



## List of Applications and Protocols

On the next pages, you will find a detailed description of all VulcanAppMix protocols and applications.

For more information about VulcanAppMix please visit our [webpage](#).

## Protocols

AFS	Secure MQTT	CNN	Mobile Bank
BitTorrent	SIP	DNS	Oracle MySQL
DNS	SMB2	Dropbox	Oracle MySQL
Echo	SMTP	eBay	Oracle MySQL
FIX	SMTP over TLS	Email	over TLS
FTP (active)	SRTP	application	Outlook Web
FTP (passive)	SSDP	Facebook	Mail
HTTP	SSHv2	Finance orders	Paypal
HTTPS	TELNET (per-character)	Firefox	QQ App
IMAP	TELNET (per-line)	Flickr	Reddit
(encrypted)	TFTP Read	Gmail	Remote Desktop
IMAPS	Request	Google App	RSS
LDAP	TFTP Write	Google Calendar	SIP VoIP
LLMNR	Request	Google Hangouts	Skype
MDNS		App	Slack App
MQTT over TCP		Google Search	Tumblr
MSEExchange		Google Drive	Twitter
MAPI	<b>Applications</b>	Google Maps	Video stream
NBNS	Amazon	Hotmail Web	1080p over HTTP
NFS	App Store App	Instagram	Video stream
POP3	Apple Map	iOS Calendar	1080p over RTP
(unencrypted)	AWS S3	IoT Temperature	WeChat App
POP3 over TLS	Bing	publish	Weibo
QUIC	BitTorrent	IoT Temperature	Wikipedia
RADIUS	(Small)	publish over TLS	Search
RDP	Bloomberg	iTunes App	Yahoo
RTP/RTCP	Chrome	LINE App	Yahoo Mail Web
RTSP	Chrome	LinkedIn	YouTube
	Incognito		

## Protocols

<b>AFS</b>	<p>The Andrew File System (AFS) is a distributed file system which uses a set of trusted servers to present a homogeneous, location-transparent file name space to all the client workstations.</p>
<b>BitTorrent</b>	<p>BitTorrent is a communications protocol of peer-to-peer file sharing ("P2P") which is used to distribute data and electronic files over the Internet. BitTorrent is one of the most common protocols for transferring large files, such as digital audio files containing TV shows or video clips or digital audio files containing songs. Peer-to-peer networks have been estimated to collectively account for approximately 43% to 70% of all Internet traffic (depending on location) as of February 2009. In November 2004, BitTorrent was responsible for 25% of all Internet traffic. As of February 2013, BitTorrent was responsible for 3.35% of all worldwide bandwidth, more than half of the 6% of total bandwidth dedicated to file sharing.</p>
<b>DNS</b>	<p>The Domain Name System (DNS) is a hierarchical decentralized naming system for computers, services, or any resource connected to the Internet or a private network. It associates various information with domain names assigned to each of the participating entities. Most prominently, it translates more readily memorized domain names to the numerical IP addresses needed for the purpose of locating and identifying computer services and devices with the underlying network protocols. By providing a worldwide, distributed directory service, the Domain Name System is an essential component of the functionality of the Internet.</p>
<b>Echo</b>	<p>The Echo Protocol is a service in the Internet Protocol Suite defined in RFC 862. A host may connect to a server that supports the Echo Protocol using the Transmission Control Protocol (TCP) or the User Datagram Protocol (UDP) on the well-known port number 7. The server sends back an identical copy of the data it received.</p>
<b>FIX4.0</b>	<p>The Financial Information eXchange (FIX) protocol is an electronic communications protocol initiated in 1992 for international real-time exchange of information related to the securities transactions and markets. Managing the delivery of trading applications and keeping latency low increasingly requires an understanding of the FIX protocol.</p>
<b>FIX4.1</b>	<p>The Financial Information eXchange (FIX) protocol is an electronic communications protocol initiated in 1992 for international real-time exchange of information related to the securities transactions and markets. Managing the delivery of trading applications and keeping latency low increasingly requires an understanding of the FIX protocol.</p>

<b>FIX4.2</b>	The Financial Information eXchange (FIX) protocol is an electronic communications protocol initiated in 1992 for international real-time exchange of information related to the securities transactions and markets. Managing the delivery of trading applications and keeping latency low increasingly requires an understanding of the FIX protocol.
<b>FIX4.3</b>	The Financial Information eXchange (FIX) protocol is an electronic communications protocol initiated in 1992 for international real-time exchange of information related to the securities transactions and markets. Managing the delivery of trading applications and keeping latency low increasingly requires an understanding of the FIX protocol.
<b>FIX4.4</b>	The Financial Information eXchange (FIX) protocol is an electronic communications protocol initiated in 1992 for international real-time exchange of information related to the securities transactions and markets. Managing the delivery of trading applications and keeping latency low increasingly requires an understanding of the FIX protocol.
<b>FIX5.0</b>	The Financial Information eXchange (FIX) protocol is an electronic communications protocol initiated in 1992 for international real-time exchange of information related to the securities transactions and markets. Managing the delivery of trading applications and keeping latency low increasingly requires an understanding of the FIX protocol.
<b>FIXT1.1</b>	The Financial Information eXchange (FIX) protocol is an electronic communications protocol initiated in 1992 for international real-time exchange of information related to the securities transactions and markets. Managing the delivery of trading applications and keeping latency low increasingly requires an understanding of the FIX protocol.
<b>FTP (active)</b>	The File Transfer Protocol (FTP) is a standard network protocol used to transfer computer files between a client and server on a computer network. FTP is built on a client-server model architecture and uses separate control and data connections between the client and the server. FTP users may authenticate themselves with a clear-text sign-in protocol, normally in the form of a username and password, but can connect anonymously if the server is configured to allow it.
<b>FTP (passive)</b>	The File Transfer Protocol (FTP) is a standard network protocol used to transfer computer files between a client and server on a computer network. FTP is built on a client-server model architecture and uses separate control and data connections between the client and the server. FTP users may authenticate themselves with a clear-text sign-in protocol, normally in the form of a username and password, but can connect anonymously if the server is configured to allow it.

<p><b>HTTP</b></p>	<p>The Hypertext Transfer Protocol (HTTP) is an application protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web. HTTP functions as a request-response protocol in the client-server computing model. A web browser, for example, may be the client and an application running on a computer hosting a web site may be the server. The client submits an HTTP request message to the server. The server, which provides resources such as HTML files and other content, or performs other functions on behalf of the client, returns a response message to the client. The response contains completion status information about the request and may also contain requested content in its message body. An HTTP client initiates a request by establishing a Transmission Control Protocol (TCP) connection to a particular port on a server (typically port 80, occasionally port 8080). An HTTP server listening on that port waits for a client's request message. Upon receiving the request, the server sends back a status line, such as "HTTP/1.1 200 OK", and a message of its own. The body of this message is typically the requested resource, although an error message or other information may also be returned.</p>
<p><b>HTTPS</b></p>	<p>The Hypertext Transfer Protocol (HTTP) is an application protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web. HTTP functions as a request-response protocol in the client-server computing model. A web browser, for example, may be the client and an application running on a computer hosting a web site may be the server. The client submits an HTTP request message to the server. The server, which provides resources such as HTML files and other content, or performs other functions on behalf of the client, returns a response message to the client. The response contains completion status information about the request and may also contain requested content in its message body. An HTTP client initiates a request by establishing a Transmission Control Protocol (TCP) connection to a particular port on a server (typically port 80, occasionally port 8080). An HTTP server listening on that port waits for a client's request message. Upon receiving the request, the server sends back a status line, such as "HTTP/1.1 200 OK", and a message of its own. The body of this message is typically the requested resource, although an error message or other information may also be returned.</p>
<p><b>IMAP (encrypted)</b></p>	<p>The Internet Message Access Protocol (IMAP) is an Internet standard protocol used by e-mail clients to retrieve e-mail messages from a mail server over a TCP/IP connection. IMAP is defined by RFC 3501. IMAP was designed with the goal of permitting complete management of an email box by multiple email clients, therefore clients generally leave messages on the server until the user explicitly deletes them. An IMAP server typically listens on port number 143. IMAP over SSL (IMAPS) is assigned the port number 993. Virtually all modern e-mail clients and servers support IMAP. IMAP and the earlier POP3 (Post Office Protocol) are the two most prevalent standard protocols for email retrieval, with many webmail service providers such as Gmail, Outlook.com and Yahoo! Mail also providing support for either IMAP or POP3.</p>
<p><b>IMAPS</b></p>	<p>The Internet Message Access Protocol (IMAP) is an Internet standard protocol used by e-mail clients to retrieve e-mail messages from a mail server over a TCP/IP connection. IMAP is defined by RFC 3501. IMAP was designed with the goal of permitting complete management of an email box by multiple email clients, therefore clients generally leave messages on the server until the user explicitly deletes them. An IMAP server typically listens on port number 143. IMAP over SSL (IMAPS) is assigned the port number 993. Virtually all modern e-mail clients and servers support IMAP. IMAP and the earlier POP3 (Post Office Protocol) are the two most prevalent standard protocols for email retrieval, with many webmail service providers such as Gmail, Outlook.com and Yahoo! Mail also providing support for either IMAP or POP3.</p>
<p><b>LDAP (search filter with AND filter)</b></p>	<p>The Lightweight Directory Access Protocol is an open, vendor-neutral, industry standard application protocol for accessing and maintaining distributed directory information services over an Internet Protocol (IP) network. Directory services play an important role in developing intranet and Internet applications by allowing the sharing of information about users, systems, networks, services, and applications throughout the network. As examples, directory services may provide any organized set of records, often with a hierarchical structure, such as a corporate email directory. Similarly, a telephone directory is a list of subscribers with an address and a phone number.</p>

<b>LDAP PDU with DIRSYNC CONTROLS</b>	<p>The Lightweight Directory Access Protocol is an open, vendor-neutral, industry standard application protocol for accessing and maintaining distributed directory information services over an Internet Protocol (IP) network. Directory services play an important role in developing intranet and Internet applications by allowing the sharing of information about users, systems, networks, services, and applications throughout the network. As examples, directory services may provide any organized set of records, often with a hierarchical structure, such as a corporate email directory. Similarly, a telephone directory is a list of subscribers with an address and a phone number.</p>
<b>LDAP (search filter with a simple extensible match)</b>	<p>The Lightweight Directory Access Protocol is an open, vendor-neutral, industry standard application protocol for accessing and maintaining distributed directory information services over an Internet Protocol (IP) network. Directory services play an important role in developing intranet and Internet applications by allowing the sharing of information about users, systems, networks, services, and applications throughout the network. As examples, directory services may provide any organized set of records, often with a hierarchical structure, such as a corporate email directory. Similarly, a telephone directory is a list of subscribers with an address and a phone number.</p>
<b>LLMNR</b>	<p>The Link-Local Multicast Name Resolution (LLMNR) is a protocol based on the Domain Name System (DNS) packet format that allows both IPv4 and IPv6 hosts to perform name resolution for hosts on the same local link. It is included in Windows Vista, Windows Server 2008, Windows 7, Windows 8 and Windows 10. It is also implemented by systemd-resolved on Linux. LLMNR is defined in RFC 4795.</p>
<b>MDNS</b>	<p>In computer networking, the multicast Domain Name System (mDNS) resolves host names to IP addresses within small networks that do not include a local name server. It is a zero-configuration service, using essentially the same programming interfaces, packet formats and operating semantics as the unicast Domain Name System (DNS). Although Stuart Cheshire designed mDNS to be stand-alone capable, it can work in concert with unicast DNS servers.</p>
<b>MQTT over TCP</b>	<p>MQTT (formerly MQ Telemetry Transport) is an ISO standard (ISO/IEC PRF 20922) publish-subscribe-based "lightweight" messaging protocol for use on top of the TCP/IP protocol. It is designed for connections with remote locations where a "small code footprint" is required or the network bandwidth is limited. The publish-subscribe messaging pattern requires a message broker. The broker is responsible for distributing messages to interested clients based on the topic of a message.</p>
<b>MSExchange MAPI</b>	<p>Messaging Application Programming Interface (MAPI) is a messaging architecture and a Component Object Model based API for Microsoft Windows. MAPI allows client programs to become (e-mail) messaging-enabled, -aware, or -based by calling MAPI subsystem routines that interface with certain messaging servers. While MAPI is designed to be independent of the protocol, it is usually used with MAPI/RPC, the proprietary protocol that Microsoft Outlook uses to communicate with Microsoft Exchange.</p>
<b>NBNS</b>	<p>This service is often called WINS on Windows systems. The NetBIOS Name Service is part of the NetBIOS-over-TCP protocol suite. NBNS serves much the same purpose as DNS does: translate human-readable names to IP addresses. As NetBIOS can run on top of several different network protocols (e.g. IP, IPX, ...), other implementations of the NetBIOS services have their own mechanisms for translating NetBIOS names to addresses.) NBNS's services are more limited, in that NetBIOS names exist in a flat name space, rather than DNS's hierarchical one (multiple flat name spaces can exist, by using NetBIOS scopes, but those are rarely used), and NBNS can only supply IPv4 addresses; NBNS doesn't support IPv6.</p>

<b>NFSv3</b>	Network File System (NFS) is a distributed file system protocol allowing a user on a client computer to access files over a computer network much like local storage is accessed. NFS, like many other protocols, builds on the Open Network Computing Remote Procedure Call (ONC RPC) system. The NFS is an open standard defined in Request for Comments (RFC), allowing anyone to implement the protocol.
<b>NFSv2</b>	Network File System (NFS) is a distributed file system protocol allowing a user on a client computer to access files over a computer network much like local storage is accessed. NFS, like many other protocols, builds on the Open Network Computing Remote Procedure Call (ONC RPC) system. The NFS is an open standard defined in Request for Comments (RFC), allowing anyone to implement the protocol.
<b>POP3 (unencrypted)</b>	The Post Office Protocol (POP) is an application-layer Internet standard protocol used by local e-mail clients to retrieve e-mail from a remote server over a TCP/IP connection. POP has been developed through several versions, with version 3 (POP3) being the last standard in common use before largely being made obsolete by the more advanced IMAP. In POP3, e-mails are downloaded from the server's inbox to your computer. E-mails are available when you are not connected.
<b>POP3 over TLS</b>	The Post Office Protocol (POP) is an application-layer Internet standard protocol used by local e-mail clients to retrieve e-mail from a remote server over a TCP/IP connection. POP has been developed through several versions, with version 3 (POP3) being the last standard in common use before largely being made obsolete by the more advanced IMAP. In POP3, e-mails are downloaded from the server's inbox to your computer. E-mails are available when you are not connected.
<b>QUIC</b>	QUIC (Quick UDP Internet Connections, pronounced quick) is an experimental transport layer network protocol designed at Google, initially implemented in 2012, and announced as experimentation broadened in 2013. QUIC supports a set of multiplexed connections between two endpoints over User Datagram Protocol (UDP), and was designed to provide security protection equivalent to TLS/SSL, along with reduced connection and transport latency, and bandwidth estimation in each direction to avoid congestion. QUIC's main goal is to improve perceived performance of connection-oriented web applications that are currently using TCP. It also provides a venue for rapid iteration of congestion avoidance algorithms, placing control into application space at both endpoints, rather than (the relatively slow to evolve) kernel space.
<b>RADIUS</b>	Emulates the RADIUS Access [RFC 2865] protocol.
<b>RDP</b>	Remote Desktop Protocol (RDP) is a proprietary protocol developed by Microsoft, which provides a user with a graphical interface to connect to another computer over a network connection. The user employs RDP client software for this purpose, while the other computer must run RDP server software. TPKT: Typically, RDP uses TPKT as its transport protocol. TPKT runs atop TCP; when used to transport RDP, the well known TCP port is 3389, rather than the normal TPKT port 102.

<b>RTP/RTCP</b>	The Real-time Transport Protocol (RTP) is a network protocol for delivering audio and video over IP networks. RTP is used extensively in communication and entertainment systems that involve streaming media, such as telephony, video teleconference applications, television services and web-based push-to-talk features. RTP typically runs over User Datagram Protocol (UDP). RTP is used in conjunction with the RTP Control Protocol (RTCP). While RTP carries the media streams (e.g., audio and video), RTCP is used to monitor transmission statistics and quality of service (QoS) and aids synchronization of multiple streams. RTP is one of the technical foundations of Voice over IP and in this context is often used in conjunction with a signaling protocol such as the Session Initiation Protocol (SIP) which establishes connections across the network.
<b>RTSP</b>	The Real Time Streaming Protocol (RTSP) is a network control protocol designed for use in entertainment and communications systems to control streaming media servers. The protocol is used for establishing and controlling media sessions between end points. Clients of media servers issue VCR-style commands, such as play, record and pause, to facilitate real-time control of the media streaming from the server to a client (Video On Demand) or from a client to the server (Voice Recording).
<b>Secure MQTT</b>	MQTT (formerly MQ Telemetry Transport) is an ISO standard (ISO/IEC PRF 20922) publish-subscribe-based "lightweight" messaging protocol for use on top of the TCP/IP protocol. It is designed for connections with remote locations where a "small code footprint" is required or the network bandwidth is limited. The publish-subscribe messaging pattern requires a message broker. The broker is responsible for distributing messages to interested clients based on the topic of a message.
<b>SIP</b>	The Session Initiation Protocol (SIP) is a communications protocol for signaling and controlling multimedia communication sessions. The most common applications of SIP are in Internet telephony for voice and video calls, as well as instant messaging, over IP networks. SIP works in conjunction with several other application layer protocols that identify and carry the session media. Media identification and negotiation is achieved with the Session Description Protocol (SDP). For the transmission of media streams (voice, video) SIP typically employs the Real-time Transport Protocol (RTP) or Secure Real-time Transport Protocol (SRTP). For secure transmissions of SIP messages, the protocol may be encrypted with Transport Layer Security (TLS).
<b>SMB2</b>	In computer networking, Server Message Block (SMB), one version of which was also known as Common Internet File System (CIFS), operates as an application-layer network protocol mainly used for providing shared access to files, printers, and serial ports and miscellaneous communications between nodes on a network. It also provides an authenticated inter-process communication mechanism.
<b>SMTP</b>	Simple Mail Transfer Protocol (SMTP) is an Internet standard for electronic mail (email) transmission. First defined by RFC 821 in 1982, it was last updated in 2008 with the Extended SMTP additions by RFC 5321—which is the protocol in widespread use today. SMTP by default uses TCP port 25. The protocol for mail submission is the same, but uses port 587. SMTP connections secured by SSL, known as SMTPS, default to port 465 (nonstandard, but sometimes used for legacy reasons).
<b>SMTP over TLS</b>	Simple Mail Transfer Protocol (SMTP) is an Internet standard for electronic mail (email) transmission. First defined by RFC 821 in 1982, it was last updated in 2008 with the Extended SMTP additions by RFC 5321—which is the protocol in widespread use today. SMTP by default uses TCP port 25. The protocol for mail submission is the same, but uses port 587. SMTP connections secured by SSL, known as SMTPS, default to port 465 (nonstandard, but sometimes used for legacy reasons).

<b>SRTP</b>	The Secure Real-time Transport Protocol (or SRTP) defines a profile of RTP (Real-time Transport Protocol), intended to provide encryption, message authentication and integrity, and replay protection to the RTP data in both unicast and multicast applications.
<b>SSDP</b>	The Simple Service Discovery Protocol (SSDP) is a network protocol based on the Internet Protocol Suite for advertisement and discovery of network services and presence information. It accomplishes this without assistance of server-based configuration mechanisms, such as the Dynamic Host Configuration Protocol (DHCP) or the Domain Name System (DNS), and without special static configuration of a network host. SSDP is the basis of the discovery protocol of Universal Plug and Play (UPnP) and is intended for use in residential or small office environments.
<b>SSHv2</b>	Secure Shell (SSH) is a cryptographic network protocol for operating network services securely over an unsecured network. The best known example application is for remote login to computer systems by users.
<b>TELNET (per-character)</b>	Telnet is an application layer protocol used on the Internet or local area networks to provide a bidirectional interactive text-oriented communication facility using a virtual terminal connection. User data is interspersed in-band with Telnet control information in an 8-bit byte oriented data connection over the Transmission Control Protocol (TCP). Telnet is a client-server protocol, based on a reliable connection-oriented transport. Typically, this protocol is used to establish a connection to Transmission Control Protocol (TCP) port number 23, where a Telnet server application (telnetd) is listening.
<b>TELNET (per-line)</b>	Telnet is an application layer protocol used on the Internet or local area networks to provide a bidirectional interactive text-oriented communication facility using a virtual terminal connection. User data is interspersed in-band with Telnet control information in an 8-bit byte oriented data connection over the Transmission Control Protocol (TCP). Telnet is a client-server protocol, based on a reliable connection-oriented transport. Typically, this protocol is used to establish a connection to Transmission Control Protocol (TCP) port number 23, where a Telnet server application (telnetd) is listening.
<b>TFTP Read Request</b>	Trivial File Transfer Protocol (TFTP) is a simple lockstep File Transfer Protocol which allows a client to get a file from or put a file onto a remote host. One of its primary uses is in the early stages of nodes booting from a local area network. TFTP has been used for this application because it is very simple to implement.
<b>TFTP Write Request</b>	Trivial File Transfer Protocol (TFTP) is a simple lockstep File Transfer Protocol which allows a client to get a file from or put a file onto a remote host. One of its primary uses is in the early stages of nodes booting from a local area network. TFTP has been used for this application because it is very simple to implement.

## Applications

<p><b>Amazon Sep16</b></p>	<p>Emulates the use of the Amazon website as of September 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.</p>
<p><b>App Store App Nov16</b></p>	<p>Emulates the use of the App Store app as of November 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.</p>
<p><b>Apple Map Nov 16</b></p>	<p>Emulates the use of the Apple iOS Map app as of November 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.</p>
<p><b>AWS S3 Jan 17</b></p>	<p>Emulates the use of the AWS S3 as of January 2017 for downloading. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.</p>
<p><b>Bing Search Nov16</b></p>	<p>Emulates the use of the Bing website as of November 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.</p>
<p><b>BitTorrent (Small)</b></p>	<p>Emulates the use of the BitTorrent to download. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.</p>
<p><b>Bloomberg Nov 16</b></p>	<p>Emulates the use of the Bloomberg app as of November 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.</p>
<p><b>Chrome Jan17</b></p>	<p>Emulates the use of the Chrome browser as of January 2017. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.</p>
<p><b>Chrome Jul16</b></p>	<p>Emulates the use of the Chrome browser as of July 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.</p>

<b>Chrome Incognito Jan17</b>	Emulates the use of the Chrome browser in incognito mode as of January 2017. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Chrome Incognito Jul16</b>	Emulates the use of the Chrome browser in incognito mode as of July 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>CNN Nov16</b>	Emulates the use of the CNN app as of November 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>DNS Jan17</b>	Emulates DNS queries as of January 2017 for downloading. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Dropbox Nov16</b>	Emulates the use of the Dropbox app as of November 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>eBay Sep16</b>	Emulates the use of the eBay website as of September 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Email application</b>	Email application uses multiple protocols to open connections for data exchange
<b>Facebook Mar17</b>	Emulates the use of the Facebook website as of March 2017. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Facebook Jan17</b>	Emulates the use of the Facebook website as of January 2017. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Facebook Sep16</b>	Emulates the use of the Facebook website as of September 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.

<b>Facebook Messenger Nov16</b>	Emulates the use of the Facebook Messenger app as of November 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Finance orders (FIX4.0)</b>	Emulates a financial client places share orders to the server using FIX 4.0
<b>Finance orders (FIX4.1)</b>	Emulates a financial client places share orders to the server using FIX 4.1
<b>Finance orders (FIX4.2)</b>	Emulates a financial client places share orders to the server using FIX 4.2
<b>Finance orders (FIX4.3)</b>	Emulates a financial client places share orders to the server using FIX 4.3
<b>Finance orders (FIX4.4)</b>	Emulates a financial client places share orders to the server using FIX 4.4
<b>Finance orders (FIX5.0)</b>	Emulates a financial client places share orders to the server using FIX 5.0
<b>Finance orders (FIXT1.1)</b>	Emulates a financial client places share orders to the server using FIXT 1.1
<b>Firefox Jan17</b>	Emulates the use of the Firefox browser as of January 2017. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Firefox Private Jan17</b>	Emulates the use of the Firefox browser in private mode as of January 2017. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Flickr Nov16</b>	Emulates the use of the Flickr app as of November 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.

<b>Gmail Web Jan17</b>	Emulates the use of the Gmail as of January 2017. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Google App Nov16</b>	Emulates the use of the Google app as of November 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Google Calendar Jan17</b>	Emulates the use of the Google Calendar app as of January 2017. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Google Hangouts App Nov16</b>	Emulates the use of the Hangouts app as of November 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Google Search Nov16</b>	Emulates the use of the Google search website as of November 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Google Drive Nov16</b>	Emulates the use of the GoogleDrive app as of November 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Google Maps Nov16</b>	Emulates the use of the GoogleMap website as of November 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Hotmail Web Jan17</b>	Emulates the use of the Hotmail as of January 2017. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Instagram Nov16</b>	Emulates the use of the Instagram app as of November 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>iOS Calendar Nov 16</b>	Emulates the use of the iOS calendar app as of November 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.

<b>IoT Temperature publish</b>	Emulates the use of IoT devices to publish temperature data periodically with MQTT.
<b>IoT Temperature publish over TLS</b>	Emulates the use of IoT devices to publish temperature data periodically with MQTT over TLS.
<b>iTunes App Nov16</b>	Emulates the use of the iTunes app as of November 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>LINE App Nov16</b>	Emulates the use of the Line app as of November 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>LinkedIn Nov16</b>	Emulates the use of the LinkedIn as of November 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Mobile Bank Nov 16</b>	Emulates the use of the mobile bank app as of November 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Oracle MySQL Mar17</b>	Emulates the use of the MySQL as of March 2017. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Oracle MySQL Jan17</b>	Emulates the use of the MySQL as of January 2017. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Oracle MySQL over TLS Mar17</b>	Emulates the use of the MySQL over TLS as of March 2017. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Outlook Web Mail Jan17</b>	Emulates the use of the Outlook Web Mail as of Januaray 2017. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.

<b>Paypal Nov16</b>	Emulates the use of the LinkedIn website as of November 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>QQ App Nov16</b>	Emulates the use of the QQ app as of November 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Reddit Nov16</b>	Emulates the use of the Reddit as of November 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Remote Desktop</b>	Emulates the use of remote desktop application between two computers in an enterprise LAN environment. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>RSS Feed</b>	RSS (Rich Site Summary) uses a family of standard web feed formats to publish frequently updated information: blog entries, news headlines, audio, video. An RSS document includes full or summarized text, and metadata, like publishing date and author's name.
<b>SIP VoIP</b>	Emulates the use of a VoIP application between two SIP clients in an enterprise LAN environment. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Skype</b>	Emulates the use of Skype to chat and call between two users. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Slack App Oct16</b>	Emulates the use of the Slack PC client as of October 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Tumblr Jan17</b>	Emulates the use of the Tumblr website as of January 2017. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Twitter Sep16</b>	Emulates the use of the Twitter website as of September 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.

<b>Video stream 1080p over HTTP</b>	Emulates the streaming application of a 1080p video over HTTP 8080 in an enterprise LAN environment. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Video stream 1080p over RTP</b>	Emulates the streaming application of a 1080p video over RTP in an enterprise LAN environment. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>WeChat App Nov16</b>	Emulates the use of the WeChat app as of November 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Weibo Nov16</b>	Emulates the use of the Weibo website as of November 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Wikipedia Search Jan16</b>	Emulates the use of the Wikipedia as of January 2017. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Yahoo Sep16</b>	Emulates the use of the Yahoo website as of September 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Yahoo Mail Web Apr17</b>	Emulates the use of the Yahoo Mail as of April 2017. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>Yahoo Mail Web Jan17</b>	Emulates the use of the Yahoo Mail as of January 2017. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>YouTube Mar17</b>	Emulates the use of the YouTube website as of March 2017. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.
<b>YouTube Nov16</b>	Emulates the use of the Youtube website as of November 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.

## **YouTube Sep16**

Emulates the use of the Youtube website as of September 2016. This application uses dynamically created flows to simulate the various internal actions performed by a web browser. Due to these dynamic flows may be large in number and may contain a large amount of generated data, scenarios creation and test initialization may require a considerable amount of time.