting the traffic associated with the one or more detected active directory protocols (step **2906**); and performing one or more actions on the traffic based on the inspecting (step **2908**).

[0170] The process **2900** can further include wherein the inline monitoring includes real-time, live inspection of traffic associated with the one or more tenants. The inline monitoring can include real-time, live inspection of Kerberos, Light-weight Directory Access Protocol (LDAP), Server Message Block (SMB), and Distributed Computing Environment/Remote Procedure Calls (DCERPC) traffic. The inspecting can include determining one or more attack signatures based on the traffic. The one or more actions can include alerting users of a tenant of the cloud-based system responsive to detecting one or more attack signatures. The alerting can include providing remediation steps for mitigating a potential attack. The one or more actions can include blocking access to an active directory domain responsive to detecting one or more attack signatures. The inspecting can be performed for each of the plurality of tenants based on an inspection profile of each of the plurality of tenants. The steps can include receiving, from each of the plurality of tenants, an inspection profile for performing the inspecting. The inspecting can be performed at an application connector of the cloud-based system.

CONCLUSION

[0171] It will be appreciated that some embodiments described herein may include one or more generic or specialized processors ("one or more processors") such as microprocessors; Central Processing Units (CPUs); Digital Signal Processors (DSPs): customized processors such as Network Processors (NPs) or Network Processing Units (NPUs), Graphics Processing Units (GPUs), or the like; Field Programmable Gate Arrays (FPGAs); and the like along with unique stored program instructions (including both software and firmware) for control thereof to implement, in conjunction with certain non-processor circuits, some, most, or all of the functions of the methods and/or systems described herein. Alternatively, some or all functions may be implemented by a state machine that has no stored program instructions, or in one or more Application Specific Integrated Circuits (ASICs), in which each function or some combinations of certain of the functions are implemented as custom logic or circuitry. Of course, a combination of the aforementioned approaches may be used. For some of the embodiments described herein, a corresponding device such as hardware, software, firmware, and a combination thereof can be referred to as "circuitry configured or adapted to," "logic configured or adapted to," etc. perform a set of operations, steps, methods, processes, algorithms, functions, techniques, etc. as described herein for the various embodiments.

[0172] Moreover, some embodiments may include a non-transitory computer-readable storage medium having computer readable code stored thereon for programming a computer, server, appliance, device, processor, circuit, etc. each of which may include a processor to perform functions as described and claimed herein. Examples of such computer-readable storage mediums include, but are not limited to, a hard disk, an optical storage device, a magnetic storage device, a ROM (Read Only Memory), a PROM (Programmable Read Only Memory), an EPROM (Erasable Programmable Read Only Memory), an EEPROM (Electrically Erasable Programmable Read Only Memory), Flash memory, and the like. When stored in the non-transitory computer readable medium, software can include instructions executable by a processor or device (e.g., any type of programmable circuitry or logic) that, in response to such execution, cause a processor or the device to perform a set of operations, steps, methods, processes, algorithms, functions, techniques, etc. as described herein for the various embodiments.

[0173] Although the present disclosure has been illustrated and described herein with reference to preferred embodiments and specific examples thereof, it will be readily apparent to those of ordinary skill in the art that other embodiments and examples may perform similar functions and/or achieve like results. All such equivalent embodiments and examples are within the spirit and scope of the present disclosure, are contemplated thereby, and are intended to be covered by the following claims. Moreover, it is noted that the various elements, operations, steps, methods, processes, algorithms, functions, techniques, etc., described herein can be used in any and all combinations with each other.

What is claimed is:

1. A method implemented by a cloud-based system, the method comprising steps of:

performing inline monitoring of traffic associated with a plurality of tenants of the cloud-based system;

classifying the traffic as being associated with any of one or more active directory protocols;

inspecting the traffic associated with the one or more detected active directory protocols; and

performing one or more actions on the traffic based on the inspecting.

2. The method of claim **1**, wherein the inline monitoring includes real-time, live inspection of traffic associated with the one or more tenants.

3. The method of claim **1**, wherein the inline monitoring includes real-time, live inspection of Kerberos, Light-weight Directory Access Protocol (LDAP), Server Message Block (SMB), and Distributed Computing Environment/Remote Procedure Calls (DCERPC) traffic.

4. The method of claim **1**, wherein the inspecting includes determining one or more attack signatures based on the traffic.

5. The method of claim **1**, wherein the one or more actions include alerting users of a tenant of the cloud-based system responsive to detecting one or more attack signatures.

6. The method of claim **5**, wherein the alerting includes providing remediation steps for mitigating a potential attack.

7. The method of claim **1**, wherein the one or more actions include blocking access to an active directory domain responsive to detecting one or more attack signatures.

8. The method of claim **1**, wherein the inspecting is performed for each of the plurality of tenants based on an inspection profile of each of the plurality of tenants.

9. The method of claim **8**, wherein the steps comprise receiving, from each of the plurality of tenants, an inspection profile for performing the inspecting.

10. The method of claim **1**, wherein the inspecting is performed at an application connector of the cloud-based system.

11. A non-transitory computer-readable medium comprising instructions that, when executed, cause one or more processors of a cloud-based system to perform steps of:

performing inline monitoring of traffic associated with a plurality of tenants of the cloud-based system;

classifying the traffic as being associated with any of one or more active directory protocols;

inspecting the traffic associated with the one or more detected active directory protocols; and

performing one or more actions on the traffic based on the inspecting.

12. The non-transitory computer-readable medium of claim **11**, wherein the inline monitoring includes real-time, live inspection of traffic associated with the one or more tenants.

13. The non-transitory computer-readable medium of claim **11**, wherein the inline monitoring includes real-time, live inspection of Kerberos, Light-weight Directory Access Protocol (LDAP), Server Message Block (SMB), and Distributed Computing Environment/Remote Procedure Calls (DCERPC) traffic.

14. The non-transitory computer-readable medium of claim **11**, wherein the inspecting includes determining one or more attack signatures based on the traffic.

15. The non-transitory computer-readable medium of claim **11**, wherein the one or more actions include alerting users of a tenant of the cloud-based system responsive to detecting one or more attack signatures.

16. The non-transitory computer-readable medium of claim **15**, wherein the alerting includes providing remediation steps for mitigating a potential attack.

17. The non-transitory computer-readable medium of claim **11**, wherein the one or more actions include blocking access to an active directory domain responsive to detecting one or more attack signatures.

18. The non-transitory computer-readable medium of claim **11**, wherein the inspecting is performed for each of the plurality of tenants based on an inspection profile of each of the plurality of tenants.

19. The non-transitory computer-readable medium of claim **18**, wherein the steps comprise receiving, from each of the plurality of tenants, an inspection profile for performing the inspecting.

20. The non-transitory computer-readable medium of claim **11**, wherein the inspecting is performed at an application connector of the cloud-based system.

* * * * *