Ports and Protocols Extended Control for Security

William R Simpson, *Member IAENG* and Kevin E. Foltz

***Abstract*— Network protocols have vulnerabilities, and one way to reduce these vulnerabilities is to reduce the protocols in use to a small set of well-tested standard protocols. This reduces the attack surface and provides high confidence in selected communications. Screening of acceptable ports and protocols can be done by network appliances known as firewalls. Communications on the approved list are permitted, and others blocked. Many appliances now have port and protocol filters, and the server or service itself may have a host-based security system that can apply this functionality. This paper covers enterprise considerations for use and screening of ports and protocols.**

***Index Terms* — Appliance, Firewall, IT Security, Ports, Ports and Protocols, Traffic Inspection**

I. INTRODUCTION

Guidance and policies that govern the use, configuration and management of the communication protocols in use by the web services and applications that are connected to the network are required for interoperability and security. Policies specify the proper use of ports and protocols in order to control what types of communications are allowed to cross the boundaries of the networks. This paper is based in part on a paper published by WCECS 2016 [1].

Basically, a port is an access channel to and from a specific service, and a protocol is a standardized way for computers to exchange information. Data on the network is sent and received by software that automatically organizes such data to be transferred into packets, made in a standardized way (defined by the protocol in use) so that the destination host can recognize them as data and properly decode them. Network clients use different ports or channels (which are given standardized numbers) to transfer data.

The port number (and the destination IP address) is included as part of the header each packet is given in order to deliver the packet to the proper end-point service. The policies on Ports, Protocols, and Services (PPS) are typically enforced by network and security appliances and software such as routers, firewalls, and intrusion detection/protection devices that protect the boundary of the network or reside at the end-points (i.e., web services or clients).

Originally, the transmission was done at half duplex, and two ports were needed for the control program.

Eventually, Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) were adopted, and only one port was needed. TCP and UDP port numbers are also used by other protocols. The Internet Assigned Numbers Authority (IANA) maintains the official assignments of port numbers for specific uses [2, 3]. However, many unofficial uses of both well-known and registered port numbers occur in practice. A few ports and their usage are given in Table 1. There are 65,536 ports available as a 16-bit unsigned integer.

Table 1 Some Example Ports and Protocols

Port	Protocol	Messaging Protocol	Status
18	TCP, UDP	The Message Send Protocol (MSP) is an application layer protocol. Defined in RFC 1312 [4].	Official
80	TCP, UDP	Hypertext Transfer Protocol (HTTP). RFC 2068 [5]	Official
110	TCP	Post Office Protocol v3 (POP3) is an email retrieval protocol. RFC 1081 [6]	Official
143	TCP	Internet Message Access Protocol (IMAP) e-mail retrieval and storage as an alternative to POP. Defined in RFC 3501 [7]	Official
161	UDP	Simple Network Management Protocol (SNMP) defined in RFC 3411[8].	Official
213	TCP, UDP	Internetwork Packet Exchange (IPX) RFC 1132 [9]	Official
443	TCP, UDP	Hypertext Transfer Protocol over TLS/SSL (HTTPS) RFC 2818. [10]	Official
587	TCP	Simple Mail Transfer Protocol (SMTP), as specified in RFC 6409 [11]	Official
1935	TCP	Adobe Systems Macromedia Flash Real Time Messaging Protocol (RTMP) "plain" protocol. Adobe proprietary, H. Parmar, M. Thornburgh (eds.) Adobe's Real Time Messaging Protocol, Adobe, December 21, 2012. [12]	Official
2195	TCP	Apple Push Notification service link. Apple proprietary. https://en.wikipedia.org/wiki/Apple_Push_Notification_Service. [13]	Unofficial
4502	TCP, UDP	Microsoft Silverlight connectable ports under non-elevated trust [14]	Official
5672	TCP	Advanced Message Queuing Protocol (AMQP) ISO/IEC 19464 [15]	Official
8080	TCP	HTTP alternate	Official
49342	TCP	Avanset Exam Simulator (Visual CertExam file format (VCE) Player). Avanset proprietary. http://www.avanset.com/purchase/vce-exam-simulator.html [16]	Unofficial

Ports may be well-known, registered, and dynamic/private:

- Well-Known: Port numbers 0 through 1023 are used for common, well-known services.
- Registered: Port numbers 1024 through 49151 are the registered ports used for IANA-registered services.
- Dynamic/Private: Ports 49152 through 65535 are dynamic ports that are not officially designated for any specific service, and may be used for any purpose. They also are used as ephemeral ports, from which software running on the host may randomly choose a port in order to define itself. In effect, they are used as temporary ports, primarily

Manuscript received 24 February 2017; revised 10 March 2017. This work was supported in part by the U.S. Secretary of the Air Force and The Institute for Defense Analyses (IDA). The publication of this paper does not indicate endorsement by any organization in the Department of Defense or IDA, nor should the contents be construed as reflecting the official position of these organizations

Kevin E. Foltz is with the Institute for Defense Analyses.(email: kfoltz@ida.org)

William R. Simpson is with the Institute for Defense Analyses, 4850 Mark Center Drive, Alexandria, Virginia 22311 USA and is the corresponding author phone: 703-845-6637, FAX: 703-845-6848 (e-mail: rsimpson@ida.org)

by clients when communicating with servers. Dynamic/private ports can also be used by end-user applications, but are less commonly used so. Dynamic/private ports do not contain any meaning outside of any particular TCP connection.

Protocol standards may be:

• Proprietary – Set by an individual developer for use with his products or products developed by members in his consortium. This creates serious interoperability problems among different developers, and is a barrier to entry to new developers who do not agree to consortium rules.

• De Facto – Openly published by an individual developer, but adopted by enough developers that the protocols are widely in use. This promotes interoperability and the open publication removes barriers to entry.

• Standards-body-based –Industry-wide protocol definitions that are not tied to a particular manufacturer. With standard protocols, you can mix and match equipment from different vendors. As long as the equipment implements the standard protocols, it should be able to coexist on the same network.

Many organizations are involved in setting standards for networking. The most important organizations for the web are:

• International Organization for Standardization (ISO) – A federation of more than 100 standards organizations from throughout the world.

• Internet Engineering Task Force (IETF) – The organization responsible for the protocols that drive the Internet. These standards are cited by reference to their Request For Comments (RFC).

• World Wide Web Consortium (W3C) – An international organization that handles the development of standards for the World Wide Web.

This work is part of a larger body of work termed Consolidate Enterprise IT Baseline (CEITB). The security aspects of this baseline are termed Enterprise Level Security (ELS). The element and sub element locations within the baseline are shown in Figure 1. Each of the sub-elements must conform to both the CEITB and ELS requirements as applicable.

Figure 1 CEITB Architectural Element

In this paper we will review the communication models for web services, and the ports and assigned protocols. We will then review ELS and its basic architecture. Next, we review the threats to be considered, including how they affect server configuration and how firewalls are used for port blocking. Finally we provide the unique factors that arise with ports and protocols with this high security environment.

II. COMMUNICATION MODELS

The Internet Model is a group of communications protocols used for the Internet and similar networks. The Internet model is commonly known as TCP/IP, because of its most important protocols, the Transmission Control Protocol (TCP) and the Internet Protocol (IP). TCP/IP provides connectivity specifying how data should be formatted, addressed, transmitted, routed, and received at the destination. This functionality has been organized into four abstraction layers:

- Application Layer – Example Protocols: **BGP**[17] **, DNS**[18] **, FTP**[19] **, others...**
- Transport Layer – Example Protocols: **TCP, UDP, DCCP**[20] **, others...**
- Internet Layer – Example Internet Layer Protocols: **IP**[21] **, ECN**[22] **, IPsec**[23] **, others...**
- Link Layer – Example Link Layer Protocols: **Ethernet**[24] **, DSL**[25] **, PPP**[26] **, others....**

These layers are used to sort all related protocols according to the scope of the networking involved. IETF documents RFC 1122 [27] and RFC 1123 [28] describe the Internet Protocol suite and model.

An alternative model, the Open Systems Interconnection (OSI) model [29], is often used to describe protocols. The OSI model defines protocols in seven layers. The layers are: (1) Physical, (2) Data Link, (3) Network, (4) Transport, (5) Session, (6) Presentation, and (7) Application. The OSI model defines protocol implementations for its layers, and some of the specific details at each layer differ from those of the Internet model.

The OSI model, while popularly referenced, has succumbed to the Internet model. Unless specified, the Internet model will be used in this document.

III. PORTS IN TRANSPORT PROTOCOLS

Two primary transport protocols are used in the Internet, along with a plethora of special purpose ones. In this description, we limit the discussion to TCP and UDP.

For both of these protocols the port information is explicit in the header information, and it can be used by firewalls and servers to make an “accept or drop” decision.

A. The Transmission Control Protocol

TCP is one of the core protocols of the Internet Protocol suite and is so common that the entire suite is often called TCP/IP. Residing at the transport layer, TCP provides end-to-end, reliable, ordered, and error-checked delivery of a stream of octets between programs running on computers connected to a local area network, an intranet, or the public Internet. Web browsers use TCP when they connect to

servers on the World Wide Web, and it is used to deliver email and transfer files from one location to another. A variety of other higher-layer protocols use TCP/IP, such as HTTP, HTTPS, SMTP, POP3, IMAP, FTP, and their messages are typically encapsulated in TCP packets. TCP also provides a form of message flow control that will adapt its transmission rate to the congestion on the network. Applications that do not require the reliability of a TCP connection may instead use the connectionless User Datagram Protocol (UDP), which emphasizes low-overhead operation and reduced latency rather than error-checking and delivery validation.

TCP uses TCP Port Numbers to identify sending and receiving application end-points on the hosts. Each side of a TCP connection has an associated internet socket, defined as the host IP address and port number reserved by the sending or receiving application. Port 0 is generally reserved and should not be used. Arriving TCP data packets are identified as belonging to a specific TCP connection by its two sockets, that is, the four-tuple from the combination of source host IP address, source port, destination host IP address, and destination port. This means that a server computer can provide several clients with services simultaneously, as long as the four-tuples differ. A single client can have concurrent requests for a service, as long as the client takes care of initiating any connections to one destination port from different source ports. Well-known applications, running as servers and passively listening for connections typically use TCP ports. Some examples include:

- **FTP (Ports 20 and 21),**
- **SMTP (Port 25),**
- **SSL/TLS, HTTPS(Port 443),**
- **HTTP (Port 80).**

B. The User Datagram Protocol

UDP is one of the core members of the Internet protocol suite (the set of network protocols used for the Internet). With UDP, computer applications can send messages, in this case referred to as datagrams, to other hosts on an Internet protocol network without prior communications to set up special transmission channels or data paths. UDP uses a simple transmission model with a minimum of protocol mechanisms and overhead. It has no handshaking dialogues, and thus exposes any unreliability of the underlying network protocol to the user's program. Because this is normally IP over unreliable media, there is no guarantee of delivery, ordering, or duplicate protection. UDP provides checksums for data integrity, and port numbers for addressing different functions at the source and destination of the datagram. UDP is suitable for purposes for which error-checking and correction either are not necessary or are performed in the application, avoiding the overhead of such processing at the network interface level. Time-sensitive applications often use UDP because dropping packets is preferable to waiting for delayed packets, which may not be an option in a real-time system. If error-correction facilities are needed at the network interface level, an application would use the TCP or Stream Control Transmission Protocol (SCTP), which are designed for this purpose.

UDP uses UDP Port Numbers to identify sending and receiving application end-points on a host, or Internet sockets. Each side of a UDP connection may have an associated port number reserved by the sending or receiving application. However, unlike TCP, a source port is not required for UDP data packets. Packets are identified as belonging to a specific UDP connection by its combination of source host address, source port (if given), destination host address, and destination port.

Some UDP port numbers include:

- **FTP (Port 20),**
- **Encrypted SMTP (Port 26),**
- **and NTP (Port 123).**

IV. ENTERPRISE LEVEL SECURITY

A. Security Process Background

This work is part of a body of work for high-assurance enterprise computing using web services. The process has been developed over the last fifteen years and is termed ELS.

ELS is a capability designed to counter adversarial threats by protecting applications and data with a dynamic claims-based access control (CBAC) solution. ELS helps provide a high assurance environment in which information can be generated, exchanged, processed, and used. It is important to note that the ELS design is based on a set of high level tenets that are the overarching guidance for every decision made, from protocol selection to product configuration and use [30]. From there, a set of enterprise level requirements are formulated that conforms to the tenets and any high level guidance, policies and requirements.

B. Design Principles

The basic tenets, used at the outset of the ELS security model are the following:

0. The ***zeroth*** tenet is that the *malicious entities are present* and can look at network traffic and may attempt to modify that traffic by sending virus software to network assets. Current threat evaluation indicates that attacks are often successful at all levels; discovering these attacks and their consequences is problematic. In many cases attackers may compromise and infiltrate before a vulnerability can be mitigated by software changes (patches).

1. The ***first*** tenet is *simplicity*. Added features come at the cost of greater complexity, less understandability, greater difficulty in administration, higher cost, and/or lower adoption rates that may be unacceptable to the organization.

2. The ***second*** tenet, and closely related to the first, is *extensibility*. Any construct we put in place for an enclave should be extensible to the domain and the enterprise, and ultimately to cross-enterprise and coalition.

3. The ***third*** tenet is *information hiding*. Essentially, information hiding involves only revealing the minimum set of information to the requester and the outside world needed for making effective, authorized use of a capability.

4. The ***fourth*** tenet is *accountability*. In this context, accountability means being able to unambiguously identify and track what active entity in the enterprise performed any particular operation (e.g., accessed a file or IP address, invoked a service). Active entities include people, machines, and software process, all of which are named

registered and credentialed. By accountability we mean attribution with supporting evidence.

5. This ***fifth*** tenet is *minimal detail* (to only add detail to the solution to the required level). This combines the principles of simplicity and information hiding, and preserves flexibility of implementation at lower levels.

6. The ***sixth*** is the emphasis on a *service driven* rather than a product-driven solution whenever possible. Services should be separated as stated in the separation of function tenant. This also allows simplification and information hiding.

7. The ***seventh*** tenet is that *lines of authority* should be preserved and information assurance decisions should be made by policy and/or agreement at the appropriate level. An example here is that data owners should implement sharing requirements even when the requirements come from "higher authority."

8. The ***eighth*** tenet is *need-to-share* as overriding the need-to-know. Often effective health, defense, and finance rely upon and are ineffective without shared information. Shared does not mean released and the differences must be clear. However, judicious use of release authority and delegated access lead to a broader distribution of information. This leads to a more formalized delegation policy both within and outside of the enterprise. #

9. The ***ninth*** tenet is *separation of function*. This makes for fewer interfaces, easier updates, maintenance of least privilege, reduced and easier identified vulnerabilities and aids in forensics. #

10. The ***tenth*** tenet is *reliability*; security needs to work even if adversaries know how the process works. In setting up a large scale enterprise we need to publish exactly how things work. Personnel, computer operations people and vendors need to know how the system works and this should not create additional vulnerabilities.

11. The ***eleventh*** tenet is to *trust but verify* (and validate). Trust should be given out sparingly and even then trusted outputs need checking. Verification includes checking signature blocks, checking that the credential identities match (binding), checking the time stamps, checking to whom information is sent. Checking information received is identical to information sent, etc. Validation includes checking issuing authority, checking certificate validity, checking identity white lists and black lists. #

12. The ***twelfth*** tenet is *minimum attack surface*; the fewer the interfaces and the less the functionality in the interfaces, the smaller the exposure to threats.

13. The ***thirteenth*** tenet is *handle exceptions* and errors. Exception handling involves three basic aspects. The first is logging. The second is alerting and all security related events should be alerted to the Enterprise Support Desk (ESD). The third is notification to the user.

14. The ***fourteenth*** tenet is to *use proven solutions*. A carefully developed program of pilots and proofs of concepts has been pursued before elements were integrated into ELS. It is our intention to follow that process even when expediency dictates a quicker solution. Immediate implementation should always be accompanied by a roadmap for integration that includes this tenet.

15. The ***fifteenth*** tenet is *do not repeat old mistakes*. From a software point of view, this has many implications. First, never field a software solution with known vulnerabilities and exploits. There are several organizations that track the known vulnerabilities and exploits and an analysis against those indexes should be required of all software. Second, a flaw remediation system is required. After a vulnerability analysis, fixes may be required, after fielding, fixes will be required as new vulnerabilities and exploits are discovered. Third, from an operations standpoint take time to patch and repair, including outputs from the flaw remediation and improvements in Security Technical Implementation Guidelines.#

16.#

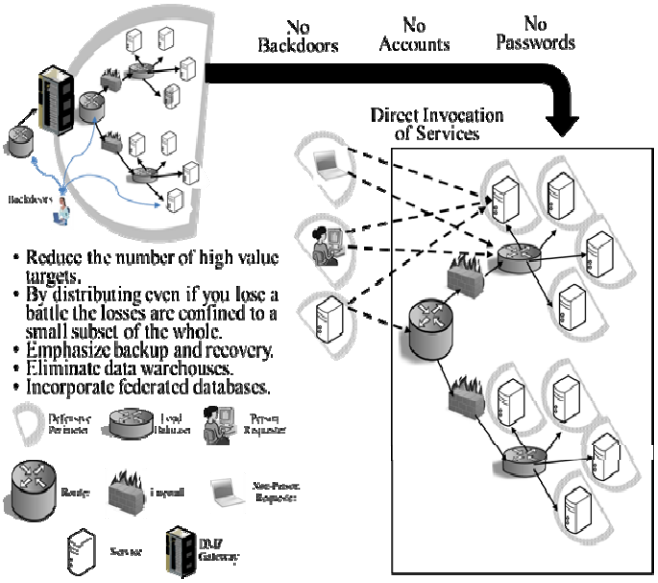

Figure 2 Distributed Security Architecture

Current paper-laden access control processes for an enterprise operation are plagued with ineffectiveness and inefficiencies. Given that in a number of enterprises tens of thousands of personnel transfer locations and duties annually, delays and security vulnerabilities are introduced daily into their operations. ELS mitigates security risks while eliminating much of the system administration required to manually grant and remove user/group permissions to specific applications/systems. Early calculations show that for government and defense 90-95% of recurring man-hours are saved and up to 3 weeks in delay for access request processing are eliminated by ELS-enabled applications [31]. While perimeter-based architecture assumes that threats are stopped at the front gates, ELS does not accept this precondition and is designed to mitigate many of the primary vulnerability points at the application using a distributed security architecture shown in Figure 2.

C. Security Principles

The ELS design addresses five security principles that are derived from the basic tenets:

- Know the Players – this is done by enforcing bi-lateral end-to-end authentication;
- Maintain Confidentiality – this entails end-to-end unbroken encryption (no in-transit decryption/payload inspection);
- Separate Access and Privilege from Identity – this is done by an authorization credential;
- Maintain Integrity – know that you received exactly what was sent;
- Require Explicit Accountability – monitor and log transactions.

Know the Players

In ELS, the identity certificate is an X.509 Public Key Infrastructure (PKI) certificate [32]. This identity is required for all active entities, both person and non-person, e.g., services, as shown in Figure 3. PKI certificates are verified and validated. Ownership is verified by a holder-of-key check. Supplemental (in combination with PKI) authentication factors may be required from certain entities, such as identity confirming information or biometric data.

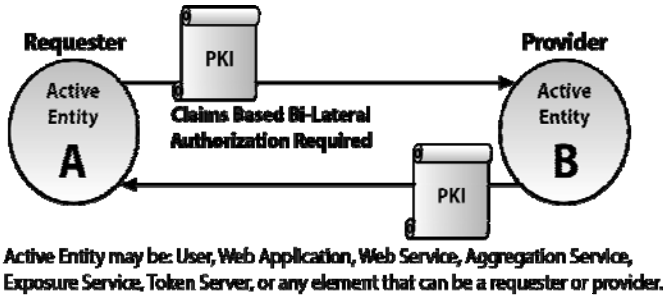

Figure 3 Bi-lateral Authentication

Maintain Confidentiality

Figure 4 shows that ELS establishes end-to-end Transport Layer Security (TLS) [33] encryption (and never gives away private keys that belong uniquely to the certificate holder).

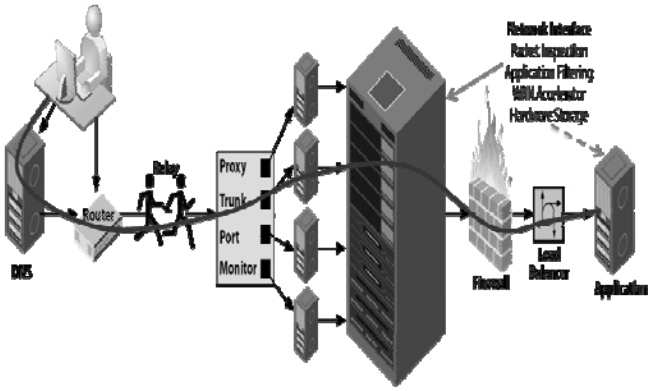

Figure 4 End-to-End Encryption

Separate Access and Privilege from Identity

ELS can accommodate changes in location, assignment and other attributes by separating the use of associated attributes from the identity. Whenever changes to attributes occur, claims are recomputed based on new associated attributes (see section III), allowing immediate access to required mission information. As shown in Figure 5, access control credentials utilize the Security Assertion Markup Language (SAML) (SAML authorization tokens differ from the more commonly used single-sign-on (SSO) tokens, and in ELS, are not used for authentication.) [34]. SAML tokens are created and signed by a Security Token Server (STS). The signatures are verified and validated before acceptance. The credentials of the signers also are verified and validated. The credential for access and privilege is bound to the requester by ensuring a match of the identity used in both authentication and authorization credentials.

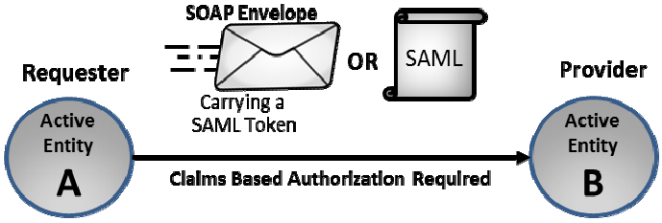

Figure 5 Claims-Based Authorization

Maintain Integrity

Integrity is implemented at the connection layer by end-to-end TLS message authentication codes (MACs), see Figure 6. Chained integrity, where trust is passed on transitively from one entity to another, is not used since it is not as strong as employing end-to-end integrity. At the application layer, packages (SAML tokens etc.) are signed, and signatures are verified and validated [35].

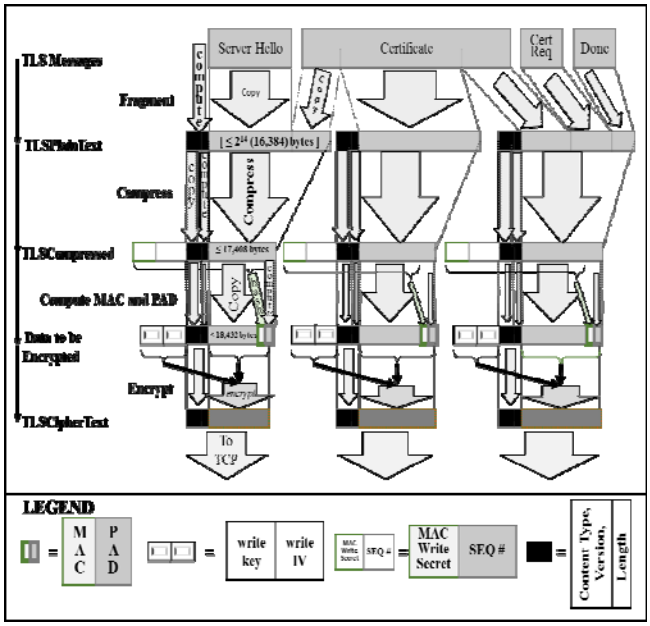

Figure 6 Integrity Measures

Require Explicit Accountability

All active entities with ELS are required to act on their own behalf (no proxies or impersonation allowed). As shown in Figure 7, ELS monitors specified activities for accountability and forensics. The monitor files are formatted in a standard way and stored locally. For enterprise files a monitor sweep agent reads, translates, cleans, and submits to an enterprise relational database for recording log records periodically, or on-demand. Local files are cleaned periodically to reduce overall storage and to provide a centralized repository for help desk, forensics, and other activities. The details of this activity are provided in [36, 37].

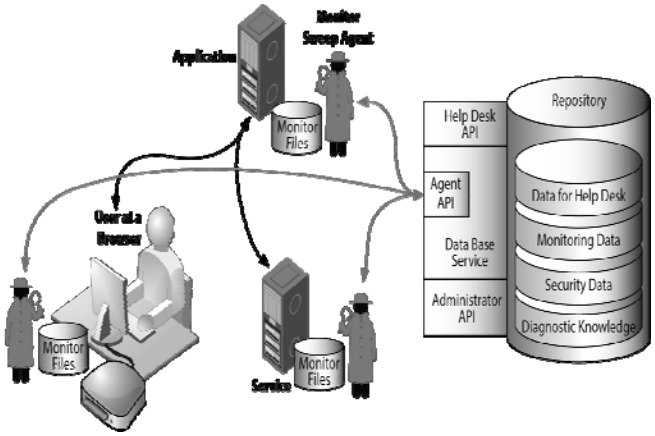

Figure 7 Accountability through Centralized Monitoring

V. THREATS CONSIDERED

Incoming ports are typically controlled, but outgoing ports are sometime left uncontrolled. If some ports are not explicitly blocked for both incoming and outgoing traffic, then it may be possible for malicious code to enter through a permitted port of an allowed service, but then to try to open or access other unused ports for malicious purposes, exfiltration of data, or reconnaissance. Restrictions should be applied to both incoming and outgoing messaging. In general, the policy should be to "deny all – permit by exception" to block all incoming and outgoing ports unless explicitly permitted. Closing of the internal ports means that the utility function ports are also blocked and the administrators must use the same allowed communication processes as any other active entities. This essentially closes the back doors. The bi-lateral authentication uses PKI credentials, eliminating passwords, and the authorization is done by a SAML claims credential eliminating the need for accounts.

At this point the content alone does not provide enough structure to achieve this approach. Many of the common protocols and services in use have known vulnerabilities and exploits and must either be prevented from operating in ELS or conditionally allowed with mitigations implemented. For example, FTP is known to have severe vulnerabilities and should not be used without mitigating actions. Some protocols are so vulnerable and dangerous that they provide unfettered entry to systems in some cases..

Once a list of all acceptable PPSs have been defined for an enterprise, it is necessary to correctly configure the security devices to allow only the permitted PPSs to pass through the enterprise network and to block all others. Constant monitoring of the networks and devices is required to ensure that only the approved PPS are allowed and that configurations have not been incorrectly modified, either by accident or by malicious intent. Since the collection of permissible PPSs and their mitigations are likely to evolve over time, this is a constant issue.

VI. ASSIGNING PORTS AND PROTOCOLS

From a technical standpoint any port can be assigned any protocol. From a practical standpoint that will only work if each user knows and agrees to use those combinations. For the internet protocol suite, the IANA is responsible for the global coordination of the DNS Root, IP addressing and other Internet protocol resources. This allows developers throughout the world to write their communication code to a standard set of ports and protocols and be reasonably assured that their communication will succeed. Annex A contains a list of official and unofficial port assignments for the commonly used ports from the IANA [2]. If this list appears daunting, remember that any protocol/port combination can be changed by mutual agreement and only requires that everybody reconfigure to the agreed combination. There are even lists of preferred service assignments. That it is important to control these is universally accepted.

The US Defense Department (DoD) has developed strict guidance on the control and management of protocols, ports, and services that can be used in national security information networks. The Department of Defense Instruction Number 8551.01, establishes policies, procedures and responsibilities for proper use of PPS [38]. In addition to the regulations concerning PPS use, the current instruction includes requirements for continuous, realtime monitoring, configuration management as well as better mechanisms for sharing information among the user community. The main points of the policy are:

- All PPS must be limited to those required for official business
- All PPS must be assessed for vulnerabilities and recommended security mitigations
- All PPS must be documented in a Category Assurance List (CAL)
- All PPS must be declared in a PPS Management Registry
- All PPS must be implemented according to procedures and policy developed by a Configuration Management Board (CMB)
- All PPS must be regulated according to ability to cause damage
- Boundary devices such as firewalls, routers, and intrusion/protection devices must be configured to allow only approved PPS
- PPS not implemented according to policy will be blocked with boundary devices
- An exception process will exist

The department CIO has overall responsibility for oversight of this instruction and the Defense Information Systems Agency is given the primary implementation responsibility.

In summary, all automated information systems (AIS) used on national security data networks must register the data communication modes, identifying the ports, protocols, and application services (PPS) used, and the network boundaries crossed. Compliance with the PPS requirements will reduce overall development time and cost, increase security, speed certification and accreditation steps, enhance AIS interoperability across the department, and speed operational deployment of all new and updated AIS.

VII. SERVER CONFIGURATIONS

Most servers come with default ports and protocols that include most of the services available to their broad class of users. For example, the IBM WebSphere would default to all of the common ports plus the IBM ports and protocols for all of their services, and perhaps Oracle, etc. In the enterprise, it is not sufficient to just use the defaults provided by the vendors, because these may include banned services or may not include recommended mitigations.

A port- by-port and protocol-by-protocol examination of the traffic generated by and accepted by a vendor product must be undertaken. This can be initiated by packet captures during normal operation. The valid and necessary traffic can be identified and remaining traffic analyzed to determine if it is needed or superfluous. After assessing normal traffic, a network scan for open ports will reveal other open ports that are not being used. These should typically be closed. In addition, detailed discussions with the vendor are required to understand what other ports and protocols may be open but not utilized during normal operations, as these are potential entry points for attackers.

VIII. FIREWALLS AND PORT BLOCKING

The network boundary protection devices, such as routers, firewalls, and intrusion detection/protection devices need to be configured to block all message traffic into the enterprise (reducing external flow to externally available ports) unless it is to or from permitted services on specific ports using permitted protocols. Internally available ports may be available as discussed in the conventional methods. However, these internal ports are the same as the external ports for ELS systems.

Conventional firewalls effectively control access to and from a requested service through ports and protocols filtering. A stateful firewall is a conventional firewall that also tracks connections by the socket pairs (source IP, source port, destination IP, destination port) and uses the port number of the source IP address to protect against the use of any other egress ports to exfiltrate data. Network firewalls protect the perimeter or boundary of a portion of the network using packet header filtering. The primary concern with network firewalls is to properly configure them to block all protocols except for the ones approved and needed for the services on the trusted side (server side) of the firewall. In addition, it is imperative to make sure the configuration is current with respect to the changing ports and protocols needs and the recommendations and banned services. In addition, the firewall appliance itself must be maintained in a secure condition with current updates and bug fixes.

A network firewall can operate in transparent (or passive) mode with respect to the end-to-end communication between a service requestor and the end-service if it does not break the end-to-end encryption. In transparent mode, the firewall is not able to decrypt the contents of an encrypted packet; it is only able to filter packets based on the packet header information that is in clear text. The alternative is a proxy firewall that breaks the end-to-end connection and operates as a man-in-the-middle. The proxy looks like the service endpoint to the requestor and is able to decrypt the incoming packets and encrypt the outgoing packets. This permits the firewall to perform content filtering on the decrypted packets.

Firewalls (and other security appliances) can be operated in inline filter mode or in observer mode (also known as promiscuous mode). An inline filter resides in the communication path and examines all packets in real time as they traverse the firewall before passing further into the network. An observer firewall is not in the direct communication path and examines a copy of the packet as it transits the firewall. The advantage of inline firewalls is that they can immediately block the first packet of a recognized attack, whereas in observer mode, the first (or first several) packet(s) will be passed to the destination before it can be blocked and damage prevented. The advantages of observer mode include real-time requirements being relaxed and that if the firewall goes down, communication is not halted.

The firewalls should block access to and from all ports that are accessible behind (the trusted side) the firewall except those that are explicitly permitted. This is called “deny all by default, permit by exception.” Firewalls that cover larger portions of the network or that front many subnets and host computers must be configured to allow any ports and protocols needed by any of the hosts on its trusted side.

Many firewall best-practices documents include details on firewall configurations (e.g., Cisco Firewall Best Practices Guide or the Defense Information Services Agency (DISA) Network Infrastructure Technology Review). For example, tunnels require special considerations to make sure packets embedded in the tunnels do not bypass the firewall. The functionality of a network firewall can be implemented as a separate security appliance that resides either in front of the application servers or in the endpoint hosts. In the latter case, each server would implement a packet header filter to perform ports and protocols filtering in its message handling process.

IX. APPLICATION FIREWALLS

Application Firewalls (AFWs) or application filters are designed to address the specific attacks on web applications and web services, which are not well addressed by other protection devices. AFWs that front applications can get more specific to the particular needs of the application and protect against attacks targeted at the application layer. For example, an AFW could be used to protect email, both incoming and outgoing to filter for damaging content or specific attachment types. Other types of application filters can examine the signatures on scripts (e.g., Java applets, JavaScript, ActiveX controls), the file extensions, virus scanning, blocking specific content, or use of specific commands.

In general, there are several different choices for deployment of AFWs:

1) as a separate hardware or software security appliance in front of the application,

2) as part of another security device such as a network firewall or content distribution controller,
3) as a cloud service, or
4) as an agent on the Application Server.

The current trend is for security appliances to integrate several functions in a single device to reduce operating costs and physical space requirements. The network firewall, intrusion detection/prevention, and application content filtering are being combined as integrated security appliances. While there are important benefits for this integration, the compromise of such a device could incapacitate all the protection functions at once.

X. NETWORK FIREWALLS IN ELS

In ELS a network firewall operates in transparent mode, does not decrypt the packets and is restricted to examining only the packet header. We note this is more restrictive than the capabilities being offered on many newer firewalls that offer more functionality but require the ability to decrypt the packet to examine its content. In ELS, network firewalls cannot operate as proxy firewalls or perform deep packet inspection since Transport Layer Security (TLS) with mutual authentication between requestor and service is a basic ELS requirement. In Figure 8, a network firewall positioned in front of several servers is shown to illustrate the use of such devices for ports and protocols filtering. The stateful firewall is shown protecting two web services implemented in two separate web servers with IP addresses IP1 and IP2. The firewall is configured to allow only requests to (IP address:port) combinations (IP1:443) and (IP2:443) and responses from them back to the requestor.

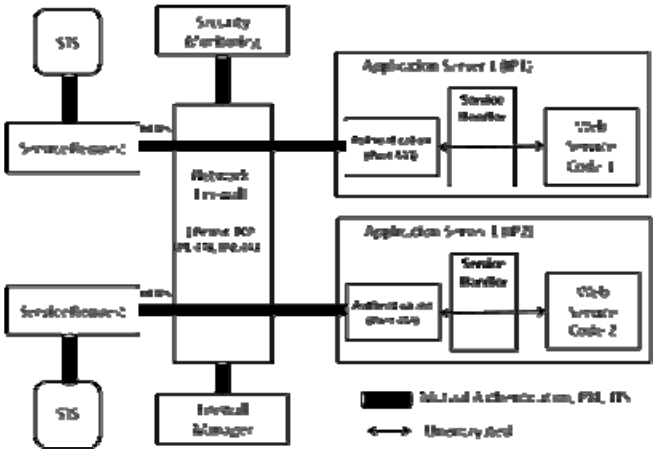

Figure 8 Network Firewall in Transparent Mode

If the web service requires access to services on other ports, then that communication must be routed through a firewall and this must be configured to permit packets on those ports.

XI. ENDPOINT PROTECTION IN ELS

In ELS, an agent-type model is preferred, one in which the packet header filtering and other security functions reside at the web server in the handler chain of the web service. The basic configuration of endpoint protection in ELS is shown in Figure 9 and provides a complete set of security functions for packet, message, and application layer security, tailored for the specific web service being protected. The new functions that are added in the server are packet header inspection, packet content inspection, message content inspection, and application protection. These functions implement the ports and protocols protection, as well as other security functions normally provided by network devices such as intrusion detection/protection, packet and message content filtering, deep packet inspection, and application/ web content filtering such as included in an application firewall.

A service requestor establishes communication with the server hosting the target web service according to the ELS practice using HTTPS. The packet is received by the destination sever and the packet header is immediately inspected to perform the ports and protocols blocking, source whitelist/ blacklist checking, and other filtering based on only the header, including stateful tracking of client addresses and ports. Until an HTTPS session has been established, only packets addressed to the server's IP address and port 443 are allowed. Other ports may be opened as needed as part of the web service following HTTPS establishment.

On the return path, the messages follow a similar process. In effect, the packet header inspection module can perform the required network-layer filtering and can block traffic based on ports and protocols (protocol, IP address, and port).

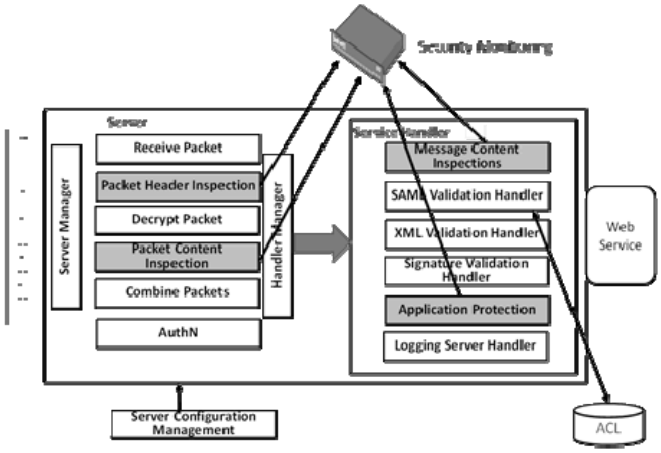

Figure 9 ELS Endpoint Security Functions

In the ELS endpoint protection architecture, the endpoint protection modules can be configured to communicate with additional security monitoring appliances, such as a NetScout , that can compile and track statistics about the security status of the server and the web service. The security appliances should be active entities and communicate with the server via TLS with mutual authentication. If required, the server could send the decrypted message traffic to other security appliances through this interface for additional security functions.

The endpoint protection functions are configured through the server configuration management interface, which communicates with the server by TLS with mutual

authentication. The ports and protocols and whitelist information and any software updates are provided through this interface.

It is recommended that the initial configuration of the packet header deny all ports and protocols, both incoming and outgoing (as opposed to the traditional incoming only), and that permissions be configured in as they are identified as needed.

XII. HANDLING AND INSPECTION OF TRAFFIC

Handling and inspection is done in software only modules in the server. The software functionality is embodied in handlers in the handler chain of the server as shown in Figure 10.

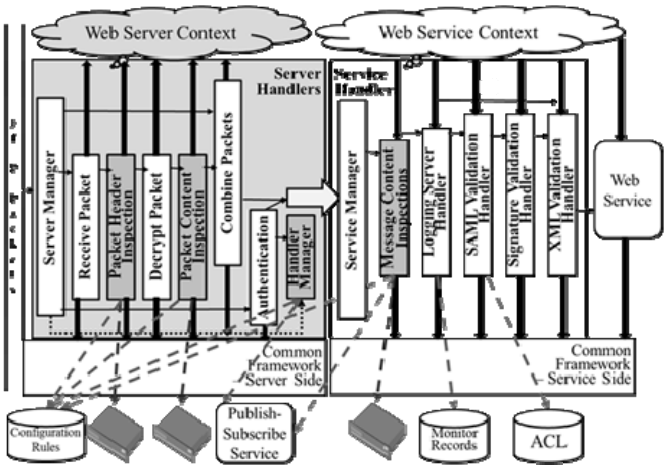

Figure 10 Server Side Handlers

Note that the handlers are embedded in the server handler chain at the point that the communication is prepared for their use, and that the functionality has been divided along those lines as opposed to the previous functionality such as virus scan, ports and protocols, intrusion detection or blacklist/whitelist, etc. These are distributed to packet header inspection, packet content inspection, and message content inspection. Each of these may perform inspection related to intrusion detection or blacklist blocking, etc. Pilots are being worked on, stay tuned for results. This is the preferred embodiment for enterprise applications. It moves the inspections to the point of the application itself, by inserting handlers within the server and service to do the inspections at the point it makes most sense. The inspections that can be done without decrypting the packets may be done at the front of the web server because they are passive entities. Moving inspections of decrypted traffic inside the server, not only preserves the end-to-end paradigm, but encapsulates the security and allows tailoring for the application itself. The encapsulated security with the application is virtualization ready.

XIII. ADDITIONAL SECURITY HARDENING

Since malicious software is assumed to be present, a request for service may come from within the enterprise bypassing firewalls, and not stating forbidden port numbers. To avoid the server software from finding a protocol resolution software set, and assign the port, all such software should be removed or not installed to begin with. The server software may come with a variety of software subsystems to satisfy a variety of customer needs such as telnet, secure shell, etc. If the allowable ports are known, the server software installation should not install other software if the installation procedure permits this. If the installation procedure does not allow this, or if the allowable ports and protocols are not worked out until after server software is installed, these non-allowable protocol software sets should be actively sought out and removed.

A more difficult option that is often not possible with off-the-shelf software is code reduction. Remove all code that implements functions that are not needed or desired. With Java, for example, remove unneeded JAR files or functions within JAR files to trim down to just code that is actually used. However, this may cause problems when updates are issued, since they revert to the "normal" set of JARs. This may require a special agreement with the vendor to support a specific configuration of their product (including testing all updates against this configuration), or manual intervention to apply updates and then remove unneeded parts and do regression testing that updates haven't changed what is needed/not needed.

XIV. SUMMARY

We have reviewed the ports and protocols used in the Internet model. We have also described the issues they raise and the vulnerabilities that may be introduced. For enterprise operations, having fewer ports open means a reduced attack space. We have also reviewed the specific requirements for an enterprise level security that is bi-laterally authenticated and encrypted end-to-end. This work is part of a body of work for high-assurance enterprise computing using web services. Elements of this work are described in [39-53]

.REFERENCES

[1] Kevin Foltz and William R. Simpson, "Enterprise Considerations for Ports and Protocols", Lecture Notes in Engineering and Computer Science: Proceedings World Congress on Engineering and Computer Science 2016, WCECS2016, 19-21 October, San Francisco, USA, pp. pp. 124-129.

[2] "Service Name and Transport Protocol Port Number Registry". The Internet Assigned Numbers Authority (IANA). "Service Name and Transport Protocol Port Number Registry"

[3] Michelle Cotton; Lars Eggert et al. (August 2011). Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry. IETF. BCP 165. RFC 6335. http://tools.ietf.org/html/rfc6335

[4] RFC 1312 Message Send Protocol 2 April 1992. http://tools.ietf.org/html/rfc1312

[5] RFC 2068 Hypertext Transfer Protocol HTTP/1.1. January 1997. http://tools.ietf.org/html/rfc2068

[6] RFC 1081 Post Office Protocol – Version 3 November 1988. http://tools.ietf.org/html/rfc1081

[7] RFC 3501 Internet Message Access Protocol – Version 4 rev1 March 2003. http://tools.ietf.org/html/rfc3501

[8] RFC 3411, An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks, Dec 2002. http://tools.ietf.org/html/rfc3411

[9] RFC 1132 A Standard for the Transmission of 802.2 Packets over IPX Networks, November 1989. http://tools.ietf.org/html/rfc1132
[10] RFC 2818 HTTP Over TLS May 2000. http://tools.ietf.org/html/rfc2818
[11] RFC 6409 Message Submission for Mail November 2011. http://tools.ietf.org/html/rfc6409
[12] Adobe proprietary, H. Parmar, M. Thornburgh (eds.) Adobe's Real Time Messaging Protocol, Adobe, December 21, 2012. http://www.adobe.com/content/dam/Adobe/en/devnet/rtmp/pdf/rtmp_specification_1.0.pdf
[13] Apple proprietary. https://en.wikipedia.org/wiki/Apple_Push_Notification_Service.
[14] Microsoft Siverlight, accessed 1 Sept 2015. https://en.wikipedia.org/wiki/Microsoft_Silverlight
[15] ISO/IEC 19464:2014, Information technology – Advanced Message Queuing Protocol (AMQP) v1.0 specification. http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=64955
[16] Avanset proprietary. http://www.avanset.com/purchase/vce-exam-simulator.html
[17] RFC4271Border Gateway Protocol 4 (BGP-4), January 2006. http://tools.ietf.org/html/rfc4271
[18] RFC 1035 Domain Names – Implementation and Specification, November 1987. http://tools.ietf.org/html/rfc1035
[19] RFC 959 File Transfer Protocol (FTP), October 1985. http://tools.ietf.org/html/rfc959
[20] RFC 4340 Datagram Congestion Control Protocol (DCCP) – March 2006. http://tools.ietf.org/html/rfc4340
[21] RFC 791Internet Protocol (IP) September 1981. http://tools.ietf.org/html/rfc791
[22] RFC 3540. Explicit Congestion Notification (ECN) - an extension to the Internet Protocol. [20] Robust Explicit Congestion Notification (ECN) Signaling with Nonces, June 2003. http://tools.ietf.org/html/rfc3540
[23] RFC 4945 The Internet IP Security PKI Profile of IKEv1/ISAKMP, IKEv2, and PKIX, August 2007. http://tools.ietf.org/html/rfc4945
[24] IEEE 802.3 Ethernet Working Group, accessed 9/1/2015. http://www.ieee802.org/3/
[25] Digital Subscriber Line, accessed 9/1/2015. https://en.wikipedia.org/wiki/Digital_subscriber_line
[26] RFC 2516 A Method for Transmitting PPP Over Ethernet (PPPoE), February 1999. http://tools.ietf.org/html/rfc2516
[27] RFC 1122 Requirements for Internet Hosts – Communication Layers, October 1989. http://tools.ietf.org/html/rfc1122
[28] RFC 1123 Requirements for Internet Hosts – Application and Support, October 1989. http://tools.ietf.org/html/rfc1123
[29] Margaret Rouse, OSI reference model (Open Systems Interconnection) definition, accessed 9/1/2015. http://searchnetworking.techtarget.com/definition/OSI
[30] William R. Simpson and Kevin Foltz, Proceedings of The 20th World Multi-Conference on Systemics, Cybernetics and Informatics: WMSCI, *"Enterprise Level Security - Basic Security Model"*, Volume I, WMSCI 2016, Orlando, Florida, 8-11 March 2016, pp. 56-61.
[31] Email from Rudy Rihani, Project Manager, Accenture Corporation, dated March 6, 2016, Subject: " manpower savings with ELS"
[32] X.509 Standards
a) DoDI 8520.2, Public Key Infrastructure (PKI) and Public Key (PK) Enabling, 24 May 2011
b) JTF-GNO CTO 06-02, Tasks for Phase I of PKI Implementation, 17 January 2006
c) X.509 Certificate Policy for the United States Department of Defense, Version 9.0, 9 February 2005
d) FPKI-Prof Federal PKI X.509 Certificate and CRL Extensions Profile, Version 6, 12 October 2005
e) RFC Internet X.509 Public Key Infrastructure: Certification Path Building, 2005
f) Public Key Cryptography Standard, PKCS #1 v2.2: RSA Cryptography Standard, RSA Laboratories, Oct 27, 2012
g) PKCS#12 format PKCS #12 v1.0: Personal Information Exchange Syntax Standard, RSA Laboratories, June 1999; http://www.rsa.com/rsalabs/node.asp?id=2138 PKCS 12 Technical Corrigendum 1, RSA laboratories, Feb 2000
[33] TLS family Internet Engineering Task Force (IETF) Standards
a) RFC 2830 Lightweight Directory Access Protocol (v3): Extension for Transport Layer Security, 2000-05
b) RFC 3749 Transport Layer Security Protocol Compression Methods, 2004-05
c) RFC 4279 Pre-Shared Key Ciphersuites for Transport Layer Security (TLS), 2005-12
d) RFC 5246 The Transport Layer Security (TLS) Protocol Version 1.2, 2008-08
e) RFC 5289 TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM), 2008-08
f) RFC 5929 Channel Bindings for TLS, 2010-07
g) RFC6358 Additional Master Secret Inputs TLS, 2012-01
h) RFC 7251 AES-CCM Elliptic Curve Cryptography (ECC) Cipher Suites for TLS, 2014-06
i) RFC 7301 Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension, 2014-07
j) RFC 7457 Summarizing Known Attacks on Transport Layer Security (TLS) and Datagram TLS (DTLS), 2015-02
[34] Organization for the Advancement of Structured Information Standards (OASIS) open set of Standards
a) N. Ragouzis et al., Security Assertion Markup Language (SAML) V2.0 Technical Overview, OASIS Committee Draft, March 2008
b) P. Mishra et al. Conformance Requirements for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS Standard, March 2005.
c) S. Cantor et al. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, March 2005
[35] William List and Rob Melville, IFIP Working Group 11.5, Integrity In Information, Computers and Security, Volume 13, Issue 4, pp. 295–301, Elsevier, doi:10.1016/0167-4048(94)90018-3, 1994.
[36] William R. Simpson and Coimbatore Chandersekaran, CCCT2010, pp. 84–89, "An Agent Based Monitoring System for Web Services," Orlando, FL, Apr 2011.
[37] William R. Simpson and Coimbatore Chandersekaran, 1st International Conference on Design, User Experience, and Usability, part of the 14th International Conference on Human-Computer Interaction (HCII 2011), "A Multi-Tiered Approach to Enterprise Support Services," 10 pp. Orlando, FL, July 2011. Also published in: A. Marcus (Ed.): Design, User Experience, and Usability, Pt I, HCII 2011, LNCS 6769, pp. 388–397, © Springer-Verlag Berlin Heidelberg 2011.
[38] Department of Defense Instruction Number 8551.0, Ports, Protocols, and Services Management (PPSM), May 28, 2014,

[39] William R. Simpson, Coimbatore Chandersekaran and Andrew Trice, "A Persona-Based Framework for Flexible Delegation and Least Privilege," Electronic Digest of the 2008 System and Software Technology Conference, Las Vegas, Nevada, May 2008.

[40] William R. Simpson, Coimbatore Chandersekaran and Andrew Trice, "Cross-Domain Solutions in an Era of Information Sharing," The 1st International Multi-Conference on Engineering and Technological Innovation: IMET2008, Volume I, Orlando, FL, June 2008, pp. 313–318.

[41] Coimbatore Chandersekaran and William R. Simpson, "The Case for Bi-lateral End-to-End Strong Authentication," World Wide Web Consortium (W3C) Workshop on Security Models for Device APIs, 4 pp., London, England, December 2008.

[42] William R. Simpson and Coimbatore Chandersekaran, "Information Sharing and Federation," The 2nd International Multi-Conf. on Engineering and Technological Innovation: IMETI2009, Volume I, Orlando, FL, July 2009, pp. 300–305.

[43] Coimbatore Chandersekaran and William R. Simpson, "A SAML Framework for Delegation, Attribution and Least Privilege," The 3rd International Multi-Conf. on Engineering and Technological Innovation: IMETI2010, Volume 2, pp. 303–308, Orlando, FL, July 2010.

[44] William R. Simpson and Coimbatore Chandersekaran, "Use Case Based Access Control," The 3rd International Multi-Conference on Engineering and Technological Innovation: IMETI2010, Volume 2, pp. 297–302, Orlando, FL, July 2010.

[45] Coimbatore Chandersekaran and William R. Simpson, "A Model for Delegation Based on Authentication and Authorization," The First International Conference on Computer Science and Information Technology (CCSIT-2011), Springer Verlag Berlin-Heildleberg, Lecture Notes in Computer Science, 20 pp.

[46] William R. Simpson and Coimbatore Chandersekaran, "An Agent Based Monitoring System for Web Services," The 16th International Command and Control Research and Technology Symposium: CCT2011, Volume II, Orlando, FL, April 2011, pp. 84–89.

[47] William R. Simpson and Coimbatore Chandersekaran, "An Agent-Based Web-Services Monitoring System," International Journal of Computer Technology and Application (IJCTA), Vol. 2, No. 9, September 2011, pp. 675–685.

[48] William R. Simpson, Coimbatore Chandersekaran and Ryan Wagner, "High Assurance Challenges for Cloud Computing," Lecture Notes in Engineering and Computer Science: Proceedings World Congress on Engineering and Computer Science 2011, WCECS 2011, 19–21 October 2011, San Francisco, USA, pp. 61–66.

[49] Coimbatore Chandersekaran and William R. Simpson, "Claims-Based Enterprise-Wide Access Control," Lecture Notes in Engineering and Computer Science: Proceedings World Congress on Engineering 2012, WCE 2012, 4-6 July 2012, London, U. K., pp. 524–529.

[50] William R. Simpson and Coimbatore Chandersekaran, "Assured Content Delivery in the Enterprise," Lecture Notes in Engineering and Computer Science: Proceedings World Congress on Engineering 2012, WCE 2012, 4–6 July 2012, London, U. K., pp. 555–560.

[51] William R. Simpson and Coimbatore Chandersekaran, "Enterprise High Assurance Scale-up," Lecture Notes in Engineering and Computer Science: Proceedings World Congress on Engineering and Computer Science 2012, WCECS 2012, 24-26 October 2012, San Francisco, USA, pp. 54–59.

[52] Coimbatore Chandersekaran and William R. Simpson, "A Uniform Claims-Based Access Control for the Enterprise," International Journal of Scientific Computing, Vol. 6, No. 2, December 2012, ISSN: 0973-578X, pp. 1–23.

[53] Simpson, William R., CRC Press, "Enterprise Level Security – Securing Information Systems in an Uncertain World", by Auerbach Publications, ISBN 9781498764452, May 2016, 397 pp.

Appendix
Standard Port Numbers and Protocols

This is a list of well-known Internet socket port numbers used by protocols of the Transport Layer of the Internet Protocol Suite for the establishment of host-to-host connectivity. Originally, these port numbers were used by the Network Control Program (NCP) and two ports were needed as transmission was done at half duplex. As Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) were adopted, only one port was needed. The even numbered ports were dropped. This is why some even numbers in the well-known port number range are unassigned. TCP and UDP port numbers are also used for the Stream Control Transmission Protocol (SCTP), and the Datagram Congestion Control Protocol (DCCP). SCTP and DCCP services usually use a port number that matches the service of the corresponding TCP or UDP implementation if they exist. The Internet Assigned Numbers Authority (IANA) is responsible for maintaining the official assignments of port numbers for specific uses. However, many unofficial uses of both well-known and registered port numbers occur in practice.

Table A- 1 Legend for Ports and Protocols

Use	Description
Official	Port is registered with IANA for the application
Unofficial	Port is not registered with IANA for the application
Multiple use	Multiple applications are known to use this port.

Well-known ports

The port numbers in the range from 0 to 1023 are the well-known ports or system ports. They are used by system processes that provide widely used types of network services. On Unix-like operating systems, a process must execute with superuser privileges to be able to bind a network socket to an IP address using one of the well-known ports.

Table A- 2 Well Known Ports

Port	TCP	UDP	Description	Status
0	TCP		Programming technique for specifying system-allocated (dynamic) ports	Unofficial
0		UDP	Reserved	Official
1	TCP	UDP	TCP Port Service Multiplexer (TCPMUX)	Official
2	TCP	UDP	CompressNET Management Utility	Official

3	TCP	UDP	CompressNET Compression Process	Official
4	TCP	UDP	Unassigned	Official
5	TCP	UDP	Remote Job Entry	Official
6	TCP	UDP	Unassigned	Official
7	TCP	UDP	Echo Protocol/ Ping/ ICMP	Official
8	TCP	UDP	Unassigned	Official
9	TCP	UDP	Discard Protocol	Official
9		UDP	Wake-on-LAN	Unofficial
10	TCP	UDP	Unassigned	Official
11	TCP	UDP	Active Users (systat service)	Official
12	TCP	UDP	Unassigned	Official
13	TCP	UDP	Daytime Protocol (RFC 867)	Official
14	TCP	UDP	Unassigned	Official
15	TCP	UDP	Previously netstat service	Unofficial
16	TCP	UDP	Unassigned	Official
17	TCP	UDP	Quote of the Day	Official
18	TCP	UDP	Message Send Protocol	Official
19	TCP	UDP	Character Generator Protocol (CHARGEN)	Official
20	TCP	UDP	FTP data transfer	Official
21	TCP		FTP control (command)	Official
22	TCP	UDP	Secure Shell (SSH)—used for secure logins, file transfers (scp, sftp) and port forwarding	Official
23	TCP	UDP	Telnet protocol—unencrypted text communications	Official
24	TCP	UDP	Priv-mail : any private mail system.	Official
25	TCP		Simple Mail Transfer Protocol (SMTP)—used for e-mail routing between mail servers	Official
26	TCP	UDP	encrypted SMTP	Official
27	TCP	UDP	NSW User System FE	Official
29	TCP	UDP	MSG ICP	Official
33	TCP	UDP	Display Support Protocol	Official
35	TCP	UDP	Any private printer server protocol	Official
37	TCP	UDP	TIME protocol	Official
39	TCP	UDP	Resource Location Protocol (RLP)—used for determining the location of higher level services from hosts on a network	Official
40	TCP	UDP	Unassigned	Official
42	TCP	UDP	ARPA Host Name Server Protocol	Official
42	TCP	UDP	Windows Internet Name Service	Unofficial
43	TCP		WHOIS protocol	Official
47	TCP	UDP	NI FTP	Official
49	TCP	UDP	TACACS Login Host protocol	Official
50	TCP	UDP	Remote Mail Checking Protocol	Official
51	TCP	UDP	IMP Logical Address Maintenance	Official
52	TCP	UDP	XNS (Xerox Network Systems) Time Protocol	Official
53	TCP	UDP	Domain Name System (DNS)	Official
54	TCP	UDP	XNS (Xerox Network Systems) Clearinghouse	Official
55	TCP	UDP	ISI Graphics Language (ISI-GL)	Official
56	TCP	UDP	XNS (Xerox Network Systems) Authentication	Official
56	TCP	UDP	Route Access Protocol (RAP)	Unofficial
57	TCP		Mail Transfer Protocol (RFC 780)	Official
58	TCP	UDP	XNS (Xerox Network Systems) Mail	Official
64	TCP	UDP	CI (Travelport) (formerly Covia) Comms Integrator	Official
67		UDP	Bootstrap Protocol (BOOTP) Server; also used by Dynamic Host Configuration Protocol (DHCP)	Official
68		UDP	Bootstrap Protocol (BOOTP) Client; also used by Dynamic Host Configuration Protocol (DHCP)	Official
69		UDP	Trivial File Transfer Protocol (TFTP)	Official
70	TCP		Gopher protocol	Official
71	TCP		NETRJS protocol	Official
72	TCP		NETRJS protocol	Official
73	TCP		NETRJS protocol	Official
74	TCP		NETRJS protocol	Official
77	TCP	UDP	Any private Remote Job Entry	Official
79	TCP		Finger protocol	Official
80	TCP		Hypertext Transfer Protocol (HTTP)	Official
80		UDP	QUIC (from Chromium) for HTTP	Unofficial
81	TCP		Torpark—Onion routing	Unofficial
82		UDP	Torpark—Control	Unofficial
88	TCP	UDP	Kerberos—authentication system	Official
90	TCP	UDP	DNSIX (DoD Network Security for Information Exchange) Security Attribute Token Map	Official
90	TCP	UDP	PointCast (dotcom)	Unofficial
99	TCP		WIP Message protocol	Unofficial
100		UDP	CyberGate RAT protocol	Unofficial
101	TCP		NIC host name	Official
102	TCP		ISO-TSAP (Transport Service Access Point) Class 0 protocol; also used by Digital Equipment Corporation DECnet (Phase V+) over TCP/IP	Official
104	TCP	UDP	ACR/NEMA Digital Imaging and Communications in Medicine (DICOM)	Official
105	TCP	UDP	CCSO Nameserver Protocol (Qi/Ph)	Official
107	TCP		Remote TELNET Service protocol	Official
108	TCP	UDP	SNA Gateway Access Server [	Official
109	TCP		Post Office Protocol v2 (POP2)	Official
110	TCP		Post Office Protocol v3 (POP3)	Official
111	TCP	UDP	ONC RPC (Sun RPC)	Official
113	TCP		Ident—Authentication Service/Identification Protocol, used by IRC servers to identify users	Official
113		UDP	Authentication Service (auth)	Official
115	TCP		Simple File Transfer Protocol (SFTP)	Official
117	STD		UUCP Path Service	Official
118	TCP	UDP	SQL (Structured Query Language) Services	Official
119	TCP		Network News Transfer Protocol (NNTP)—retrieval of newsgroup messages	Official
123		UDP	Network Time Protocol (NTP)—used for time synchronization	Official
126	TCP	UDP	Formerly Unisys Unitary Login, renamed by Unisys to NXEdit. Used by Unisys Programmer's Workbench for Clearpath MCP, an IDE for Unisys MCP software development	Official
135	TCP	UDP	DCE endpoint resolution	Official
135	TCP	UDP	Microsoft EPMAP (End Point Mapper), also known as DCE/RPC Locator service, used to remotely manage services including DHCP server, DNS server and WINS. Also used by DCOM	Unofficial
137	TCP	UDP	NetBIOS NetBIOS Name Service	Official
138	TCP	UDP	NetBIOS NetBIOS Datagram Service	Official
139	TCP	UDP	NetBIOS NetBIOS Session Service	Official
143	TCP		Internet Message Access Protocol (IMAP)—management of email messages	Official
152	TCP	UDP	Background File Transfer Program (BFTP)	Official
153	TCP	UDP	SGMP, Simple Gateway Monitoring Protocol	Official
156	TCP	UDP	SQL Service	Official
158	TCP	UDP	DMSP, Distributed Mail Service Protocol	Unofficial
161		UDP	Simple Network Management Protocol (SNMP)	Official
162	TCP	UDP	Simple Network Management Protocol Trap (SNMPTRAP)	Official
170	TCP		Print-srv, Network PostScript	Official

175	TCP		VMNET (IBM z/VM, z/OS & z/VSE—Network Job Entry (NJE))	Official
177	TCP	UDP	X Display Manager Control Protocol (XDMCP)	Official
179	TCP		BGP (Border Gateway Protocol)	Official
194	TCP	UDP	Internet Relay Chat (IRC)	Official
199	TCP	UDP	SMUX, SNMP Unix Multiplexer	Official
201	TCP	UDP	AppleTalk Routing Maintenance	Official
209	TCP	UDP	The Quick Mail Transfer Protocol	Official
210	TCP	UDP	ANSI Z39.50	Official
213	TCP	UDP	Internetwork Packet Exchange (IPX)	Official
218	TCP	UDP	Message posting protocol (MPP)	Official
220	TCP	UDP	Internet Message Access Protocol (IMAP), version 3	Official
259	TCP	UDP	ESRO, Efficient Short Remote Operations	Official
264	TCP	UDP	BGMP, Border Gateway Multicast Protocol	Official
280	TCP	UDP	http-mgmt	Official
300	TCP		ThinLinc Web Access	Unofficial
308	TCP		Novastor Online Backup	Official
311	TCP		Mac OS X Server Admin (officially AppleShare IP Web administration)	Official
318	TCP	UDP	PKIX TSP, Time Stamp Protocol	Official
319		UDP	Precision Time Protocol event messages	Official
320		UDP	Precision Time Protocol general messages	Official
350	TCP	UDP	MATIP-Type A, Mapping of Airline Traffic over Internet Protocol	Official
351	TCP	UDP	MATIP-Type B, Mapping of Airline Traffic over Internet Protocol	Official
366	TCP	UDP	ODMR, On-Demand Mail Relay	Official
369	TCP	UDP	Rpc2portmap	Official
370	TCP		codaauth2—Coda authentication server	Official
370		UDP	codaauth2—Coda authentication server	Official
370		UDP	securecast1—Outgoing packets to NAI's SecureCast servers As of 2000[update]	Unofficial
371	TCP	UDP	ClearCase albd	Official
383	TCP	UDP	HP data alarm manager	Official
384	TCP	UDP	A Remote Network Server System	Official
387	TCP	UDP	AURP, AppleTalk Update-based Routing Protocol	Official
389	TCP	UDP	Lightweight Directory Access Protocol (LDAP)	Official
399	TCP	UDP	Digital Equipment Corporation DECnet (Phase V+) over TCP/IP	Official
401	TCP	UDP	UPS Uninterruptible Power Supply	Official
427	TCP	UDP	Service Location Protocol (SLP)	Official
433	TCP	UDP	NNSP, part of Network News Transfer Protocol	Official
443	TCP		Hypertext Transfer Protocol over TLS/SSL (HTTPS)	Official
443		UDP	QUIC (from Chromium) for HTTPS	Unofficial
444	TCP	UDP	SNPP, Simple Network Paging Protocol (RFC 1568)	Official
445	TCP		Microsoft-DS Active Directory, Windows shares	Official
445	TCP		Microsoft-DS SMB file sharing	Official
464	TCP	UDP	Kerberos Change/Set password	Official
465	TCP		URL Rendezvous Directory for SSM (Cisco protocol)	Official
465	TCP		Simple Mail Transfer Protocol over TLS/SSL (SMTPS)	Unofficial
475	TCP	UDP	tcpnethaspsrv (Aladdin Knowledge Systems Hasp services, TCP/IP version)	Official
491	TCP		GO-Global remote access and application publishing software	Unofficial
497	TCP		Dantz Retrospect	Official
500	TCP	UDP	Internet Security Association and Key Management Protocol (ISAKMP)	Official
502	TCP	UDP	Modbus, Protocol	Official
504	TCP	UDP	Citadel—multiservice protocol for dedicated clients for the Citadel groupware system	Official
512	TCP		Rexec, Remote Process Execution	Official
512		UDP	comsat, together with biff	Official
513	TCP		rlogin	Official
513		UDP	Who	Official
514	TCP		Shell—used to execute non-interactive commands on a remote system (Remote Shell, rsh, remsh)	Official
514		UDP	Syslog—used for system logging	Official
515	TCP		Line Printer Daemon—print service	Official
517		UDP	Talk	Official
518		UDP	NTalk	Official
520	TCP		efs, extended file name server	Official
520		UDP	Routing Information Protocol (RIP)	Official
521		UDP	Routing Information Protocol Next Generation (RIPng)	Official
524	TCP	UDP	NetWare Core Protocol (NCP) is used for a variety things such as access to primary NetWare server resources, Time Synchronization, etc.	Official
525		UDP	Timed, Timeserver	Official
530	TCP	UDP	RPC	Official
531	TCP	UDP	AOL Instant Messenger	Unofficial
532	TCP		netnews	Official
533		UDP	netwall, For Emergency Broadcasts	Official
540	TCP		UUCP (Unix-to-Unix Copy Protocol)	Official
542	TCP	UDP	commerce (Commerce Applications)	Official
543	TCP		klogin, Kerberos login	Official
544	TCP		kshell, Kerberos Remote shell	Official
545	TCP		OSIsoft PI (VMS), OSISoft PI Server Client Access	Unofficial
546	TCP	UDP	DHCPv6 client	Official
547	TCP	UDP	DHCPv6 server	Official
548	TCP		Apple Filing Protocol (AFP) over TCP	Official
550	TCP	UDP	new-rwho, new-who	Official
554	TCP	UDP	Real Time Streaming Protocol (RTSP)	Official
556	TCP		Remotefs, RFS, rfs_server	Official
560		UDP	rmonitor, Remote Monitor	Official
561		UDP	monitor	Official
563	TCP	UDP	NNTP protocol over TLS/SSL (NNTPS)	Official
587	TCP		e-mail message submission (SMTP)	Official
591	TCP		FileMaker 6.0 (and later) Web Sharing (HTTP Alternate, also see port 80)	Official
593	TCP	UDP	HTTP RPC Ep Map, Remote procedure call over Hypertext Transfer Protocol, often used by Distributed Component Object Model services and Microsoft Exchange Server	Official
604	TCP		TUNNEL profile, a protocol for BEEP peers to form an application layer tunnel	Official
623		UDP	ASF Remote Management and Control Protocol (ASF-RMCP)	Official
631	TCP	UDP	Internet Printing Protocol (IPP)	Official
631	TCP	UDP	Common Unix Printing System (CUPS)	Unofficial
635	TCP	UDP	RLZ DBase	Official
636	TCP	UDP	Lightweight Directory Access Protocol over TLS/SSL (LDAPS)	Official
639	TCP	UDP	MSDP, Multicast Source Discovery Protocol	Official

641	TCP	UDP	SupportSoft Nexus Remote Command (control/listening): A proxy gateway connecting remote control traffic	Official
646	TCP	UDP	LDP, Label Distribution Protocol, a routing protocol used in MPLS networks	Official
647	TCP		DHCP Failover protocol	Official
648	TCP		RRP (Registry Registrar Protocol)	Official
651	TCP	UDP	IEEE-MMS	Official
653	TCP	UDP	SupportSoft Nexus Remote Command (data): A proxy gateway connecting remote control traffic	Official
654	TCP		Media Management System (MMS) Media Management Protocol (MMP)	Official
657	TCP	UDP	IBM RMC (Remote monitoring and Control) protocol, used by System p5 AIX Integrated Virtualization Manager (IVM) and Hardware Management Console to connect managed logical partitions (LPAR) to enable dynamic partition reconfiguration	Official
660	TCP		Mac OS X Server administration	Official
666	TCP	UDP	Doom, first online first-person shooter	Official
666	TCP		airserv-ng, aircrack-ng's server for remote-controlling wireless devices	Unofficial
674	TCP		ACAP (Application Configuration Access Protocol)	Official
688	TCP	UDP	REALM-RUSD (ApplianceWare Server Appliance Management Protocol)	Official
691	TCP		MS Exchange Routing	Official
694	TCP	UDP	Linux-HA High availability Heartbeat	Official
695	TCP		IEEE-MMS-SSL (IEEE Media Management System over SSL)	Official
698		UDP	OLSR (Optimized Link State Routing)	Official
700	TCP		EPP (Extensible Provisioning Protocol), a protocol for communication between domain name registries and registrars (RFC 5734)	Official
701	TCP		LMP (Link Management Protocol (Internet)), a protocol that runs between a pair of nodes and is used to manage traffic engineering (TE) links	Official
702	TCP		IRIS (Internet Registry Information Service) over BEEP (Blocks Extensible Exchange Protocol) (RFC 3983)	Official
706	TCP		Secure Internet Live Conferencing (SILC)	Official
711	TCP		Cisco Tag Distribution Protocol—being replaced by the MPLS Label Distribution Protocol	Official
712	TCP		Topology Broadcast based on Reverse-Path Forwarding routing protocol (TBRPF) (RFC 3684)	Official
749	TCP	UDP	Kerberos (protocol) administration	Official
750		UDP	kerberos-iv, Kerberos version IV	Official
751	TCP	UDP	kerberos_master, Kerberos authentication	Unofficial
752		UDP	passwd_server, Kerberos Password (kpasswd) server	Unofficial
753	TCP		Reverse Routing Header (rrh)	Official
753		UDP	Reverse Routing Header (rrh)	Official
753		UDP	userreg_server, Kerberos userreg server	Unofficial
754	TCP		tell send	Official
754	TCP		krb5_prop, Kerberos v5 slave	Unofficial
			propagation	
754		UDP	tell send	Official
760	TCP	UDP	krbupdate [kreg], Kerberos registration	Unofficial
782	TCP		Conserver serial-console management server	Unofficial
783	TCP		SpamAssassin spamd daemon	Unofficial
800		UDP	mdbe daemon	Official
808	TCP		Microsoft Net.TCP Port Sharing Service	Official
829	TCP		Certificate Management Protocol	Unofficial
843	TCP		Adobe Flash	Unofficial
847	TCP		DHCP Failover protocol	Official
848	TCP	UDP	Group Domain Of Interpretation (GDOI) protocol	Official
860	TCP		iSCSI (RFC 3720)	Official
861	TCP	UDP	OWAMP control (RFC 4656)	Official
862	TCP	UDP	TWAMP control (RFC 5357)	Official
873	TCP		rsync file synchronization protocol	Official
888	TCP		cddbp, CD DataBase (CDDB) protocol (CDDBP), IBM Endpoint Manager Remote Control	Unofficial
897	TCP	UDP	Brocade SMI-S RPC	Unofficial
898	TCP	UDP	Brocade SMI-S RPC SSL	Unofficial
901	TCP		Samba Web Administration Tool (SWAT)	Unofficial
901	TCP	UDP	VMware Virtual Infrastructure Client (from managed device to management console)	Unofficial
902	TCP	UDP	ideafarm-door	Official
902	TCP	UDP	VMware Server Console (from management console to managed device)	Unofficial
903	TCP		VMware Remote Console	Unofficial
904	TCP		VMware Server Alternate (if 902 is in use, i.e. SUSE linux)	Unofficial
911	TCP		Network Console on Acid (NCA)—local tty redirection over OpenSSH	Unofficial
944		UDP	Network File System Service	Unofficial
953	TCP	UDP	Domain Name System (DNS) RNDC Service	Unofficial
973		UDP	Network File System over IPv6 Service	Unofficial
981	TCP		SofaWare Technologies Remote HTTPS management for firewall devices running embedded Check Point FireWall-1 software	Unofficial
987	TCP		Microsoft Windows SBS SharePoint	Unofficial
989	TCP	UDP	FTPS Protocol (data): FTP over TLS/SSL	Official
990	TCP	UDP	FTPS Protocol (control): FTP over TLS/SSL	Official
991	TCP	UDP	NAS (Netnews Administration System)	Official
992	TCP	UDP	TELNET protocol over TLS/SSL	Official
993	TCP		Internet Message Access Protocol over TLS/SSL (IMAPS)	Official
994	TCP	UDP	Internet Relay Chat over TLS/SSL (IRCS)	Official
995	TCP		Post Office Protocol 3 over TLS/SSL (POP3S)	Official
999	TCP		ScimoreDB Database System	Unofficial
1002	TCP		Opsware agent (aka cogbot)	Unofficial
1010	TCP		ThinLinc Web Administration	Unofficial
1023	TCP	UDP	Reserved	Official

Registered ports

The range of port numbers from 1024 to 49151 are the registered ports. They are assigned by IANA for specific service upon application by a requesting entity. On most systems, registered ports can be used by ordinary users. See the Service Name and Transport Protocol Port Number Registry of IANA for the complete list of assigned ports [52].