Security and the Sun™ Java System Web Server

White Paper
October 2003

Table of Contents

Introduction **1**
Security By Design **2**
Secure Sockets Layer 2
External Modules 3
SSL Administration 3
Access Control 3
Specifying Restricted Resources 4
Specifying Authorization Constraints 4
Access Control Through .htaccess 6
Virtual Server Support 6
Architectural Security 6
Sun Java System Web Server Resistance to Attacks 7
Security By Default **8**
Deployment Security **9**
Running in chroot Environments 9
Multiplatform Benefits 9
Web Application Security **11**
Administration Security **12**
Special Administration Access Controls 12
Tested Security **13**
Track Record **14**

Chapter 1

Introduction

This document provides a brief overview of the security services and related aspects of the Sun™ Java System Web Server (formerly Sun ONE Web Server). Links are provided to documentation that should be consulted for a more detailed description of the features summarized in this paper.

Chapter 2

Security By Design

Secure Sockets Layer

The Java System Web Server fully supports both Secure Sockets Layer[1] (SSL) v2 and v3 and Transport Layer Security (TLS) with all common algorithm combinations. Java System Web Server security functionality is implemented using the Network Security Services (NSS[2]) libraries. Through NSS, the Java System Web Server supports a wide range of security standards and functionality, as well as gaining the stability and performance of a well-tested and widely used implementation.

In order to match the security needs of various Web services, virtual servers, and server instances, which may be needed in customer installations, Java System Web Server SSL support is highly configurable. It can manage multiple server certificates that can be mapped to various server instances, virtual servers, or listen ports for maximum deployment flexibility.

In general, Java System Web Server instances may listen on any number of ports, and each listen port can be defined to be SSL enabled or not. For each listen port where SSL is enabled, the implementation and behavior parameters can be individually configured. Some of the configurable attributes include:

- Which server certificate will be used to identify the server on a given port
- Which crypto module is used to process the requests on a given port
- Which digest/encryption/signature algorithms are supported on a given port
- Whether client authentication (two-way SSL authentication) is required on a given port

1. sun.com/docs/manuals/security/sslin/index.htm
2. mozilla.org/projects/security/pki/nss/

- Whether authenticating clients must be known users present in the LDAP user database (see authentication section later in this document) on that port
- File paths may be specified that require a certain level of protection on the connection, based on active encryption key size (this enables the user to prevent certain documents from being transmitted over a channel deemed insufficiently secure for the type of document)

External Modules

Crypto modules that implement the PKCS#11 interface can be plugged into the Java System Web Server, thus delegating some or all storage and processing to such external modules. Depending on the specific capabilities of the external module, this can be done for a number of reasons, including:

- Cryptographically strong hardware-based random number generators for key generation assure good key material.
- Tamper-resistant hardware for storing cryptographic key material provides increased protection for private keys versus storing them in software.
- Off-loading computationally intensive crypto processing to the external module, often powered by hardware crypto accelerators, delivers performance benefits.
- Implementation or enforcement of customized key generation, storage, usage, or backup policies provides an extra level of protection for the essential private keys used in SSL.

SSL Administration

The Java System Web Server administration server provides a convenient user interface for managing all aspects of the certificate life cycle. Some of the certificate-handling UI functionality includes:

- Server key generation and certificate request generation
- Automated certificate request transmission to a Certificate Authority (CA); all well-known public CAs are preconfigured, or the administrator can specify their own
- Receipt and storage of certificate from CA; certificates and keys may be stored in the built-in, file-based implementation or using any mechanisms provided by external modules, as described previously
- Processing Certificate Revocation Lists (CRLs) and Compromised Key Lists (CKLs)

For step-by-step procedures on how to use these services, refer to the *Securing Your Web Server*[3] section of the *Sun Java System Web Server Administrators Guide*.

Access Control

The Java System Web Server supports flexible access control configurations that enable administrators to restrict access to server resources based on a number of possible criteria. Access control rules are specified in Access Control List (ACL) files. Each ACL entry consists of one or more Access Control Entries (ACEs). ACL entries define what is protected and what is required to gain access. The following sections describe the various combinations supported by the Java System Web Server.

For step-by-step details on how to configure Java System Web Server access control functionality, refer to the *Access Control*[4] section of the *Sun Java System Web Server* Administrators Guide.

3. docs.sun.com/docs/manuals/enterprise/60sp1/ag/esecurty.htm
4. docs.sun.com/docs/manuals/enterprise/60sp1/ag/esaccess.htm

 © 2003 Sun Microsystems, Inc.

Specifying Restricted Resources

In order to design an access control policy, the first step is to categorize which resources available from the Web server are to be protected in some way. The Java System Web Server supports multiple selection criteria that can be used to mark a set of resources as protected:

- Global — Global restrictions that apply to all requests across all managed instances
- By server instance — Restrictions that apply to a specific Java System Web Server instance
- By virtual server— Virtual server restrictions that can be specified simply by stating restrictions on the appropriate file paths (for more distributed virtual server administration, each virtual server can be assigned an individual ACL file that can contain its independent set of access restrictions)
- By file path — Specifies restrictions based on file system paths
- By URI path— Specifies restrictions based on accessed URI paths
- By file type (extension) — Specifies restrictions based on file type extensions

Specifying Authorization Constraints

The next step in designing an access control policy is to decide, for each protected resource, what conditions must be met in order to gain access to it. Again, the Java System Web Server offers several possible decision criteria:

- Client host — Only named hosts or IP addresses are allowed access to the controlled resource. Wild card patterns can be given to match specific subnets or DNS namespaces.
- Time — Access is allowed only during given time periods or days.
- Connection type — A constraint can be given to require an SSL connection in order to grant the authorization. This can be used to assure that certain content will never be transmitted in the clear. Specific SSL/TLS ciphers can also be required on the connection in order to grant access.
- Authenticated user or group — Access is granted if the request is coming from any authenticated user, a specified user, or a user belonging to a given group. In order to establish whether this is the case, the requesting user must be authenticated. Supported authentication mechanisms are discussed in the next section.
- ACL syntax — While the existing ACL syntax is quite flexible, it cannot cover all possible conditions an enterprise may wish to specify. For this reason, the Java System Web Server supports custom loadable authentication service (LAS) modules that can be developed by the customer to address any special needs.

For details on how to develop custom authentication modules to be used in ACL definitions, refer to the *Access Control Programmer's Guide*[5].

Authentication Support

Authentication support requires that the Java System Web Server be configured to access an LDAP server. The Java System Directory Server, OpenLDAP, and even Microsoft Directory can be used for user and group information. The Java System Web Server supports using SSL on the connections to the LDAP server in order to protect the user data over the wire.

5. docs.sun.com/docs/manuals/enterprise/50/pg/1-intro.htm#13631

Several mechanisms are supported for establishing an authenticated identity for the request being processed, including:

- HTTP Basic — As defined in RFC 2617[6]. If basic authentication is configured for a resource, the Java System Web Server intercepts the requests and responds with the authentication request, prompting the client browser to request user and password from the user. This user login information is validated against the configured LDAP server. This is the most widely used Web authentication protocol, however, unless combined with session encryption (e.g., SSL), it is not secure, because login information is sent in the clear.
- HTTP Digest — As defined in RFC 2617. Similar to basic authentication, but the client computes a hash of its login information combined with some request-specific data. Only this hash (digest) is sent over the wire, so the authentication data is not compromised if intercepted. Not all browsers support this, therefore, it is not very widely used. Digest authentication is implemented as a directory plug-in (for the Java System Directory Server 5.0), thus the plain text user password is only known by the directory and never exported, further improving its security.
- SSL Client Authentication[7] — Requires the client to posses a valid SSL certificate issued by a CA that is configured as trusted in the Java System Web Server. If the certificate is valid, a mapping is established using preconfigured rules[8] to some user in the LDAP directory, and this value is used for subsequent ACL processing. In addition to the configurable certificate mappings, custom programmatic mapping modules can be written to address specialized circumstances. (For full details, see the Certificate-Mapping Programmer's Guide[9].)

SSL authentication can be done in several ways. This is the most complete form. There are less complete forms, including:

- Enable SSL for the server and simply require that the client have a certificate from a trusted CA
- Using ACLs that simply require that a client certificate was issued by any trusted CA or that a certificate be issued from an internal CA

For detailed information on how the Java System Web Server administration server supports administering LDAP user data, refer to the *Managing Users and Groups*[10] section of the *Java System Web Server Administrators Guide*.

The previously mentioned LAS mechanism can also be used to integrate the Java System Web Server with other data sources for user and group data instead of using the default LDAP directory support. This provides customers with the ability to link against existing enterprise or legacy user authentication sources.

6. www.ietf.org/rfc/rfc2617.txt
7. docs.sun.com/docs/manuals/security/sslin/index.htm
8. docs.sun.com/docs/manuals/enterprise/60sp1/ag/esecurty.htm#16636
9. docs.sun.com/docs/manuals/enterprise/50/pg/1-intro.htm#31826
10. docs.sun.com/docs/manuals/enterprise/60sp1/ag/esusrgrp.htm

 © 2003 Sun Microsystems, Inc.

Access Types

In addition to the previously documented cases, access can be granted for only a limited set of operations in combination with any of the other constraints above. The following access operations are supported:

- Read — Allow reading the files
- Write — Allow modifying or moving the files
- Execute — Allow execution of server-side executables
- Delete — Allow removal of files (in conjunction with write permission)
- List — Allow listing directory contents
- Info — Allow accessing meta-information about a given resource

Access Control Through .htaccess

The Java System Web Server also supports the common .htaccess file format for specifying per-directory access control restrictions. ACL processing is done first, if applicable, and then the .htaccess directives, if any, are processed.

Using .htaccess is attractive for distributing the management of access control restrictions among users who control various segments of the document tree but do not have access to the Java System Web Server administration system. For example, an Internet service provider (ISP) that will not grant Java System Web Server configuration access to its customers can still allow each customer to control their Web content through .htaccess files in their Web document directories.

Virtual Server Support

As noted previously, each virtual server can have its own ACL files in order to facilitate distributed management. Additionally, user databases used for authentication may be mapped on a per virtual server basis in order to support different sets of users on each virtual domain. This would be typical in, for example, an ISP installation where each virtual server may belong to a different customer or entity.

Architectural Security

While every security exploit is different and makes use of different entry points, the majority of successful attacks fall into a few common categories:

- Buffer overflow attacks — Perhaps the most common type of attack, this attack takes advantage of faulty memory buffer management to inject some exploit code onto the server process stack and execute it. Code Red is a good example (affecting only Microsoft IIS Web servers).
- Exploit of other specific bugs — This type of attack generally attempts to invoke some otherwise valid request in a carefully constructed fashion in order to trigger some bug in the server code. An example of this is the Nimbda worm (affecting only Microsoft IIS Web servers).
- Denial of service attacks — This attack floods a server with overwhelming traffic in order to slow it down or crash it. Ultimately, it is impossible to clearly separate excessive legitimate traffic from a denial-of-service attack, but some tactics can be used to reduce susceptibility.

Java System Web Server Resistance to Attacks

The primary method used by the Java System Web Server to protect against these types of exploitable bugs is through code reviews. Since the most common attack models are known, it is frequently possible to predict many of the locations where such bugs may exist and find them through code inspections instead of waiting until an exploit shows up in the field.

Of course, not all exploits will be avoided, but such preventive efforts can go a long way. Inevitably, some errors will occur at some point in time. Knowing this, it is also useful to consider whether the server architecture offers any secondary defenses during such events. Along these lines it is interesting to look at the case of Microsoft IIS, which executes in kernel space. The implications of such a design are severe. Any bugs in the Web server will bring down the entire system. Any successful exploits that can execute arbitrary code on the Web server will have unlimited access to all services and resources on the affected host.

The Java System Web Server runs as a regular process in user space and does not need to run with root privileges. This means that even if the Java System Web Server were to be compromised, the underlying system can remain secure (assuming the system has been properly configured). Similarly, crashing the Java System Web Server will only bring down the Java System Web Server, notthe entire server (and the Java System Web Server watchdog will often succeed in quickly restarting the Web server automatically).

When running on the Solaris™ Operating System (OS), even further lines of defense can be established if desired. Running in a chroot environment offers an extra line of defense that can be applied on all supported UNIX® (and UNIX-like) operating systems, including the Solaris OS.

Chapter 3

Security By Default

One of the most common causes of server security incidents is the installation of insecure default configurations done under the assumption that production deployments will always properly reconfigure the system as documented. In practice, default installation values are often kept as is, so any weaknesses become popular attack targets.

Common examples of insecure default installations include:

- Default passwords
- Demo/test content with CGI scripts or other dynamic content with exploitable security holes (for example, the IIS Nimbda worm)
- Wide open default configurations such as null passwords, write permissions to all by default, and so forth

The Java System Web Server installation does not set any default or null passwords that could be used to access the server. The administration server password must be selected by the user during installation. A Java System Web Server installation only includes static text as default pages, no dynamic content (such as CGI scripts) is included. This greatly reduces the likelihood of exploitable bugs in default installations.

Chapter 4

Deployment Security

No server exists as an isolated system, therefore, even though the server security is critical, it is equally important to consider the overall environment in which it is deployed. This is a vast topic that covers all the networks, protocols, operating systems, and hardware surrounding the server installation. This section will only touch on a few aspects directly related to the Java System Web Server deployment.

Running in chroot Environments

Running in a chroot environment provides an extra line of defense. If the server is somehow compromised and the attacker gains access to executing arbitrary code on the host OS, it still cannot access any file system areas outside of the Java System Web Server space, greatly limiting the opportunity for further damage. Refer to the documentation[11] for details on setting up a chroot environment for the Java System Web Server.

Multiplatform Benefits

The Java System Web Server supports multiple platforms. In terms of security, this provides benefits such as being able to do testing and development on common desktop systems, while still deploying onto enterprise-grade Sun servers which are far less likely to fall for common viruses, worms and other attacks.

11.docs.sun.com/docs/manuals/enterprise/60sp1/ag/esecurty.htm#16925

 © 2003 Sun Microsystems, Inc.

Also, several papers have been written on the security resilience of diverse systems. If a server farm contains a variety of system types, an exploit that takes down any one of them will usually have no effect on the others. Therefore, graceful degradation of service is achieved. The multiplatform nature of the Java System Web Server permits this type of installation; it would not be possible with single-platform Web servers.

Chapter 5

Web Application Security

In the past, most Web content was static. Today, a large amount, if not most, of the content served by Web servers is dynamically generated from various data sources as needed. This means that more and more application code is executed directly or indirectly by the Web server as a result of client requests, often with parameters coming from the client itself. Inevitably, this leads to numerous security problems as attackers attempt to find and exploit bugs in the Web application code in order to gain access to the Web server or OS. Witness the numerous attacks against all types of CGI scripts; while not an inherent weakness of the CGI model, such attacks take advantage of common errors typical of the various languages often used for CGI scripts.

While the Java System Web Server certainly supports CGI scripts, it also provides a full-featured Java™ Servlet and JavaServer Pages™ (JSP™) engine. By its nature, the Java language offers significantly improved resistance against memory corruption and other overflow problems typical of C or Perl CGI scripts. The Java System Web Server Web container separates individual Web applications into independent class-loader spaces, providing some further separation between each application.

Chapter 6

Administration Security

The administration server is an independent Java System Web Server instance dedicated to administration. As such, all access control capabilities described previously are available for controlling access to administration functionality.

The administration server always runs on the same host as the instance(s) that it manages. Therefore, all communication between the administration server and the managed instances is always local and not susceptible to network snooping or masquerading.

Remote access to the administration server is done from the client browser. By default, HTTP Basic authentication is always required. For more secure installations, the administration server can be configured to use SSL in order to protect the communication over the wire. All of the user authentication and access control capabilities of the Java System Web Server are available to control and limit administration functionality as well.

Special Administration Access Controls

The functionality provided by the administration server is categorized into program groups. Access to the various program groups can be granted or denied for specific users. The program groups are defined along the lines of the main administration server screen groups: Servers, Preferences, Global Settings, Users & Groups, Security, and Cluster Management.

Furthermore, within each program group, individual pages can be restricted or allowed. These capabilities mean that it is possible to define various administrator users and carefully tailor the exact capabilities granted to each one.

Chapter 7

Tested Security

As mentioned previously, Java System Web Server code is audited and peer-reviewed in order to identify and correct as many potential security problems as possible before the product ships to customers.

Whether discovered internally or in the field, test cases are added for all security bugs into the Java System Web Server QA process in order to help guarantee that there will be no regressions in the future.

Chapter 8

Track Record

A search of the CERT alert database[12] shows only a few security bugs for the Java System Web Server. As a point of comparison, both Microsoft IIS and Apache have more security alerts listed.

The stability and uptime record of the Java System Web Server is also quite impressive, with major sites such as CNN.com experiencing excellent availability during the most demanding times, such as the 2000 election coverage and the September 2001 terrorist attacks. In September 2001 alone, CNN had over 24.5 million unique users go to their site for up to the minute news coverage. In addition, four of the top five news Web sites are hosted on the Java System Web Server.

Of course, security robustness is only one of the many factors contributing to the stability record of the Java System Web Server. However, a quick look at the extensive downtime suffered by hundreds of Microsoft IIS sites due to the numerous exploits against that product should be an indicator as to the overall importance of a strong security package.

12.www.kb.cert.org/vuls/html/search

SUN™ Copyright 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, Java, JavaServer Pages, JSP, and Solaris are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-3(a).

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

SUN™ Copyright 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, Californie 95054 Etats-Unis. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, Java, JavaServer Pages, JSP, et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REPONDRE A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.

Please
Recycle

Adobe PostScript™

Learn More

Get the inside story on the trends and technologies shaping the future of computing by signing up for the Sun Inner Circle program. You'll receive a monthly newsletter packed with information on the latest innovations, plus access to a wealth of resources. Register today to join the Sun Inner Circle Program at sun.com/joinic.

To receive additional information on Sun software, products, programs, and solutions, visit sun.com/software.

Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 USA **Phone** 800 786-7638 or +1 512 434-1577 **Web** sun.com

We make the net work.

Sun Worldwide Sales Offices: Africa (North, West and Central) +33-13-067-4680, Argentina +5411-4317-5600, Australia +61-2-9844-5000, Austria +43-1-60563-0, Belgium +32-2-704-8000, Brazil +55-11-5187-2100, Canada +905-477-6745, Chile +56-2-3724500, Colombia +571-629-2323, Commonwealth of Independent States +7-502-935-8411, Czech Republic +420-2-3300-9311, Denmark +45 4556 5000, Egypt +202-570-9442, Estonia +372-6-308-900, Finland +358-9-525-561, France +33-134-03-00-00, Germany +49-89-46008-0, Greece +30-1-618-8111, Hungary +36-1-489-8900, Iceland +354-563-3010, India–Bangalore +91-80-2298989/2295454; New Delhi +91-11-6106000; Mumbai +91-22-697-8111, Ireland +353-1-8055-666, Israel +972-9-9710500, Italy +39-02-641511, Japan +81-3-5717-5000, Kazakhstan +7-3272-466774, Korea +822-2193-5114, Latvia +371-750-3700, Lithuania +370-729-8468, Luxembourg +352-49 11 33 1, Malaysia +603-21161888, Mexico +52-5-258-6100, The Netherlands +00-31-33-45-15-000, New Zealand–Auckland +64-9-976-6800; Wellington +64-4-462-0780, Norway +47 23 36 96 00, People's Republic of China–Beijing +86-10-6803-5588; Chengdu +86-28-619-9333; Guangzhou +86-20-8755-5900; Shanghai +86-21-6466-1228; Hong Kong +852-2202-6688, Poland +48-22-8747800, Portugal +351-21-4134000, Russia +7-502-935-8411, Singapore +65-6438-1888, Slovak Republic +421-2-4342-94-85, South Africa +27 11 256-6300, Spain +34-91-596-9900, Sweden +46-8-631-10-00, Switzerland–German 41-1-908-90-00; French 41-22-999-0444, Taiwan +886-2-8732-9933, Thailand +662-344-6888, Turkey +90-212-335-22-00, United Arab Emirates +9714-3366333, United Kingdom +44-1-276-20444, United States +1-800-555-9SUN or +1-650-960-1300, Venezuela +58-2-905-3800 4/03 FE1961-0