.NET FRAMEWORK SECURITY CHECKLIST

Version 1, Release 3

22 April 2016

Developed by DISA for the DoD

Trademark Information

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

1. INTRODUCTION

1.1 Overview

The .NET Framework Security Readiness Review (SRR) targets conditions that undermine the integrity of security, contribute to inefficient security operations and administration, or may lead to interruption of production operations. Additionally, the review ensures the site has properly installed and implemented the .NET environment and that it is being managed in a way that is secure, efficient, and effective. The items reviewed are based the NSA guide, *Guide to Microsoft .NET Framework Security*. The results of the review should be recorded in the SRR Results section with the following status designations: F- Finding, N/F- Not A Finding, N/A- Not Applicable, MR -Manual Review, or NR – Not Reviewed.

DISA has assigned a level of urgency to each finding based on Chief Information Officer (CIO) established criteria for certification and accreditation.All findings are based on regulations and guidelines. All findings require correction by the host organization. Category I findings are any vulnerabilities that provide an attacker immediate access into a machine, super user access, or access that bypasses a firewall. Category II findings are any vulnerabilities that provide information that has a high potential of giving access to an intruder. Category III findings are any vulnerabilities that provide information that potentially could lead to compromise.

Note: Security patches required by the DOD IAVM process are reviewed during an operating system security review. Information for security patch compliance is available in Appendix A of this .Net Framework Security Checklist.

1.2 Organization of the Checklist

The .NET Framework Security Checklist is composed of five major sections and three appendices. The organizational breakdown proceeds as follows:

Section 1 Introduction

This section contains summary information about the sections and appendices that comprise the *.NET Framework Security Checklist*, and defines its scope. Supporting documents consulted are listed in this section.

Section 2 .NET SRR Result Report

This section is the matrix that allows the reviewer to document vulnerabilities discovered during the SRR process. This section is used for a .NET Framework Security review.

Section 3 .NET Framework Overview

This section describes the components of the .Net Framework architecture

and explains their relationships.

Section 4 Checklist Instructions

This section gives reviewers more detailed information about conducting the review. Sample configuration information and default configurations information are provided. Instructions specific to conducting reviews on default installations of the .Net Framework.

Section 5 .NET Checklist Procedures

This section documents the procedures that instruct the reviewer on how to perform an SRR using the manual procedures, and how to interpret the resulting information for vulnerabilities. Each procedure maps to a specific vulnerability listed in Section 2.

1.3 Supported Versions

The vulnerabilities discussed in Section 5 of this document are applicable to .NET Framework 1.0, 1.1, 2.0, 3.0 and 3.5.

1.4 Review Method

To perform a successful Security Readiness Review (SRR), a manual process must be employed. There are currently no automated tools to check for compliance with this checklist.

Since each version of the .NET Framework is configured separately a SRR must be performed against each version of the .NET framework that is installed on the system. Refer to Section 4.2 of this document for instructions on determining which versions of the .NET Framework are installed on the system.

1.5 Referenced Documents

The following table enumerates the documents and resources consulted:

Table 1-1: Referenced Documents

Date	Document Description
22 Sep 2004	*Guide to Microsoft .NET Framework Security*, NSA SNAC, v1.4
21 Dec 2006	*Guide to Microsoft .NET Framework 2.0 Security*

1.6 Vulnerability Severity Category Code Definitions

Severity Category Codes (referred to as CAT) are a measure of vulnerabilities used to assess a facility or system security posture. Each security policy specified in this document is assigned a Severity Category Code of CAT I, II, or III.

Table 1-2: Vulnerability Severity Category Code Definitions

	DISA Category Code Guidelines
CAT I	Any vulnerability, the exploitation of which will, **directly and immediately** result in loss of Confidentiality, Availability, or Integrity.
CAT II	Any vulnerability, the exploitation of which **has a potential** to result in loss of Confidentiality, Availability, or Integrity.
CAT III	Any vulnerability, the existence of which **degrades measures** to protect against loss of Confidentiality, Availability, or Integrity.

1.7 Document Revisions

Comments or proposed revisions to this document should be sent via email to the following address: disa.stig_spt@mail.mil. DISA will coordinate all change requests with the relevant DoD organizations before inclusion in this document. Approved changes will be made in accordance with the DISA maintenance release schedule.

1.8 Other Considerations

DISA accepts no liability for the consequences of applying specific configuration settings made on the basis of the SRGs/STIGs. It must be noted that the configurations settings specified should be evaluated in a local, representative test environment before implementation in a production environment, especially within large user populations. The extensive variety of environments makes it impossible to test these configuration settings for all potential software configurations.

For some production environments, failure to test before implementation may lead to a loss of required functionality. Evaluating the risks and benefits to a system's particular circumstances and requirements is the system owner's responsibility. The evaluated risks resulting from not applying specified configuration settings must be approved by the responsible Authorizing Official. Furthermore, DISA implies no warranty that the application of all specified configurations will make a system 100% secure.

Security guidance is provided for the Department of Defense. While other agencies and organizations are free to use it, care must be given to ensure that all applicable security guidance is applied both at the device hardening level as well as the architectural level due to the fact that some of the settings may not be able to be configured in environments outside the DoD architecture.

1.9 Product Approval Disclaimer

The existence of a STIG does not equate to DoD approval for the procurement or use of a product.

STIGs provide configurable operational security guidance for products being used by the DoD. STIGs, along with vendor confidential documentation, also provide a basis for assessing compliance with Cybersecurity controls/control enhancements which supports system Assessment and Authorization (A&A) under the DoD Risk Management Framework (RMF). DoD Authorizing Officials (AOs) may request available vendor confidential documentation for a product that has a STIG for product evaluation and RMF purposes from disa.stig_spt@mail.mil. This documentation is not published for general access to protect vendor's proprietary information.

AOs have the purview to determine product use/approval IAW DoD policy and through RMF risk acceptance. Inputs into acquisition or pre-acquisition product selection include such processes as:

- National Information Assurance Partnership (NIAP) evaluation for National Security Systems (NSS) (http://www.niap-ccevs.org/) IAW CNSSP #11
- National Institute of Standards and Technology (NIST) Cryptographic Module Validation Program (CMVP) (http://csrc.nist.gov/groups/STM/cmvp/) IAW Federal/DoD mandated standards
- DoD Unified Capabilities (UC) Approved Products List (APL) (http://www.disa.mil/network-services/ucco) IAW DoDI 8100.04

2. .NET FRAMEWORK SRR RESULTS REPORT

Unclassified UNTIL FILLED IN
CIRCLE ONE
FOR OFFICIAL USE ONLY (mark each page)
CONFIDENTIAL and SECRET (mark each page and each finding)

Classification is based on classification of system reviewed:
Unclassified System = FOUO Checklist
Confidential System = CONFIDENTIAL Checklist
Secret System = SECRET Checklist
Top Secret System = SECRET Checklist

This checklist becomes effective September 30, 2005

Reviewer: _____ Date:

System: Type of Review (Remote, Sample, Full):

FindingTotal:	**Comments:**
Category I: _____	
Category II: _____	
Category III: _____	
Total:	

2.1 Site Information

Site: _____
System Administrator Information:
Name: _____
E-mail Address: _____
Phone # (Commercial): (___) _____ DSN: _____

ISSO Information:
Name: _____
E-Mail Address _____
Phone # (Commercial) _____ DSN: _____

2.2 System Information

Table 2-1: System Detail

System Detail	
System ID or Host Name	
Hardware Platform	
Operating System	
Operating System Version	
MAC Level	
Confidentiality Level	

Table 2-2: Summary of .Net Framework SRR Findings By Category

Summary of .Net Framework SRR Findings By Category		
Category	**Total Possible Findings**	**Actual Findings**
Category I	1	
Category II	33	
Category III	8	
Total Findings	42	

Table 2-3: .Net Security Framework Findings

(A=Completely Automated, MR = Partially Automated (Manual Review), NC=Can Be Automated, NR = Not Reviewed (Cannot be Automated)).

Procedure Section #	Finding Information		Vulnerability Information		
Section #	**Status**	**Finding Details**	**SDID/ Vulnerability Key**	**Brief Description**	**Category**
5.1	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *File IO* permission is granted with PathDiscovery. The *File IO* permission is granted with *unrestricted="true"*. The *File IO* permission is granted to unrestricted paths.	APPNET0001	File IO Permission	CAT III
5.2	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *Isolated Storage* permission is granted with *unrestricted="true"*. The *Isolated Storage* permission is granted with *allowed=Administrator Isolated Storage By User* and is not monitored The *Isolated Storage* permission is granted with *allowed=Assembly Isolation Storage By Roaming User* and is not monitored.	APPNET0003	Isolated Storage Permission	CAT III
5.3	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *User Interface* permission is granted with *unrestricted="true"*. The *User Interface* permission is granted with *Window=AllWindowsandEvents*. An unauthorized *User Interface* permission with *Windows=SafeTopLevelWindows* is granted.	APPNET0004	User Interface Permission (Windowing)	CAT II
5.4	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *User Interface* permission is granted with *unrestricted="true"*. The *User Interface* permission is granted with access other than *Clipboard=NoClipboard.*	APPNET0005	User Interface Permission (Clipboard)	CAT II

Procedure Section #	Finding Information		Vulnerability Information		
Section #	**Status**	**Finding Details**	**SDID/ Vulnerability Key**	**Brief Description**	**Category**
5.5	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *Reflection* permission is granted with *unrestricted="true"*. The *Reflection* permission is granted with *Flags="Member"*. The *Reflection* permission is granted with *Flags="Type"* to software that is not a confirmed engineering tool or software interoperability service.	APPNET0006	Reflection Permission	CAT III
5.6	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *Printing* permission is granted with *unrestricted="true"*. The *Printing* permission is granted with *Level=AllPrinting.*	APPNET0007	Printing Permission	CAT III
5.7	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *DNS* permission is granted with *unrestricted="true"* to assemblies that do not originate within the local network.	APPNET0008	DNS Permission	CAT II
5.8	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *Socket* permission is granted with *unrestricted="true"*. The *Socket* permission is granted to code that does not provide networking service. The *Socket* permission is granted to code that originates from an external network.	APPNET0009	Socket Access Permission	CAT II
5.9	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *Web* permission is granted with *unrestricted="true"*. The *Web* permission is granted to specific URLs that are not documented and approved for sharing data.	APPNET0010	Web Access Permission	CAT II

Procedure Section #	Finding Information		Vulnerability Information		
Section #	**Status**	**Finding Details**	**SDID/ Vulnerability Key**	**Brief Description**	**Category**
5.10	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *Message Queue* permission is granted with *unrestricted="true"*. The *Message Queue* permission is granted with *access=Administrater* to a queue that is not used by a confirmed administrative tool. The *Message Queue* permission is granted *with access=Browse* to all queues code that is not a verified administrative tool.	APPNET0011	Message Queue Permission	CAT II
5.11	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *Service Controller* permission is granted with *unrestricted="true"*. The *Service Controller* permission is granted to a service whose code is not documented as having the same level of trust and value as the service(s) itself.	APPNET0012	Service Controller Permission	CAT II
5.12	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The database client permission is granted with *unrestricted="true"*. The database permission to specific providers is granted to unauthorized code.	APPNET0013	Database Permission	CAT III
5.13	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *Security* permission is granted with *unrestricted="true"*. The *Security* permission is granted with *Flags="Infrastructure" (Extend infrastructure)* to any code that has not been verified as having complete control over message processing.	APPNET0014	Security Permission (Extend Infrastructure)	CAT II

Procedure Section #	Finding Information		Vulnerability Information		
Section #	**Status**	**Finding Details**	**SDID/ Vulnerability Key**	**Brief Description**	**Category**
5.14	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *Security* permission is granted with *Flags="RemotingConfiguration"* to code that is not highly trusted (has a strong name with public key associated with a local entity) and that does not require network access.	APPNET0015	Security Permission (Enable Remoting Configuration)	CAT II
5.15	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *Security* permission is granted with *Flags="SerializationFormatter"* to code that is not authorized as an extension to the CLR's trusted library base.	APPNET0016	Security Permission (Enable Serialization Formatter)	CAT II
5.16	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *Security* permission is granted with *Flags="ControlThread"* to code that is not Fully Trusted (a member of the FullTrust Permission set).	APPNET0017	Security Permission (Enable Thread Control)	CAT II
5.17	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *Security* permission is granted with *Flags="ControlPrincipal"* (Allow principal control) to code that is not documented as being as trusted as the most privileged system user account).	APPNET0018	Security Permission (Allow Principal Control)	CAT II
5.18	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *Security* permission is granted with *Flags="Execution" (Allow assembly execution)* to unauthorized code.	APPNET0019	Security Permission (Enable Assembly Execution)	CAT II
5.19	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *Security* permission is granted with *Flags="Skipverification" (Skip verification)* to code that is not highly trusted (has a strong name with a public key associated with a trusted party).	APPNET0020	Security Permission (Skip Verification)	CAT II

Procedure Section #	Finding Information		Vulnerability Information		
Section #	**Status**	**Finding Details**	**SDID/ Vulnerability Key**	**Brief Description**	**Category**
5.20	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *Security* permission is granted with *Flags="UnmanagedCode" (Allow calls to unmanaged assemblies).*	APPNET0021	Security Permission (Allow Calls to Unmanaged Assemblies)	CAT II
5.21	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *Security* permission is granted with *Flags="ControlPolicy" (Allow Policy Control)* to code that is not a highly trusted (has a strong name with a public key associated with a local entity) administrative tool.	APPNET0022	Security Permission (Allow Policy Control)	CAT II
5.22	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *Security* permission is granted with *Flags="ControlDomainPolicy" (Allow Domain Policy Control)* to code that is not highly trusted (has a strong name with a public key associated with a local entity) and is not a custom Runtime Host application that implements organizational policy using the AppDomain CAS policy level or to code that does not require the dynamic launching of applications that are less trusted than itself.	APPNET0023	Security Permission (Allow Domain Policy Control)	CAT II
5.23	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *Security* permission is granted with *Flags="ControlEvidence" (Allow Evidence Control)* to code that is not highly trusted (has a strong name with a public key associated with a trusted entity) and has not been developed using secure coding techniques.	APPNET0024	Security Permission (Allow Evidence Control)	CAT II

Procedure Section #	Finding Information		Vulnerability Information		
Section #	**Status**	**Finding Details**	**SDID/ Vulnerability Key**	**Brief Description**	**Category**
5.24	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *Security* permission is granted with *Flags=”Assertion” (Assert any Permission that has been granted)* to code that is not a highly trusted extension to the CLR base libraries.	APPNET0025	Security Permission (Assert any Permission that Has Been Granted)	CAT II
5.25	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *Performance Counter* permission is granted with *unrestricted=”true”* to non-default permission set. The *Performance Counter* permission is granted to an unauthorized machine or category. The *Performance Counter* permission is granted with *Instrument* or *Administrator* access to code that does not provide or administer monitoring.	APPNET0026	Performance Counter Permission	CAT III
5.26	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *Environment Variables* permission is granted with *unrestricted=”true”* to a non-default permission set.	APPNET0027	Environment Variables Permission	CAT II
5.27	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *EventLog* permission is granted with *unrestricted=”true”* to a non-default permission set assigned to code that is not used to monitor system and application events.	APPNET0028	EventLog Permission	CAT II
5.28	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *Registry* permission is granted with *unrestricted=”true”* to a non-default permission set to *Registry* permissions are granted to unauthorized code.	APPNET0029	Registry Permission	CAT II

Procedure Section #	Finding Information		Vulnerability Information		
Section #	**Status**	**Finding Details**	**SDID/ Vulnerability Key**	**Brief Description**	**Category**
5.29	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *Directory Services* permission is granted with *unrestricted=”true” (Grant assemblies unrestricted access to all directory service paths is selected)* to a non-default permission set. The *Directory Services* permission specifies unauthorized service paths. *Write* access to the Windows system directory is granted to code that is not a trusted administrative tool *Browse* access to the Windows system directory services (Active Directory/Global catalog IIS Metabase) is granted to code that is not of local origin (with a strong name with a public key associated with a local entity).	APPNET0030	Directory Services Permission	CAT II
5.30	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	Strong names are simulated on a production system.	APPNET0031	No Strong Name Verification	CAT II
5.31	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	Non-default First Match Code Groups are defined.	APPNET0032	First Match Code Groups	CAT II
5.32	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	File Code Groups have been manually added to a .config file.	APPNET0033	File Code Groups, Net Code Groups	CAT II

Procedure Section #	Finding Information		Vulnerability Information		
Section #	**Status**	**Finding Details**	**SDID/ Vulnerability Key**	**Brief Description**	**Category**
5.33	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	Unauthorized code groups are assigned to *LevelFinal* attribute.	APPNET0035	Level Final Code Group Attribute	CAT III
5.34	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	A non-default code group is assigned the *Zone* Membership Condition.	APPNET0041	Zone Membership Condition	CAT II
5.35	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	CAS security is not enabled.	APPNET0045	Administering CAS Policy	CAT I
5.36	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *Trust Providers Software Publishing State* must be set to *23C00*.	APPNET0046	Software Publishing Certificate	CAT II
5.37	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	An unauthorized code group is assigned the *Publisher* Membership Condition.	APPNET0048	Publisher Membership Condition	CAT II
5.38	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	The *StrongName* condition type listed for a non-default code group does not use a DOD PKI or an authorized third-party certificate.	APPNET0052	Strong Name Membership Condition	CAT II
5.39	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	Non-default code group names are not unique.	APPNET0054	Administering CAS Policy for Group Names	CAT III

Procedure Section #	Finding Information		Vulnerability Information		
Section #	**Status**	**Finding Details**	**SDID/ Vulnerability Key**	**Brief Description**	**Category**
5.40	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	CAS policy and configuration files are not included as part of a reliable backup strategy.	APPNET0055	Administering CAS Policy and Policy Configuration File Backups	CAT II
5.41	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	Applications assigned the *typefilterlevel="Full"* attribute do not require authentication and encryption.	APPNET0060	Remoting Services Authentication and Encryption	CAT II
5.42	□ Finding □ Not a Finding □ Not Applicable □ Not Reviewed	Verify the installed .Net Frameworks are still supported by Microsoft.	APPNET0061	Unsupported .Net Framework Versions	CAT II

3. .NET FRAMEWORK OVERVIEW

Programs written for the .NET Framework execute with the credentials of the user account used to launch the program. As such these programs are constrained by any operating system security settings that may be in place. This is identical to the operation of any non .NET application. The additional restrictions that may be imposed through the .NET Framework are designed to further restrict .NET applications, providing an additional layer of protection.

The following components are used to establish which permissions are granted by the .NET Framework.

3.1 Assemblies

An assembly is the .NET Frameworks' term for a program. An assembly may consist of multiple executables, DLLs, and libraries.

3.2 Evidence

Evidence is information about an assembly. Evidence may be contained in the assembly itself or may be presented by the host. There are currently seven types of evidence in the .NET Framework. These evidence types are:

- Application Directory – the directory where the assembly resides.
- Hash – a cryptographic hash of the assembly.
- Publisher – The publisher of the application, based upon Authenticode signing of the assembly.
- Site – The site where the assembly originated. This is only valid when the assembly is executed directly from the site.
- StrongName – A cryptographic signing of the assembly.
- URL – The URL where the assembly originated. This is only valid when the assembly is executed directly from the URL.
- Zone – The Internet Explorer Security Zone associated with the site of origin for the assembly.

3.3 Permission Sets

Permission Sets are groups of permission that can be granted to .NET Assemblies. There are several default Permission Sets, and non-default Permission Sets may be created.

3.4 Code Group

A code group is used to assign a Permission Set to an Assembly. Assemblies are placed into 1 or more Code Groups based upon the Evidence they present. As part of membership detection any membership conditions for parent code groups must also be met.

When performing an SRR of the .NET Framework it is not enough to simply evaluate the

permissions assigned to a Permission Set to determine whether a vulnerability exists or not. Code Groups which are granted that permission set must be considered as part of the evaluation process to ensure that potentially dangerous permissions are not granted to unapproved assemblies.

For example: Access to the file system is one of the permissions that can be granted through a Permission Set. These permissions range from no access to the file system, to limited access to specific files or directories, to full access to the file system. A Permission Set that grants unrestricted access to the file system is not a vulnerability in and of itself. However if that Permission Set were granted to a Code Group whose membership condition was the Internet Zone , essentially granting full file system access to any program downloaded from the Internet, then this would be a vulnerability. The same Permission Set assigned to a Code Group whose membership is restricted by a Strong Name signed assembly, where the keys used for the signing are controlled by the site, would not be considered a vulnerability.

Note: In the example above, an Assembly that is granted unrestricted access to the file system would still be restricted by the File ACLS of the system.

3.5 Determining Effective Permissions

.NET Framework security policies can be defined at four levels: Enterprise, Machine, User, and Application Domain. Of these four, only the Enterprise, Machine, and User levels will be considered as part of the evaluation process. The configuration information for each level is stored in configuration files within each .NET Framework directory. There is currently no central management capability, so these files must be copied to every system in order for them to be effective.

Determining the effective permissions for a given assembly involves determining which code groups the assemblies belong to and combining all of the permission sets granted to the assembly to arrive at an effective permission set. Please refer to the *NSA Guide to Microsoft .NET Framework Security*, pages 66 – 71 for a detailed description of the rules and procedures used to determine the effective permission set.

4. CHECKLIST INSTRUCTIONS

This section details the procedures needed to perform a Security Readiness Review (SRR) of a .NET Framework installation. The .NET SRR is a manual process that uses the following tools: Microsoft .NET Framework Configuration Tools Code Access Security (CAS) Policy Tool caspol.exe, regedit.exe. These tools reside in the installations directory of .Net Framework.

4.1 .NET Framework Vulnerability Types

The checklist is categorized with eleven different types of the vulnerabilities.

Table 4-1: .Net Framework Vulnerability Types

Permission Vulnerabilities	APPNET0001-APPNET0030
Strong Name Verification	APPNET031
Special Code Group Type	APPNET0032, APPNET0033, APPNET0035
Membership Conditions	APPNET0041, APPNET0048, APPNET0052
CAS Policy	APPNET045
Software Publishing Certificate	APPNET0046
Duplicate Code Group Name	APPNET0054
Backup Vulnerabilities	APPNET0055
Remoting Services and Authentication	APPNET0060
Unsupported .Net Framework Versions	APPNET0061

4.2 Versions of .NET Framework Installed

To determine which versions of .Net Framework are installed, search for all the Mscorlib.dll files in %systemroot%\Microsoft.NET\Framework. Click on each of the files and view properties and click Version tab to determine the version installed. If there is no Mscorlib.dll, there is no installed version of .Net Framework in that directory.

More specific information on determining versions of .Net Framework installed can be found at the following link. http://support.microsoft.com/kb/318785

4.3 Default Installations of .NET

Most systems are not setup to perform .Net development. As a result most systems are installed with a default configuration of .Net Frameworks. A default installation of .Net is default configuration provided either when the operating system was first loaded or when the .Net Framework was loaded.

Note: Window 2003 has a default installation of .Net Framework 1.1 which cannot be removed.

- **Default Code Group:** Code group that are installed by the default installations of .Net 1.0, 1.1, 2.0, 3.0, and 3.5.
- **Non-Default Code Group:** Code group that is not part of the default installation of .Net.
- **Default Permission Set:** Permission set that is part of the default installation of .Net.
- **Non-Default Permission Set**: Permission set that is not part of the default installation of .Net.

If the system has a default installation of .Net then verify that default code groups and default permission sets against the list of default code groups and permission set in the following sections.

After a verifying the only default code groups and permissions sets exist, use the table below to mark the findings in for a default installation.

Table 4-2: Default Installation Findings

Vulnerability ID	Finding Status	Brief Description	CAT
APPNET0001	Not a Finding	File IO Permission	CAT III
APPNET0003	Not a Finding	Isolated Storage Permission	CAT III
APPNET0004	Not a Finding	User Interface Permission (Windowing)	CAT II
APPNET0005	Not a Finding	User Interface Permission (Clipboard)	CAT II
APPNET0006	Not a Finding	Reflection Permission	CAT III
APPNET0007	Not a Finding	Printing Permission	CAT III
APPNET0008	Not a Finding	DNS Permission	CAT II
APPNET0009	Not a Finding	Socket Access Permission	CAT II
APPNET0010	Not a Finding	Web Access Permission	CAT II
APPNET0011	Not a Finding	Message Queue Permission	CAT II
APPNET0012	Not a Finding	Service Controller Permission	CAT II
APPNET0013	Not a Finding	Database Permission	CAT III
APPNET0014	Not a Finding	Security Permission (Extend Infrastructure)	CAT II
APPNET0015	Not a Finding	Security Permission (Enable Remoting Configuration)	CAT II
APPNET0016	Not a Finding	Security Permission (Enable Serialization Formatter)	CAT II
APPNET0017	Not a Finding	Security Permission (Enable Thread Control)	CAT II
APPNET0018	Not a Finding	Security Permission (Allow Principal Control)	CAT II
APPNET0019	Not a Finding	Security Permission (Enable Assembly Execution)	CAT II
APPNET0020	Not a Finding	Security Permission (Skip Verification)	CAT II
APPNET0021	Not a Finding	Security Permission (Allow Calls to Unmanaged Assemblies)	CAT II
APPNET0022	Not a Finding	Security Permission (Allow Policy Control)	CAT II
APPNET0023	Not a Finding	Security Permission (Allow Domain Policy Control)	CAT II

Vulnerability ID	Finding Status	Brief Description	CAT
APPNET0024	Not a Finding	Security Permission (Allow Evidence Control)	CAT II
APPNET0025	Not a Finding	Security Permission (Assert any Permission that Has Been Granted)	CAT II
APPNET0026	Not a Finding	Performance Counter Permission	CAT III
APPNET0027	Not a Finding	Environment Variables Permission	CAT II
APPNET0028	Not a Finding	Event Log Permission	CAT II
APPNET0029	Not a Finding	Registry Permission	CAT II
APPNET0030	Not a Finding	Directory Services Permission	CAT II
APPNET0031		No Strong Name Verification	CAT II
APPNET0032	Not a Finding	First Match Code Groups	CAT II
APPNET0033		File Code Groups, Net Code Groups	CAT II
APPNET0035	Not a Finding	Level Final Code Group Attribute	CAT III
APPNET0041	Not a Finding	Zone Membership Condition	CAT II
APPNET0045		Administering CAS Policy	CAT I
APPNET0046		Software Publishing Certificate	CAT II
APPNET0048	Not a Finding	Publisher Membership Condition	CAT II
APPNET0052	Not a Finding	Strong Name Membership Condition	CAT II
APPNET0054	Not a Finding	Administering CAS Policy for Group Names	CAT III
APPNET0055		Administering CAS Policy and Policy Configuration File Backups	CAT II
APPNET0060		Remoting Services Authentication and Encryption	CAT II
APPNET0061		Unsupported .Net Framework Versions	CAT II

Note: APPNET0031, APPNET0033, APPNET0045, APPNET0046, APPNET0055, APPNET0060 and APPNET0061 must still be reviewed.

4.3.1 .NET Framework 1.0 Default Code Groups

C:\WINDOWS\Microsoft.NET\Framework\v1.0.3705\CasPol.exe -all -lg

Microsoft (R) .NET Framework CasPol 1.0.3705.6018
Copyright (C) Microsoft Corporation 1998-2001. All rights reserved.

Security is OFF
Execution checking is ON
Policy change prompt is ON

Level = Enterprise

Code Groups:

1. All code: FullTrust

Level = Machine

Code Groups:

1. All code: Nothing
  1.1. Zone - MyComputer: FullTrust
    1.1.1. StrongName -
00240000048000009400000006020000002400005253413100040000010001000 7D1FA57C4A
ED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DC
C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E82
1C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0
FC4963D261C8A12436518206DC093344D5AD293: FullTrust
    1.1.2. StrongName - 00000000000000000400000000000000: FullTrust
  1.2. Zone - Intranet: LocalIntranet
    1.2.1. All code: Same site Web.
    1.2.2. All code: Same directory FileIO - Read, PathDiscovery
  1.3. Zone - Internet: Nothing
  1.4. Zone - Untrusted: Nothing
  1.5. Zone - Trusted: Internet
    1.5.1. All code: Same site Web.
Level = User
Code Groups:

1. All code: FullTrust
Success.

4.3.2 NET Framework 1.1 Default Code Groups

C:\WINDOWS\Microsoft.NET\Framework\v1.1.4322\CasPol.exe -all –lg

Microsoft (R) .NET Framework CasPol 1.1.4322.573
Copyright (C) Microsoft Corporation 1998-2002. All rights reserved.

Security is OFF
Execution checking is ON
Policy change prompt is ON

Level = Enterprise

Code Groups:

1. All code: FullTrust

Level = Machine

Code Groups:

1. All code: Nothing
  1.1. Zone - MyComputer: FullTrust
    1.1.1. StrongName -
00240000048000009400000006020000002400005253413100040000010001000 7D1FA57C4A
ED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DC
C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E82
1C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0
FC4963D261C8A12436518206DC093344D5AD293: FullTrust
    1.1.2. StrongName - 00000000000000000400000000000000: FullTrust
  1.2. Zone - Intranet: LocalIntranet
    1.2.1. All code: Same site Web.
    1.2.2. All code: Same directory FileIO - Read, PathDiscovery
  1.3. Zone - Internet: Internet
    1.3.1. All code: Same site Web.
  1.4. Zone - Untrusted: Nothing
  1.5. Zone - Trusted: Internet
    1.5.1. All code: Same site Web.

Level = User

Code Groups:

1. All code: FullTrust
Success

4.3.3 .NET framework 2.0, 3.0 & 3.5 Default Code Groups

C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\CasPol.exe -all -lg

Microsoft (R) .NET Framework CasPol 2.0.50727.3053
Copyright (c) Microsoft Corporation. All rights reserved.

Security is ON
Execution checking is ON
Policy change prompt is ON

Level = Enterprise

Code Groups:

1. All code: FullTrust

Level = Machine

Code Groups:

1. All code: Nothing
  1.1. Zone - MyComputer: FullTrust
    1.1.1. StrongName -
00240000048000009400000006020000002400005253413100040000010001000 7D1FA57C4A
ED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DC
C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E82
1C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0
FC4963D261C8A12436518206DC093344D5AD293: FullTrust
    1.1.2. StrongName - 0000000000000000040000000000000: FullTrust
  1.2. Zone - Intranet: LocalIntranet
    1.2.1. All code: Same site Web
    1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
  1.3. Zone - Internet: Internet
    1.3.1. All code: Same site Web
  1.4. Zone - Untrusted: Nothing
  1.5. Zone - Trusted: Internet
    1.5.1. All code: Same site Web
  1.6. ApplicationDirectory: Nothing

Level = User

Code Groups:

1. All code: FullTrust
Success

4.3.4 .NET Framework 1.0 Default Permission Sets

C:\WINDOWS\Microsoft.NET\Framework\v1.0.3705\CasPol.exe -all –lp

Microsoft (R) .NET Framework CasPol 1.0.3705.6018
Copyright (C) Microsoft Corporation 1998-2001. All rights reserved.

Security is OFF
Execution checking is ON
Policy change prompt is ON

Level = Enterprise

Named Permission Sets:

1. FullTrust (Allows full access to all resources) =
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"

```
        Unrestricted="true"
        Name="FullTrust"
        Description="Allows full access to all resources"/>

2. Skip Verification (Grants right to bypass the verification) =
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="SkipVerification"
        Description="Grants right to bypass the verification">
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="SkipVerification"/>
</PermissionSet>

3. Execution (Permits execution) =
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="Execution"
        Description="Permits execution">
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Execution"/>
</PermissionSet>

4. Nothing (Denies all resources, including the right to execute) =
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="Nothing"
        Description="Denies all resources, including the right to execute"/>

5. LocalIntranet (Default rights given to applications on the local intranet) =
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="LocalIntranet"
        Description="Default rights given to applications on the local intranet">
<IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Read="USERNAME"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
```

```
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Allowed="AssemblyIsolationByUser"
        UserQuota="9223372036854775807"
        Expiry="9223372036854775807"
        Permanent="True"/>
<IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="ReflectionEmit"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Assertion, Execution"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Net.DnsPermission, System, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Level="DefaultPrinting"/>
<IPermission class="System.Diagnostics.EventLogPermission, System, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1">
<Machine name="."
        access="Instrument"/>
</IPermission>
</PermissionSet>
```

6. Internet (Default rights given to internet applications) =

```
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="Internet"
        Description="Default rights given to internet applications">
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Access="Open"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
```

```
        Allowed="DomainIsolationByUser"
        UserQuota="10240"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Execution"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Window="SafeTopLevelWindows"
        Clipboard="OwnClipboard"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Level="SafePrinting"/>
</PermissionSet>
```

7. Everything (Allows unrestricted access to all resources covered by built-in permissions) =

```
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="Everything"
        Description="Allows unrestricted access to all resources covered by built-in
        permissions">
<IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.FileIOPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.RegistryPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
```

```
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Assertion, UnmanagedCode, Execution, ControlThread, ControlEvidence,
        ControlPolicy, SerializationFormatter, ControlDomainPolicy, ControlPrincipal,
        ControlAppDomain, RemotingConfiguration, Infrastructure"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Net.DnsPermission, System, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Diagnostics.EventLogPermission, System, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Net.SocketPermission, System, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Net.WebPermission, System, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Diagnostics.PerformanceCounterPermission, System,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.DirectoryServices.DirectoryServicesPermission,
System.DirectoryServices, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Messaging.MessageQueuePermission, System.Messaging,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.ServiceProcess.ServiceControllerPermission,
System.ServiceProcess, Version=1.0.3300.0, Culture=neutral,
```

```
PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Data.OleDb.OleDbPermission, System.Data, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        AllowBlankPassword="False"
        Unrestricted="true"/>
<IPermission class="System.Data.SqlClient.SqlClientPermission, System.Data,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        AllowBlankPassword="False"
        Unrestricted="true"/>
</PermissionSet>
```

Level = Machine

Named Permission Sets:

1. FullTrust (Allows full access to all resources) =

```
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Unrestricted="true"
        Name="FullTrust"
        Description="Allows full access to all resources"/>
```

2. SkipVerification (Grants right to bypass the verification) =

```
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="SkipVerification"
        Description="Grants right to bypass the verification">
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="SkipVerification"/>
</PermissionSet>
```

3. Execution (Permits execution) =

```
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="Execution"
        Description="Permits execution">
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Execution"/>
```

</PermissionSet>

4. Nothing (Denies all resources, including the right to execute) =
<PermissionSet class="System.Security.NamedPermissionSet"
version="1"
Name="Nothing"
Description="Denies all resources, including the right to execute"/>

5. LocalIntranet (Default rights given to applications on the local intranet) =
<PermissionSet class="System.Security.NamedPermissionSet"
version="1"
Name="LocalIntranet"
Description="Default rights given to applications on the local intranet">
<IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Read="USERNAME"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Allowed="AssemblyIsolationByUser"
UserQuota="9223372036854775807"
Expiry="9223372036854775807"
Permanent="True"/>
<IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Flags="ReflectionEmit"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Flags="Assertion, Execution"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1" Unrestricted="true"/>
<IPermission class="System.Net.DnsPermission, System, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
version="1"

```
        Level="DefaultPrinting"/>
<IPermission class="System.Diagnostics.EventLogPermission, System, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1">
<Machine name="."
        access="Instrument"/>
</IPermission>
</PermissionSet>
```

6. Internet (Default rights given to internet applications) =

```
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="Internet"
        Description="Default rights given to internet applications">
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Access="Open"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Allowed="DomainIsolationByUser"
        UserQuota="10240"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Execution"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Window="SafeTopLevelWindows"
        Clipboard="OwnClipboard"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Level="SafePrinting"/>
</PermissionSet>
```

7. Everything (Allows unrestricted access to all resources covered by built-in permissions) =

```
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="Everything"
        Description="Allows unrestricted access to all resources covered by built-in
        permissions">
<IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
```

```
            version="1"
            Unrestricted="true"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
            version="1"
            Unrestricted="true"/>
<IPermission class="System.Security.Permissions.FileIOPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
            version="1"
            Unrestricted="true"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
            version="1"
            Unrestricted="true"/>
<IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
            version="1"
            Unrestricted="true"/>
<IPermission class="System.Security.Permissions.RegistryPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
            version="1"
            Unrestricted="true"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
            version="1"
            Flags="Assertion, UnmanagedCode, Execution, ControlThread, ControlEvidence,
            ControlPolicy, SerializationFormatter, ControlDomainPolicy, ControlPrincipal,
            ControlAppDomain, RemotingConfiguration, Infrastructure"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
            version="1"
            Unrestricted="true"/>
<IPermission class="System.Net.DnsPermission, System, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
            version="1"
            Unrestricted="true"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
            version="1"
            Unrestricted="true"/>
<IPermission class="System.Diagnostics.EventLogPermission, System, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
            version="1"
            Unrestricted="true"/>
<IPermission class="System.Net.SocketPermission, System, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
```

```
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Net.WebPermission, System, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Diagnostics.PerformanceCounterPermission, System,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.DirectoryServices.DirectoryServicesPermission,
System.DirectoryServices, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Messaging.MessageQueuePermission, System.Messaging,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.ServiceProcess.ServiceControllerPermission,
System.ServiceProcess, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Data.OleDb.OleDbPermission, System.Data, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        AllowBlankPassword="False"
        Unrestricted="true"/>
<IPermission class="System.Data.SqlClient.SqlClientPermission, System.Data,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        AllowBlankPassword="False"
        Unrestricted="true"/>
</PermissionSet>
```

Level = User

Named Permission Sets:

1. FullTrust (Allows full access to all resources) =

```
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Unrestricted="true" Name="FullTrust"
        Description="Allows full access to all resources"/>
```

2. SkipVerification (Grants right to bypass the verification) =
<PermissionSet class="System.Security.NamedPermissionSet"
version="1"
Name="SkipVerification"
Description="Grants right to bypass the verification">
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Flags="SkipVerification"/>
</PermissionSet>

3. Execution (Permits execution) =
<PermissionSet class="System.Security.NamedPermissionSet" version="1"
Name="Execution" Description="Permits execution">
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1" Flags="Execution"/>
</PermissionSet>

4. Nothing (Denies all resources, including the right to execute) =
<PermissionSet class="System.Security.NamedPermissionSet"
version="1"
Name="Nothing"
Description="Denies all resources, including the right to execute"/>

5. LocalIntranet (Default rights given to applications on the local intranet) =
<PermissionSet class="System.Security.NamedPermissionSet"
version="1"
Name="LocalIntranet"
Description="Default rights given to applications on the local intranet">
<IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Read="USERNAME"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Allowed="AssemblyIsolationByUser"
UserQuota="9223372036854775807"
Expiry="9223372036854775807" Permanent="True"/>
<IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"

```
        version="1"
        Flags="ReflectionEmit"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Assertion, Execution"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Net.DnsPermission, System, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Level="DefaultPrinting"/>
<IPermission class="System.Diagnostics.EventLogPermission, System, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1">
<Machine name="."
        access="Instrument"/>
</IPermission>
</PermissionSet>
```

6. Internet (Default rights given to internet applications) =

```
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="Internet"
        Description="Default rights given to internet applications">
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Access="Open"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Allowed="DomainIsolationByUser"
        UserQuota="10240"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Execution"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
```

```
        version="1"
        Window="SafeTopLevelWindows"
        Clipboard="OwnClipboard"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Level="SafePrinting"/>
</PermissionSet>
```

7. Everything (Allows unrestricted access to all resources covered by built-in permissions) =

```
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="Everything"
        Description="Allows unrestricted access to all resources covered by built-in
        permissions">
<IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.FileIOPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.RegistryPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Assertion, UnmanagedCode, Execution, ControlThread, ControlEvidence,
        ControlPolicy, SerializationFormatter, ControlDomainPolicy, ControlPrincipal,
        ControlAppDomain, RemotingConfiguration, Infrastructure"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=1.0.3300.0,
```

```
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Net.DnsPermission, System, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Diagnostics.EventLogPermission, System, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Net.SocketPermission, System, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Net.WebPermission, System, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Diagnostics.PerformanceCounterPermission, System,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.DirectoryServices.DirectoryServicesPermission,
System.DirectoryServices, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Messaging.MessageQueuePermission, System.Messaging,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.ServiceProcess.ServiceControllerPermission,
System.ServiceProcess, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Data.OleDb.OleDbPermission, System.Data, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        AllowBlankPassword="False"
        Unrestricted="true"/>
```

<IPermission class="System.Data.SqlClient.SqlClientPermission, System.Data,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
AllowBlankPassword="False"
Unrestricted="true"/>
</PermissionSet>

Success

4.3.5 .NET Framework 1.1 Default Permission Sets

C:\WINDOWS\Microsoft.NET\Framework\v1.1.4322\CasPol.exe -all –lp
Microsoft (R) .NET Framework CasPol 1.0.3705.6018
Copyright (C) Microsoft Corporation 1998-2001. All rights reserved.

Security is OFF
Execution checking is ON
Policy change prompt is ON

Level = Enterprise

Named Permission Sets:

1. FullTrust (Allows full access to all resources) =
<PermissionSet class="System.Security.NamedPermissionSet"
version="1"
Unrestricted="true" Name="FullTrust"
Description="Allows full access to all resources"/>

2. SkipVerification (Grants right to bypass the verification) =
<PermissionSet class="System.Security.NamedPermissionSet"
version="1"
Name="SkipVerification"
Description="Grants right to bypass the verification">
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Flags="SkipVerification"/>
</PermissionSet>

3. Execution (Permits execution) =
<PermissionSet class="System.Security.NamedPermissionSet"
version="1"
Name="Execution"
Description="Permits execution">

<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Flags="Execution"/>
</PermissionSet>

4. Nothing (Denies all resources, including the right to execute) =
<PermissionSet class="System.Security.NamedPermissionSet"
version="1"
Name="Nothing"
Description="Denies all resources, including the right to execute"/>

5. LocalIntranet (Default rights given to applications on the local intranet) =
<PermissionSet class="System.Security.NamedPermissionSet"
version="1"
Name="LocalIntranet"
Description="Default rights given to applications on the local intranet">
<IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Read="USERNAME"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Allowed="AssemblyIsolationByUser"
UserQuota="9223372036854775807"
Expiry="9223372036854775807"
Permanent="True"/>
<IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Flags="ReflectionEmit"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Flags="Assertion, Execution"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Net.DnsPermission, System, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"

```
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Level="DefaultPrinting"/>
<IPermission class="System.Diagnostics.EventLogPermission, System, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1">
<Machine name="."
        access="Instrument"/>
</IPermission>
</PermissionSet>
```

6. Internet (Default rights given to internet applications) =

```
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="Internet"
        Description="Default rights given to internet applications">
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Access="Open"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Allowed="DomainIsolationByUser"
        UserQuota="10240"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Execution"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Window="SafeTopLevelWindows"
        Clipboard="OwnClipboard"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Level="SafePrinting"/>
</PermissionSet>
```

7. Everything (Allows unrestricted access to all resources covered by built-in permissions) =

```
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
```

```
        Name="Everything"
        Description="Allows unrestricted access to all resources covered by built-in
        permissions">
<IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.FileIOPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.RegistryPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Assertion, UnmanagedCode, Execution, ControlThread, ControlEvidence,
        ControlPolicy, SerializationFormatter, ControlDomainPolicy, ControlPrincipal,
        ControlAppDomain, RemotingConfiguration, Infrastructure"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Net.DnsPermission, System, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Diagnostics.EventLogPermission, System, Version=1.0.3300.0,
```

```
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Net.SocketPermission, System, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Net.WebPermission, System, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Diagnostics.PerformanceCounterPermission, System,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.DirectoryServices.DirectoryServicesPermission,
System.DirectoryServices, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Messaging.MessageQueuePermission, System.Messaging,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.ServiceProcess.ServiceControllerPermission,
System.ServiceProcess, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Data.OleDb.OleDbPermission, System.Data, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        AllowBlankPassword="False"
        Unrestricted="true"/>
<IPermission class="System.Data.SqlClient.SqlClientPermission, System.Data,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        AllowBlankPassword="False"
        Unrestricted="true"/>
</PermissionSet>
```

Level = Machine

Named Permission Sets:

1. FullTrust (Allows full access to all resources) =

```
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Unrestricted="true" Name="FullTrust"
        Description="Allows full access to all resources"/>
```

2. SkipVerification (Grants right to bypass the verification) =

```
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="SkipVerification"
        Description="Grants right to bypass the verification">
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="SkipVerification"/>
</PermissionSet>
```

3. Execution (Permits execution) =

```
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="Execution"
        Description="Permits execution">
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Execution"/>
</PermissionSet>
```

4. Nothing (Denies all resources, including the right to execute) =

```
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="Nothing"
        Description="Denies all resources, including the right to execute"/>
```

5. LocalIntranet (Default rights given to applications on the local intranet) =

```
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="LocalIntranet"
        Description="Default rights given to applications on the local intranet">
<IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Read="USERNAME"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
```

<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Allowed="AssemblyIsolationByUser"
UserQuota="9223372036854775807"
Expiry="9223372036854775807"
Permanent="True"/>
<IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Flags="ReflectionEmit"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Flags="Assertion, Execution"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Net.DnsPermission, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
version="1"
Level="DefaultPrinting"/>
<IPermission class="System.Diagnostics.EventLogPermission, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1">
<Machine name="."
access="Instrument"/>
</IPermission>
</PermissionSet>

6. Internet (Default rights given to internet applications) =
<PermissionSet class="System.Security.NamedPermissionSet"
version="1"
Name="Internet"
Description="Default rights given to internet applications">
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Access="Open"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"

```
        version="1"
        Allowed="DomainIsolationByUser"
        UserQuota="10240"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Execution"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Window="SafeTopLevelWindows"
        Clipboard="OwnClipboard"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Level="SafePrinting"/>
</PermissionSet>
```

7. Everything (Allows unrestricted access to all resources covered by built-in permissions) =

```
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="Everything"
        Description="Allows unrestricted access to all resources covered by built-in
        permissions">
<IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.FileIOPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.RegistryPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
```

```
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Assertion, UnmanagedCode, Execution, ControlThread, ControlEvidence,
        ControlPolicy, SerializationFormatter, ControlDomainPolicy, ControlPrincipal,
        ControlAppDomain, RemotingConfiguration, Infrastructure"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Net.DnsPermission, System, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Diagnostics.EventLogPermission, System, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Net.SocketPermission, System, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Net.WebPermission, System, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Diagnostics.PerformanceCounterPermission, System,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.DirectoryServices.DirectoryServicesPermission,
System.DirectoryServices, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Messaging.MessageQueuePermission, System.Messaging,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.ServiceProcess.ServiceControllerPermission,
```

```
System.ServiceProcess, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Data.OleDb.OleDbPermission, System.Data, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        AllowBlankPassword="False"
        Unrestricted="true"/>
<IPermission class="System.Data.SqlClient.SqlClientPermission, System.Data,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        AllowBlankPassword="False"
        Unrestricted="true"/>
</PermissionSet>
```

Level = User

Named Permission Sets:

```
1. FullTrust (Allows full access to all resources) =
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Unrestricted="true" Name="FullTrust"
        Description="Allows full access to all resources"/>

2. SkipVerification (Grants right to bypass the verification) =
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="SkipVerification"
        Description="Grants right to bypass the verification">
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="SkipVerification"/>
</PermissionSet>

3. Execution (Permits execution) =
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="Execution"
        Description="Permits execution">
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Execution"/>
```

</PermissionSet>

4. Nothing (Denies all resources, including the right to execute) =
<PermissionSet class="System.Security.NamedPermissionSet"
version="1"
Name="Nothing"
Description="Denies all resources, including the right to execute"/>

5. LocalIntranet (Default rights given to applications on the local intranet) =
<PermissionSet class="System.Security.NamedPermissionSet"
version="1"
Name="LocalIntranet"
Description="Default rights given to applications on the local intranet">
<IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Read="USERNAME"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Allowed="AssemblyIsolationByUser"
UserQuota="9223372036854775807"
Expiry="9223372036854775807"
Permanent="True"/>
<IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Flags="ReflectionEmit"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Flags="Assertion, Execution"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Net.DnsPermission, System, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"

```
        version="1"
        Level="DefaultPrinting"/>
<IPermission class="System.Diagnostics.EventLogPermission, System, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1">
<Machine name="."
        access="Instrument"/>
</IPermission>
</PermissionSet>
```

6. Internet (Default rights given to internet applications) =

```
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="Internet"
        Description="Default rights given to internet applications">
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Access="Open"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Allowed="DomainIsolationByUser"
UserQuota="10240"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Execution"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Window="SafeTopLevelWindows"
        Clipboard="OwnClipboard"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Level="SafePrinting"/>
</PermissionSet>
```

7. Everything (Allows unrestricted access to all resources covered by built-in permissions) =

```
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="Everything"
        Description="Allows unrestricted access to all resources covered by built-in
        permissions">
<IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
```

```
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.FileIOPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.RegistryPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Assertion, UnmanagedCode, Execution, ControlThread, ControlEvidence,
        ControlPolicy, SerializationFormatter, ControlDomainPolicy, ControlPrincipal,
        ControlAppDomain, RemotingConfiguration, Infrastructure"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Net.DnsPermission, System, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Diagnostics.EventLogPermission, System, Version=1.0.3300.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Net.SocketPermission, System, Version=1.0.3300.0,
```

Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Net.WebPermission, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Diagnostics.PerformanceCounterPermission, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.DirectoryServices.DirectoryServicesPermission, System.DirectoryServices, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
version="1"
Unrestricted="true"/>
<IPermission class="System.Messaging.MessageQueuePermission, System.Messaging, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
version="1"
Unrestricted="true"/>
<IPermission class="System.ServiceProcess.ServiceControllerPermission, System.ServiceProcess, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
version="1"
Unrestricted="true"/>
<IPermission class="System.Data.OleDb.OleDbPermission, System.Data, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
AllowBlankPassword="False"
Unrestricted="true"/>
<IPermission class="System.Data.SqlClient.SqlClientPermission, System.Data, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
AllowBlankPassword="False"
Unrestricted="true"/>
</PermissionSet>

Success

4.3.6 .NET Framework 2.0, 3.0 & 3.5 Default Permission Sets

C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\CasPol.exe -all –lp

Microsoft (R) .NET Framework CasPol 2.0.50727.3053

Security is ON
Execution checking is ON
Policy change prompt is ON

Level = Enterprise

Named Permission Sets:

1. FullTrust (Allows full access to all resources) =
<PermissionSet class="System.Security.NamedPermissionSet"
version="1"
Unrestricted="true"
Name="FullTrust"
Description="Allows full access to all resources"/>

2. SkipVerification (Grants right to bypass the verification) =
<PermissionSet class="System.Security.NamedPermissionSet"
version="1"
Name="SkipVerification"
Description="Grants right to bypass the verification">
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Flags="SkipVerification"/>
</PermissionSet>

3. Execution (Permits execution) =
<PermissionSet class="System.Security.NamedPermissionSet"
version="1"
Name="Execution"
Description="Permits execution">
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Flags="Execution"/>
</PermissionSet>

4. Nothing (Denies all resources, including the right to execute) =
<PermissionSet class="System.Security.NamedPermissionSet"
version="1"
Name="Nothing"
Description="Denies all resources, including the right to execute"/>

5. LocalIntranet (Default rights given to applications on the local intranet) =
<PermissionSet class="System.Security.NamedPermissionSet"
version="1"

```
        Name="LocalIntranet"
        Description="Default rights given to applications on the local intranet">
<IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Read="USERNAME"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true" />
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Allowed="AssemblyIsolationByUser"
        UserQuota="9223372036854775807"
        Expiry="9223372036854775807"
        Permanent="True"/>
<IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="ReflectionEmit"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Assertion,
        Execution, BindingRedirects"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Net.DnsPermission, System, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Level="DefaultPrinting"/>
</PermissionSet>
```

6. Internet (Default rights given to Internet applications) =

```
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="Internet"
        Description="Default rights given to Internet applications">
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
```

```
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Access="Open" />
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Allowed="ApplicationIsolationByUser"
        UserQuota="512000"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Execution"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Window="SafeTopLevelWindows"
        Clipboard="OwnClipboard"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Level="SafePrinting"/>
</PermissionSet>
```

7. Everything (Allows unrestricted access to all resources covered by built-in permissions) =

```
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="Everything"
        Description="Allows unrestricted access to all resources covered by built-in
        permissions">
<IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.FileIOPermission, mscorlib, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib,
```

```
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.RegistryPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Assertion, UnmanagedCode, Execution, ControlThread, ControlEvidence,
        ControlPolicy, SerializationFormatter, ControlDomainPolicy, ControlPrincipal,
        ControlAppDomain, RemotingConfiguration, Infrastructure"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.KeyContainerPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Net.DnsPermission, System, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Net.SocketPermission, System, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
        version="1"Unrestricted="true"/>
<IPermission class="System.Net.WebPermission, System, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Diagnostics.EventLogPermission, System, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.StorePermission, System, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Diagnostics.PerformanceCounterPermission, System,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
```

version="1"
Unrestricted="true"/>
<IPermission class="System.Data.OleDb.OleDbPermission, System.Data, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Data.SqlClient.SqlClientPermission, System.Data, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Security.Permissions.DataProtectionPermission, System.Security, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
version="1"
Unrestricted="true"/>
</PermissionSet>

Level = Machine

Named Permission Sets:

1. FullTrust (Allows full access to all resources) =
<PermissionSet class="System.Security.NamedPermissionSet"
version="1"
Unrestricted="true"
Name="FullTrust"
Description="Allows full access to all resources"/>

2. SkipVerification (Grants right to bypass the verification) =
<PermissionSet class="System.Security.NamedPermissionSet"
version="1"
Name="SkipVerification"
Description="Grants right to bypass the verification">
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Flags="SkipVerification"/>
</PermissionSet>

3. Execution (Permits execution) =
<PermissionSet class="System.Security.NamedPermissionSet"
version="1"
Name="Execution"
Description="Permits execution">
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"

```
        Flags="Execution"/>
</PermissionSet>

4. Nothing (Denies all resources, including the right to execute) =
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="Nothing"
        Description="Denies all resources, including the right to execute"/>

5. LocalIntranet (Default rights given to applications on the local intranet) =
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="LocalIntranet"
        Description="Default rights given to applications on the local intranet">
<IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Read="USERNAME"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Allowed="AssemblyIsolationByUser"
        UserQuota="9223372036854775807"
        Expiry="9223372036854775807"
        Permanent="True"/>
<IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="ReflectionEmit"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Assertion, Execution, BindingRedirects"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Net.DnsPermission, System, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
```

```
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Level="DefaultPrinting"/>
</PermissionSet>
```

6. Internet (Default rights given to Internet applications) =

```
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="Internet"
        Description="Default rights given to Internet applications">
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Access="Open"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Allowed="ApplicationIsolationByUser"
        UserQuota="512000"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Execution"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Window="SafeTopLevelWindows"
        Clipboard="OwnClipboard"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Level="SafePrinting"/>
</PermissionSet>
```

7. Everything (Allows unrestricted access to all resources covered by built-in permissions) =

```
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="Everything"
        Description="Allows unrestricted access to all resources covered by built-in
        permissions">
<IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
```

```
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.FileIOPermission, mscorlib, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.RegistryPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Assertion, UnmanagedCode, Execution, ControlThread, ControlEvidence,
        ControlPolicy, SerializationFormatter, ControlDomainPolicy, ControlPrincipal,
        ControlAppDomain, RemotingConfiguration, Infrastructure"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.KeyContainerPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Net.DnsPermission, System, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Net.SocketPermission, System, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Net.WebPermission, System, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
```

version="1"
Unrestricted="true"/>
<IPermission class="System.Diagnostics.EventLogPermission, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Security.Permissions.StorePermission, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Diagnostics.PerformanceCounterPermission, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Data.OleDb.OleDbPermission, System.Data, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Data.SqlClient.SqlClientPermission, System.Data, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Security.Permissions.DataProtectionPermission, System.Security, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
version="1"
Unrestricted="true"/>
</PermissionSet>

Level = User

Named Permission Sets:

1. FullTrust (Allows full access to all resources) =
<PermissionSet class="System.Security.NamedPermissionSet"
version="1"
Unrestricted="true"
Name="FullTrust"
Description="Allows full access to all resources"/>

2. SkipVerification (Grants right to bypass the verification) =
<PermissionSet class="System.Security.NamedPermissionSet"
version="1"
Name="SkipVerification"
Description="Grants right to bypass the verification">
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1"

```
Flags="SkipVerification" />
</PermissionSet>

3. Execution (Permits execution) =
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="Execution"
        Description="Permits execution">
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Execution"/>
</PermissionSet>

4. Nothing (Denies all resources, including the right to execute) =
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="Nothing"
        Description="Denies all resources, including the right to execute"/>

5. LocalIntranet (Default rights given to applications on the local intranet) =
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="LocalIntranet"
        Description="Default rights given to applications on the local intranet">
<IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Read="USERNAME"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Allowed="AssemblyIsolationByUser"
        UserQuota="9223372036854775807"
        Expiry="9223372036854775807"
        Permanent="True"/>
<IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="ReflectionEmit"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
```

version="1"
Flags="Assertion, Execution, BindingRedirects"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Net.DnsPermission, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
version="1"
Level="DefaultPrinting"/>
</PermissionSet>

6. Internet (Default rights given to Internet applications) =
<PermissionSet class="System.Security.NamedPermissionSet"
version="1"
Name="Internet"
Description="Default rights given to Internet applications">
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Access="Open"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Allowed="ApplicationIsolationByUser"
UserQuota="512000"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Flags="Execution"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Window="SafeTopLevelWindows"
Clipboard="OwnClipboard"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
version="1"
Level="SafePrinting"/>
</PermissionSet>

7. Everything (Allows unrestricted access to all resources covered by built-in permissions) =

```
<PermissionSet class="System.Security.NamedPermissionSet"
        version="1"
        Name="Everything"
        Description="Allows unrestricted access to all resources covered by built-in
        permissions">
<IPermission class="System.Security.Permissions.EnvironmentPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.FileDialogPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.FileIOPermission, mscorlib, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.IsolatedStorageFilePermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.ReflectionPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.RegistryPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Flags="Assertion, UnmanagedCode, Execution, ControlThread, ControlEvidence,
        ControlPolicy, SerializationFormatter, ControlDomainPolicy, ControlPrincipal,
        ControlAppDomain, RemotingConfiguration, Infrastructure, BindingRedirects"/>
<IPermission class="System.Security.Permissions.UIPermission, mscorlib, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Security.Permissions.KeyContainerPermission, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
        version="1"
        Unrestricted="true"/>
<IPermission class="System.Net.DnsPermission, System, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089"
        version="1"
```

Unrestricted="true"/>
<IPermission class="System.Drawing.Printing.PrintingPermission, System.Drawing, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
version="1"Unrestricted="true"/>
<IPermission class="System.Net.SocketPermission, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Net.WebPermission, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Diagnostics.EventLogPermission, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Security.Permissions.StorePermission, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Diagnostics.PerformanceCounterPermission, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true" >
<IPermission class="System.Data.OleDb.OleDbPermission, System.Data, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Data.SqlClient.SqlClientPermission, System.Data, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
version="1"
Unrestricted="true"/>
<IPermission class="System.Security.Permissions.DataProtectionPermission, System.Security, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
version="1"
Unrestricted="true"/>
</PermissionSet>

Success

4.4 IAVM Compliance

IAVM alerts, bulletins, and advisories were instituted to provide positive control of vulnerability notification and corresponding corrective action within DOD. All DOD program managers and system administrators, and/or other personnel responsible for system networks shall comply with the IAVM process. Security patches that address .NET vulnerabilities are

reviewed during an operating system security review and are not included in this checklist.

4.5 Version-specific Vulnerabilities

Vulnerabilities that apply only to a specific version of .NET are so noted. Versions to which the vulnerability does not apply should have that finding marked as N/A.

4.6 .NET Configuration File Location

Table 4-3: Location of Configuration Files

Level	Version	File
Enterprise	1.0	%SystemRoot%\Microsoft.NET\Framework\Config\v1.0.3705\Enterprisesec.config
Enterprise	1.1	%SystemRoot%\Microsoft.NET\Framework\Config\v1.1.4322\Enterprisesec.config
Enterprise	2.0, 3.0 & 3.5	%SystemRoot%\Microsoft.NET\Framework\Config\v2.0.50727\Enterprisesec.config
Machine	1.0	%SystemRoot%\Microsoft.NET\Framework\Config\v1.0.3705\Security.config
Machine	1.1	%SystemRoot%\Microsoft.NET\Framework\Config\v1.1.4322\Security.config
Machine	2.0, 3.0 & 3.5	%SystemRoot%\Microsoft.NET\Framework\Config\v2.0.50727\Security.config
User	1.0	*User Profile*\Application Data\Microsoft\CLR Security Config\v1.0.3705
User	1.1	*User Profile*\Application Data\Microsoft\CLR Security Config\v1.1.4322
User	2.0, 3.0 & 3.5	*User Profile*\Application Data\Microsoft\CLR Security Config\v2.0.50727.42

Note: Microsoft .Net versions 3.0 and 3.5 utilize the .Net 2.0 CLR and the same security configuration files.

Note: If configuration files are missing or corrupt the default configuration will be applied.

4.7 Reviewing Permissions with Code Access Security Policy Tool

The .NET Framework Code Access Security (CAS) Policy Tool caspol.exe may be used to review the .NET Framework code access security configuration. Each version of the .Net Framework comes with its own version of caspol.exe. Each version of caspol.exe can only be used to administer the .Net Framework version for which it was built. Use the version of caspol.exe found in the same directory structure as the .NET Framework version that is being reviewed. Security configurations for .Net 3.0 and 3.5 use the same underling CLR and share configuration information as .Net 2.0.

Note: Any findings found in .Net 2.0 should be marked in .Net 3.0 and .Net 3.5 Frameworks if they are installed.

Following are caspol.exe command line syntaxes for displaying the .NET code groups, permission sets, and trust assemblies. Issue all commands from the Windows command line accessed via the Windows Start>>Run>> open: cmd. Output may be directed to a text file with the use of the > redirection pipe. The resulting file may then be reviewed using the Windows Notepad or other text editor.

To list code groups for all levels, type the following command.

For .Net Framework Versions 1.0

C:\WINDOWS\Microsoft.NET\Framework\v1.0.3705\CasPol.exe -all -lg

For .Net Framework Versions 1.1

C:\WINDOWS\Microsoft.NET\Framework\v1.1.4322\CasPol.exe -all -lg

For .Net Framework Versions 2.0, 3.0 & 3.5

C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\CasPol.exe -all -lg

To list code group names and descriptions for all levels, type the following command.

For .Net Framework Versions 1.0

C:\WINDOWS\Microsoft.NET\Framework\v1.0.3705\CasPol.exe -all -ld

For .Net Framework Versions 1.1

C:\WINDOWS\Microsoft.NET\Framework\v1.1.4322\CasPol.exe -all -ld

For .Net Framework Versions 2.0, 3.0 & 3.5

C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\CasPol.exe -all -ld

To list permission sets for all levels, type the following command.

For .Net Framework Versions 1.0

C:\WINDOWS\Microsoft.NET\Framework\v1.0.3705\CasPol.exe -all -lp

For .Net Framework Versions 1.1

C:\WINDOWS\Microsoft.NET\Framework\v1.1.4322\CasPol.exe -all -lp

For .Net Framework Versions 2.0, 3.0 & 3.5

C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\CasPol.exe -all -lp

Caspol.exe can typically be found in the %systemroot%\Microsoft.NET\Framework\<version>\ directory.

Microsoft .Net version 2.0, 3.0 & 3.5 use the same underlying Common Language Runtime (CLR) is and share the same configuration files.

4.8 Reviewing Software Publishing State Values

Use regedit to determine the value of the registry keys for all users of the machine. All values for software publishing should be reviewed.

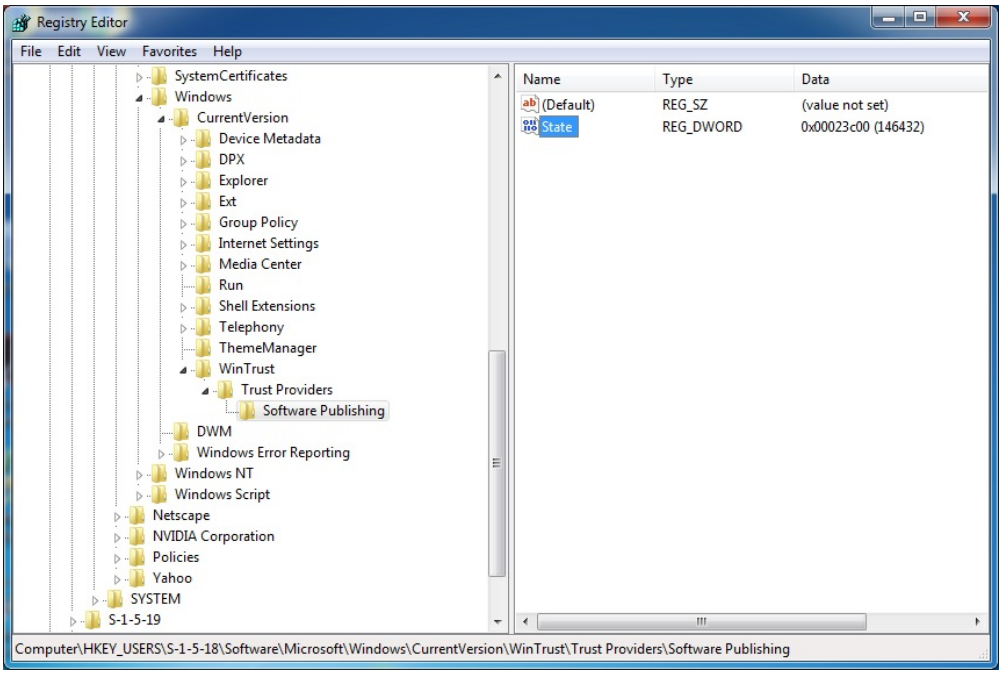
Example Locations:
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\WinTrust\Trust Providers\Software Publishing\State

HKEY_USERS\S-1-5-18\Software\Microsoft\Windows\CurrentVersion\WinTrust\Trust Providers\Software Publishing\State

HKEY_USERS\S-1-5-19\Software\Microsoft\Windows\CurrentVersion\WinTrust\Trust Providers\Software Publishing\State

HKEY_USERS\S-1-5-20\Software\Microsoft\Windows\CurrentVersion\WinTrust\Trust Providers\Software Publishing\State

Figure 4-1: Review Software Publishing State Values



Convert the State DWORD value to binary with the hexadecimal calculator.

For Example:

If the registry setting is 23c00 Hex it converts to 00100011110000000000 Binary with 2 leading zeros for 19 bits total.

Use Table 4-4 to determine the individual values for software publishing.

Table 4-4: Software Publishing State Value Table

Hex	2				3				C				0				0			
Binary	0	0	1	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
		If Bit 19 = 1 Only trust items found in the Trust DB = TRUE	If Bit 18 = 1 Check the revocation list on Time Stamp	If Bit 17 = 1 Invalidate version 1 signed objects = TRUE			If Bit 14 = 1 Java offline revocation server OK (Commercial) = TRUE	If Bit 13 = 1 Java offline revocation server OK (Individual) = TRUE	If Bit 12 = 1 Offline revocation server OK (Commercial) = TRUE	If Bit 11 = 1 Offline revocation server OK (Individual) = TRUE	If Bit 10 = 1 Check the revocation list = FALSE If Bit	If Bit 9 = 1 Use expiration date on certificates =	If Bits 6 & 8 = 0 Trust the Test Root = FALSE		If Bits 6 & 8 = 1 Trust the Test Root = TRUE					
Only trust items found in the Trust DB = FALSE																				
Check the revocation list on Time Stamp Signer = FALSE																				
Invalidate version 1 signed objects = FALSE																				
Java offline revocation server OK (Commercial) = TRUE																				
Java offline revocation server OK (Individual) = TRUE																				
Offline revocation server OK (Commercial) = TRUE																				
Offline revocation server OK (Individual) = TRUE																				
Check the revocation list = TRUE																				

5. .NET SECURITY FRAMEWORK CHECKS AND PROCEDURES

The following instructions should be used for checks APPNET0001 through APPNET0030. Use the permission name specified in the individual instruction to determine the specific permission to review. Any instructions included under a check are specific to that check and should be followed.

Performing a SRR on the .NET Framework involves identifying which permission sets have potentially dangerous permissions. After determining which permission sets are of interest the code groups to which those permission sets are assigned must be identified. Once the code groups are identified the membership conditions for those code groups must be documented. When determining whether or not a given vulnerability exists the reviewer must evaluate the membership conditions of code groups that grant the permission and determine if the permission is restricted to the appropriate assemblies. When entering the finding details the reviewer should include the name of the permission set(s) and the code group(s) that grant the permission set(s).

5.1 APPNET0001: File IO Permission

Description: The *File IO* permission allows an application to access system files directly.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.FileIOPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
<IPermission class="**System.Security.Permissions.FileIOPermission**, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
1.1. Zone - MyComputer: FullTrust
1.1.1. StrongName -
0024000004800000940000000602000000240000525341310004000001000100007D1FA57C4A

ED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DC
C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E82
1C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0
FC4963D261C8A12436518206DC093344D5AD293: FullTrust

1.1.2. StrongName - 00000000000000000400000000000000: FullTrust

1.2. Zone - Intranet: LocalIntranet

1.2.1. All code: Same site Web

1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'

1.3. Zone - Internet: Internet

1.3.1. All code: Same site Web

1.4. Zone - Untrusted: Nothing

1.5. Zone - Trusted: Internet

1.5.1. All code: Same site Web

1.6. ApplicationDirectory: Dev

Validate:

1. If a Permission Set with the *File IO* permission of *Grant Assemblies unrestricted access to the file system (unrestricted="true")*is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.
2. If a Permission Set granting limited *File IO* permissions is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	File IO Permission	
Reference:	.NET Framework Security Guide	IA Control: ECCD-1, ECCD-2

5.2 APPNET0003: Isolated Storage Permission

Description: The *Isolated Storage* permission is used to allow applications to store temporary data to a local user data store.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.IsolatedStorageFilePermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
<IPermission class="**System.Security.Permissions.IsolatedStorageFilePermission**, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1"

Allowed="AdministerIsolatedStorageByUser" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
  1.1. Zone - MyComputer: FullTrust
    1.1.1. StrongName - 00240000048000009400000006020000002400005253413100040000010001000 7D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DCC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
    1.1.2. StrongName - 00000000000000000400000000000000: FullTrust
  1.2. Zone - Intranet: LocalIntranet
    1.2.1. All code: Same site Web
    1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
  1.3. Zone - Internet: Internet
    1.3.1. All code: Same site Web
  1.4. Zone - Untrusted: Nothing
  1.5. Zone - Trusted: Internet
    1.5.1. All code: Same site Web
  1.6. ApplicationDirectory: Dev

Validate:

1. If the *Isolated Storage* permission of *Grant assemblies unrestricted access to file-based storage (unrestricted=”true”)* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.
2. If the *Isolated Storage* permission *Administer Isolated Storage by User* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.
3. If the *Isolated Storage* permissions *Assembly Isolation by User* or *Assembly Isolation by Roaming User* is assigned to a Non-default Code Group whose membership criteria has not been evaluated and approved by the ISSO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Isolated Storage Permission	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

5.3 APPNET0004: User Interface Permission (Windowing)

Description: The User Interface Permission for windowing controls access to user interface windows.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.UIPermission for Windows.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
<IPermission class="**System.Security.Permissions.UIPermission**, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1"
Window="SafeSubWindows" Clipboard="OwnClipboard" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
  1.1. Zone - MyComputer: FullTrust
    1.1.1. StrongName - 0024000004800000940000000602000000240000525341310004000001000100007D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DCC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
    1.1.2. StrongName - 0000000000000000040000000000000000: FullTrust
  1.2. Zone - Intranet: LocalIntranet
    1.2.1. All code: Same site Web
    1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
  1.3. Zone - Internet: Internet
    1.3.1. All code: Same site Web
  1.4. Zone - Untrusted: Nothing

1.5. Zone - Trusted: Internet
1.5.1. All code: Same site Web
1.6. ApplicationDirectory: Dev

Validate:

1. If the *User Interface* permission of *Grant assemblies unrestricted access to user interface elements (unrestricted=”true”)* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding..
2. If the *User Interface* permission *All Windows Events* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as its’ membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.
3. If the *User Interface* permissions *Safe Top Level Windows* is assigned to a Non-default Code Group whose membership criteria has not been evaluated and approved by the ISSO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	User Interface Permission (Windowing)	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

5.4 APPNET0005: User Interface Permission (Clipboard)

Description: The User Interface Permission for clipboard controls application access to clipboards used by the user or other applications.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.UIPermission for Clipboard.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
<IPermission class="**System.Security.Permissions.UIPermission**, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Window="SafeSubWindows" **Clipboard="OwnClipboard"** />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
  1.1. Zone - MyComputer: FullTrust
    1.1.1. StrongName - 0024000004800000940000000602000000240000525341310004000001000100007D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DCC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
    1.1.2. StrongName - 0000000000000000040000000000000: FullTrust
  1.2. Zone - Intranet: LocalIntranet
    1.2.1. All code: Same site Web
    1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
  1.3. Zone - Internet: Internet
    1.3.1. All code: Same site Web
  1.4. Zone - Untrusted: Nothing
  1.5. Zone - Trusted: Internet
    1.5.1. All code: Same site Web
  1.6. ApplicationDirectory: Dev

Validate:

1. If any *User Interface* permission other than *No Clipboard* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as its' membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	User Interface Permission (Clipboard)	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

5.5 APPNET0006: Reflection Permission

Description: The Reflection permission controls an application's discovery of other system resources and applications.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.ReflectionPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
<IPermission class="**System.Security.Permissions.ReflectionPermission**, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
  1.1. Zone - MyComputer: FullTrust
    1.1.1. StrongName - 00240000048000009400000006020000002400005253413100040000010001000 7D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DCC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
    1.1.2. StrongName - 00000000000000000400000000000000: FullTrust
  1.2. Zone - Intranet: LocalIntranet
    1.2.1. All code: Same site Web
    1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
  1.3. Zone - Internet: Internet
    1.3.1. All code: Same site Web
  1.4. Zone - Untrusted: Nothing
  1.5. Zone - Trusted: Internet
    1.5.1. All code: Same site Web
  1.6. ApplicationDirectory: Dev

Validate:

1. If the *Reflection* permission of *Grant assemblies unrestricted permission to discover information about other assemblies (unrestricted=”true”)* is assigned to a Non- default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.
2. If the *Reflection* permission *Member* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as its’ membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

3. If the *Reflection* permissions *Type* is assigned to a Non-default Code Group whose membership criteria has not been reviewed and approved by the ISSO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Reflection Permission	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

5.6 APPNET0007: Printing Permission

Description: The Printing permission controls application access to system printing resources.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Drawing.Printing.PrintingPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
<IPermission class="**System.Drawing.Printing.PrintingPermission**, System.Drawing, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" version="1" Unrestricted="true" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
1.1. Zone - MyComputer: FullTrust
1.1.1. StrongName - 00240000048000009400000006020000002400005253413100040000010001000 7D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DCC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
1.1.2. StrongName - 0000000000000000040000000000000000: FullTrust
1.2. Zone - Intranet: LocalIntranet
1.2.1. All code: Same site Web
1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'

1.3. Zone - Internet: Internet
  1.3.1. All code: Same site Web
1.4. Zone - Untrusted: Nothing
1.5. Zone - Trusted: Internet
  1.5.1. All code: Same site Web
1.6. ApplicationDirectory: Dev

Validate:

1. If the *Printing* permission of *Grant assemblies unrestricted access to printers (unrestricted="true")* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.
2. If the *Printing* permission *All Printing* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as its' membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Printing Permission	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

5.7 APPNET0008: DNS Permission

Description: The DNS permission controls application access to DNS resources available to the host system.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Net.DnsPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
  <IPermission class="**System.Net.DnsPermission**, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
  1.1. Zone - MyComputer: FullTrust
    1.1.1. StrongName -
00240000048000009400000006020000002400005253413100040000010001000 7D1FA57C4A
ED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DC
C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E82
1C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0
FC4963D261C8A12436518206DC093344D5AD293: FullTrust
    1.1.2. StrongName - 00000000000000000400000000000000: FullTrust
  1.2. Zone - Intranet: LocalIntranet
    1.2.1. All code: Same site Web
    1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
  1.3. Zone - Internet: Internet
    1.3.1. All code: Same site Web
  1.4. Zone - Untrusted: Nothing
  1.5. Zone - Trusted: Internet
    1.5.1. All code: Same site Web
  1.6. ApplicationDirectory: Dev

Validate:

1. If the *DNS* permission of *Grant assemblies unrestricted access to DNS (unrestricted=”true”)* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	DNS Permission	
Reference:	.NET Framework Security Guide	IA Control: ECLP-1

5.8 APPNET0009: Socket Access Permission

Description: The Socket Access permission controls application access to network ports defined on the host system.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Net.SocketPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
  <IPermission class="**System.Net.SocketPermission**, System, Version=2.0.0.0,
  Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true" />

</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
  1.1. Zone - MyComputer: FullTrust
    1.1.1. StrongName - 00240000048000009400000006020000002400005253413100040000010001000 7D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DCC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
    1.1.2. StrongName - 0000000000000000040000000000000: FullTrust
  1.2. Zone - Intranet: LocalIntranet
    1.2.1. All code: Same site Web
    1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
  1.3. Zone - Internet: Internet
    1.3.1. All code: Same site Web
  1.4. Zone - Untrusted: Nothing
  1.5. Zone - Trusted: Internet
    1.5.1. All code: Same site Web
  1.6. ApplicationDirectory: Dev

Validate:

1. If the *Socket Access* permission of *Grant assemblies unrestricted access to sockets (unrestricted=”true”)* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.
2. Ask the System Administrator if any *Socket Access* permissions are granted to Non-default Code groups that do not provide networking services. If these permissions exist then this is a finding.
3. Ask the System Administrator if any *Socket Access* permissions are granted to Non-default Code groups to hosts outside the enclave. If these permissions exist then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Socket Access Permission	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

5.9 APPNET0010: Web Access Permission

Description: The Web Access permission controls application access to HTTP requests to designated URLs or the configuration of HTTP settings.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Net.WebPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
<IPermission class="**System.Net.WebPermission**, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
  1.1. Zone - MyComputer: FullTrust
    1.1.1. StrongName - 0024000004800000940000000602000000240000525341310004000001000100007D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DCC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
    1.1.2. StrongName - 0000000000000000040000000000000000: FullTrust
  1.2. Zone - Intranet: LocalIntranet
    1.2.1. All code: Same site Web
    1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
  1.3. Zone - Internet: Internet
    1.3.1. All code: Same site Web
  1.4. Zone - Untrusted: Nothing
  1.5. Zone - Trusted: Internet
    1.5.1. All code: Same site Web
  1.6. ApplicationDirectory: Dev

Validate:

1. If the *Web Access* permission of *Grant assemblies unrestricted access to Web Sites*

(unrestricted="true") is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

2. If specific URL(s) (*Web Access* permissions) are assigned to a Non-default Code Group whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID : APPNET10	Web Access Permission	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

5.10 APPNET0011: Message Queue Permission

Description: The Message Queue permission controls application access to communications across the network.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Messaging.MessageQueuePermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
<IPermission class="**System.Messaging.MessageQueuePermission**, System.Messaging, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" version="1" Unrestricted="true" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
1.1. Zone - MyComputer: FullTrust
1.1.1. StrongName -
00240000048000009400000006020000002400005253413100040000010001000 7D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DCC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust

1.1.2. StrongName - 0000000000000000040000000000000: FullTrust
1.2. Zone - Intranet: LocalIntranet
1.2.1. All code: Same site Web
1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
1.3. Zone - Internet: Internet
1.3.1. All code: Same site Web
1.4. Zone - Untrusted: Nothing
1.5. Zone - Trusted: Internet
1.5.1. All code: Same site Web
1.6. ApplicationDirectory: Dev

Validate:

1. If the *Message Queue* permission of *Grant assemblies unrestricted access to all message queues (unrestricted=”true”)* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.
2. If the *Message Queue* permission *Administer* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding..
3. If the *Message Queue* permission *Browse* is assigned to a Non-default Code Group whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Message Queue Permission	
Reference:	.NET Framework Security Guide	IA Control: ECLP-1

5.11 APPNET0012: Service Controller Permission

Description: The Service Controller permission controls application access to the control of Windows services.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.ServiceProcess.ServiceControllerPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
<IPermission class="**System.ServiceProcess.ServiceControllerPermission**,
System.ServiceProcess, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a" version="1" Unrestricted="true" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
  1.1. Zone - MyComputer: FullTrust
    1.1.1. StrongName - 00240000048000009400000006020000002400005253413100040000010001000 7D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DCC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
    1.1.2. StrongName - 00000000000000000400000000000000: FullTrust
  1.2. Zone - Intranet: LocalIntranet
    1.2.1. All code: Same site Web
    1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
  1.3. Zone - Internet: Internet
    1.3.1. All code: Same site Web
  1.4. Zone - Untrusted: Nothing
  1.5. Zone - Trusted: Internet
    1.5.1. All code: Same site Web
  1.6. ApplicationDirectory: Dev

Validate:

1. If the any *Service Controller* permissions are assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as its' membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Service Controller Permission	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

5.12 APPNET0013: Database Permission

Description: The Database permissions control application access to databases defined on the host system.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Data.SqlClient.SqlClientPermission or System.Data.OleDb.OleDbPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
<IPermission class="**System.Data.SqlClient.SqlClientPermission**, System.Data, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true" />
<IPermission class="**System.Data.OleDb.OleDbPermission**, System.Data, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
  1.1. Zone - MyComputer: FullTrust
    1.1.1. StrongName - 00240000048000009400000006020000002400005253413100040000010001000 7D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DCC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
    1.1.2. StrongName - 0000000000000000040000000000000: FullTrust
  1.2. Zone - Intranet: LocalIntranet
    1.2.1. All code: Same site Web
    1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
  1.3. Zone - Internet: Internet
    1.3.1. All code: Same site Web
  1.4. Zone - Untrusted: Nothing
  1.5. Zone - Trusted: Internet
    1.5.1. All code: Same site Web
  1.6. ApplicationDirectory: Dev

Validate:

1. If the *SQLClientPermission or OLEDBPermission*permission (*Grant assemblies unrestricted access to all providers (unrestricted=”true”)* is assigned to a Non- default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Database Permission	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

Checks APPNET014 – APPNET025 refer to the Security permission contained within a permission set assigned to a non-default code group. Review all permission sets that include the security permission.

5.13 APPNET0014: Security Permission (Extend Infrastructure)

Description: The Security permission Extend Infrastructure controls application access to message processing.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.SecurityPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
<IPermission class="**System.Security.Permissions.SecurityPermission**, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Flags="Assertion, UnmanagedCode, SkipVerification, Execution, ControlThread, ControlEvidence, ControlPolicy, SerializationFormatter, ControlDomainPolicy, ControlPrincipal, ControlAppDomain, RemotingConfiguration, **Infrastructure**" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
1.1. Zone - MyComputer: FullTrust
1.1.1. StrongName -
00240000048000009400000006020000002400005253413100040000010001000 7D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DCC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust

1.1.2. StrongName - 00000000000000000400000000000000: FullTrust
1.2. Zone - Intranet: LocalIntranet
1.2.1. All code: Same site Web
1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
1.3. Zone - Internet: Internet
1.3.1. All code: Same site Web
1.4. Zone - Untrusted: Nothing
1.5. Zone - Trusted: Internet
1.5.1. All code: Same site Web
1.6. ApplicationDirectory: Dev

Validate:

1. If the *Security* permission of *Extend Infrastructure* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Security Permission (Extend Infrastructure)	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

5.14 APPNET0015: Security Permission (Enable Remoting Configuration)

Description: The Security permission Enable Remoting Configuration defines thecommunication channels available to an application.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.SecurityPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
<IPermission class="**System.Security.Permissions.SecurityPermission**, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Flags="Assertion, UnmanagedCode, SkipVerification, Execution, ControlThread, ControlEvidence, ControlPolicy, SerializationFormatter, ControlDomainPolicy, ControlPrincipal, ControlAppDomain, **RemotingConfiguration,** Infrastructure" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:

Level = Machine

Code Groups:

1. All code: Nothing
  1.1. Zone - MyComputer: FullTrust
    1.1.1. StrongName - 0024000004800000940000000602000000240000525341310004000001000100007D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DCC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
    1.1.2. StrongName - 0000000000000000040000000000000000: FullTrust
  1.2. Zone - Intranet: LocalIntranet
    1.2.1. All code: Same site Web
    1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
  1.3. Zone - Internet: Internet
    1.3.1. All code: Same site Web
  1.4. Zone - Untrusted: Nothing
  1.5. Zone - Trusted: Internet
    1.5.1. All code: Same site Web
  1.6. ApplicationDirectory: Dev

Validate:

1. If the *Security* permission of *Enable remoting configuration (Flags=”RemotingConfiguration”)* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Security Permission (Enable Remoting configuration)	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

5.15 APPNET0016: Security Permission (Enable Serialization Formatter)

Description: The Security permission Enable Serialization Formatter controls access to serialized data. Serialized data is data formatted into a series of bits for storing or transmitting.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.SecurityPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
<IPermission class="**System.Security.Permissions.SecurityPermission**, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Flags="Assertion, UnmanagedCode, SkipVerification, Execution, ControlThread, ControlEvidence, ControlPolicy, **SerializationFormatter**, ControlDomainPolicy, ControlPrincipal, ControlAppDomain, RemotingConfiguration, Infrastructure" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
  1.1. Zone - MyComputer: FullTrust
    1.1.1. StrongName - 00240000048000009400000006020000002400005253413100040000010001000 7D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DCC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
    1.1.2. StrongName - 0000000000000000040000000000000000: FullTrust
  1.2. Zone - Intranet: LocalIntranet
    1.2.1. All code: Same site Web
    1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
  1.3. Zone - Internet: Internet
    1.3.1. All code: Same site Web
  1.4. Zone - Untrusted: Nothing
  1.5. Zone - Trusted: Internet
    1.5.1. All code: Same site Web
  1.6. ApplicationDirectory: Dev

Validate:

1. If the *Security* permission of *Enable Serialization Formatter (Flags=" SerializationFormatter")* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Security Permission (Enable Serialization Formatter)	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

5.16 APPNET0017: Security Permission (Enable Thread Control)

Description: The Security permission Enable Thread Control is used to control application access to abort, suspend, or resume its threads.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.SecurityPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
<IPermission class="**System.Security.Permissions.SecurityPermission**, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Flags="Assertion, UnmanagedCode, SkipVerification, Execution, **ControlThread**, ControlEvidence, ControlPolicy, SerializationFormatter, ControlDomainPolicy, ControlPrincipal, ControlAppDomain, RemotingConfiguration, Infrastructure" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
  1.1. Zone - MyComputer: FullTrust
    1.1.1. StrongName - 00240000048000009400000006020000002400005253413100040000010001000 7D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DCC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
    1.1.2. StrongName - 0000000000000000040000000000000: FullTrust
  1.2. Zone - Intranet: LocalIntranet
    1.2.1. All code: Same site Web
    1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
  1.3. Zone - Internet: Internet

1.3.1. All code: Same site Web
1.4. Zone - Untrusted: Nothing
1.5. Zone - Trusted: Internet
1.5.1. All code: Same site Web
1.6. ApplicationDirectory: Dev

Validate:

1. If the *Security* permission of *Enable thread control (Flags="ControlThread")* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Security Permission (Enable Thread Control)	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

5.17 APPNET0018: Security Permission (Allow Principal Control)

Description: The Security permission Allow Principal control controls application access to Windows user information.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.SecurityPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
<IPermission class="**System.Security.Permissions.SecurityPermission**, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Flags="Assertion, UnmanagedCode, SkipVerification, Execution, ControlThread, ControlEvidence, ControlPolicy, SerializationFormatter, ControlDomainPolicy, **ControlPrincipal**, ControlAppDomain, RemotingConfiguration, Infrastructure" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing

1.1. Zone - MyComputer: FullTrust
1.1.1. StrongName -
00240000048000009400000006020000002400005253413100040000010001000 7D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DCC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
1.1.2. StrongName - 00000000000000000400000000000000: FullTrust
1.2. Zone - Intranet: LocalIntranet
1.2.1. All code: Same site Web
1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
1.3. Zone - Internet: Internet
1.3.1. All code: Same site Web
1.4. Zone - Untrusted: Nothing
1.5. Zone - Trusted: Internet
1.5.1. All code: Same site Web
1.6. ApplicationDirectory: Dev

Validate:

1. If the *Security* permission of *Allow principal control (Flags=”ControlPrincipal”)* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Security Permission (Allow Principal Control)	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

5.18 APPNET0019: Security Permission (Enable Assembly Execution)

Description: The Security permission Enable Assembly Execution allows applications to execute.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.SecurityPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
<IPermission class="**System.Security.Permissions.SecurityPermission**, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Flags="Assertion, UnmanagedCode, SkipVerification, **Execution,** ControlThread, ControlEvidence, ControlPolicy, SerializationFormatter, ControlDomainPolicy,

ControlPrincipal, ControlAppDomain, RemotingConfiguration, Infrastructure" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
  1.1. Zone - MyComputer: FullTrust
    1.1.1. StrongName - 00240000048000009400000006020000002400005253413100040000010001000 7D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DCC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
    1.1.2. StrongName - 0000000000000000040000000000000: FullTrust
  1.2. Zone - Intranet: LocalIntranet
    1.2.1. All code: Same site Web
    1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
  1.3. Zone - Internet: Internet
    1.3.1. All code: Same site Web
  1.4. Zone - Untrusted: Nothing
  1.5. Zone - Trusted: Internet
    1.5.1. All code: Same site Web
  1.6. ApplicationDirectory: Dev

Validate:

1. If the *Security* permission of *Enable assembly execution* (*Flags=”Execution”)* is in a permission set that is assigned to a Non-default Code Group with a *Zone* membership condition then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Security Permission (Enable Assembly Execution)	
Reference:	.NET Framework Security Guide	IA Control: ECLP-1

5.19 APPNET0020: Security Permission (Skip Verification)

Description: The Security permission Skip Verification controls the execution of code that is verified as being type safe.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.SecurityPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
<IPermission class="**System.Security.Permissions.SecurityPermission**, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Flags="Assertion, UnmanagedCode, **SkipVerification**, Execution, ControlThread, ControlEvidence, ControlPolicy, SerializationFormatter, ControlDomainPolicy, ControlPrincipal, ControlAppDomain, RemotingConfiguration, Infrastructure" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
  1.1. Zone - MyComputer: FullTrust
    1.1.1. StrongName - 0024000004800000940000000602000000240000525341310004000001000100007D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DCC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
    1.1.2. StrongName - 0000000000000000040000000000000000: FullTrust
  1.2. Zone - Intranet: LocalIntranet
    1.2.1. All code: Same site Web
    1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
  1.3. Zone - Internet: Internet
    1.3.1. All code: Same site Web
  1.4. Zone - Untrusted: Nothing
  1.5. Zone - Trusted: Internet
    1.5.1. All code: Same site Web
  1.6. ApplicationDirectory: Dev

Validate:

1. If the *Security* permission of *Skip verification (Flags=”SkipVerification”)* is assigned to any non-default Code Group then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Security Permission (Skip Verification)	
Reference:	.NET Framework Security Guide	IA Control: ECLP-1

5.20 APPNET0021: Security Permission (Allow Calls to Unmanaged Assemblies)

Description: The Security permission Allow Calls to Unmanaged Assemblies controls application access to applications not managed by the .Net Framework.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.SecurityPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
<IPermission class="**System.Security.Permissions.SecurityPermission**, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Flags="Assertion, **UnmanagedCode**, SkipVerification, Execution, ControlThread, ControlEvidence, ControlPolicy, SerializationFormatter, ControlDomainPolicy, ControlPrincipal, ControlAppDomain, RemotingConfiguration, Infrastructure" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
  1.1. Zone - MyComputer: FullTrust
    1.1.1. StrongName - 00240000048000009400000006020000002400005253413100040000010001000 7D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DCC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
    1.1.2. StrongName - 0000000000000000040000000000000: FullTrust
  1.2. Zone - Intranet: LocalIntranet
    1.2.1. All code: Same site Web
    1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
  1.3. Zone - Internet: Internet

1.3.1. All code: Same site Web
1.4. Zone - Untrusted: Nothing
1.5. Zone - Trusted: Internet
1.5.1. All code: Same site Web
1.6. ApplicationDirectory: Dev

Validate:

1. If the *Security* permission of *Allow calls to unmanaged assemblies (Flags="UnmanagedCode")* is assigned to a non-default Code Group then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Security Permission (Allow Calls to Unmanaged Assemblies)	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

5.21 APPNET0022: Security Permission (Allow Policy Control)

Description: The Security permission Allow Policy Control controls application access to it's the current security policy configuration.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.SecurityPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
<IPermission class="**System.Security.Permissions.SecurityPermission**, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Flags="Assertion, UnmanagedCode, SkipVerification, Execution, ControlThread, ControlEvidence, **ControlPolicy**, SerializationFormatter, ControlDomainPolicy, ControlPrincipal, ControlAppDomain, RemotingConfiguration, Infrastructure" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
1.1. Zone - MyComputer: FullTrust

1.1.1. StrongName - 0024000004800000940000000602000000240000525341310004000001000100007D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DCC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust

1.1.2. StrongName - 0000000000000000040000000000000000: FullTrust

1.2. Zone - Intranet: LocalIntranet

1.2.1. All code: Same site Web

1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'

1.3. Zone - Internet: Internet

1.3.1. All code: Same site Web

1.4. Zone - Untrusted: Nothing

1.5. Zone - Trusted: Internet

1.5.1. All code: Same site Web

1.6. ApplicationDirectory: Dev

Validate:

1. If the *Security* permission of *Allow Policy Control (Flags=”ControlPolicy”)* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Security Permission (Allow Policy Control)	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

5.22 APPNET0023: Security Permission (Allow Domain Policy Control)

Description: The Security permission Allow Domain Policy controls defines application access to its own application domain security policy.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.SecurityPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
<IPermission class="**System.Security.Permissions.SecurityPermission**, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Flags="Assertion, UnmanagedCode, SkipVerification, Execution, ControlThread, ControlEvidence, ControlPolicy, SerializationFormatter, **ControlDomainPolicy**, ControlPrincipal, ControlAppDomain, RemotingConfiguration, Infrastructure" />

</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
  1.1. Zone - MyComputer: FullTrust
    1.1.1. StrongName - 00240000048000009400000006020000002400005253413100040000010001000 7D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DCC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
    1.1.2. StrongName - 00000000000000000400000000000000: FullTrust
  1.2. Zone - Intranet: LocalIntranet
    1.2.1. All code: Same site Web
    1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
  1.3. Zone - Internet: Internet
    1.3.1. All code: Same site Web
  1.4. Zone - Untrusted: Nothing
  1.5. Zone - Trusted: Internet
    1.5.1. All code: Same site Web
  1.6. ApplicationDirectory: Dev

Validate:

1. If the *Security* permission of *Allow domain policy control (Flags="ControlDomainPolicy")* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Security Permission (Allow Domain Policy Control)	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

5.23 APPNET0024: Security Permission (Allow Evidence Control)

Description: The Security permission Allow Evidence Control is used to control an application's access to supply or modify evidence used to determine access to system resources.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.SecurityPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
<IPermission class="**System.Security.Permissions.SecurityPermission**, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Flags="Assertion, UnmanagedCode, SkipVerification, Execution, ControlThread, **ControlEvidence**, ControlPolicy, SerializationFormatter, ControlDomainPolicy, ControlPrincipal, ControlAppDomain, RemotingConfiguration, Infrastructure" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
  1.1. Zone - MyComputer: FullTrust
    1.1.1. StrongName - 0024000004800000940000000602000000240000525341310004000001000100007D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DCC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
    1.1.2. StrongName - 0000000000000000040000000000000000: FullTrust
  1.2. Zone - Intranet: LocalIntranet
    1.2.1. All code: Same site Web
    1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
  1.3. Zone - Internet: Internet
    1.3.1. All code: Same site Web
  1.4. Zone - Untrusted: Nothing
  1.5. Zone - Trusted: Internet
    1.5.1. All code: Same site Web
  1.6. ApplicationDirectory: Dev

Validate:

1. If the *Security* permission of *Allow evidence control (Flags="ControlEvidence")* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed

and approved by the ISSO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Security Permission (Allow Evidence Control)	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

5.24 APPNET0025: Security Permission (Assert any Permission that Has Been Granted)

Description: The Security permission Assert any Permission that Has Been Granted controls application access to permissions assigned to any code in the assembly that called it.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.SecurityPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
<IPermission class="**System.Security.Permissions.SecurityPermission**, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Flags="**Assertion**, UnmanagedCode, SkipVerification, Execution, ControlThread, ControlEvidence, ControlPolicy, SerializationFormatter, ControlDomainPolicy, ControlPrincipal, ControlAppDomain, RemotingConfiguration, Infrastructure" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
1.1. Zone - MyComputer: FullTrust
1.1.1. StrongName - 00240000048000009400000006020000002400005253413100040000010001000 7D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DCC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
1.1.2. StrongName - 0000000000000000040000000000000: FullTrust
1.2. Zone - Intranet: LocalIntranet
1.2.1. All code: Same site Web

1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
1.3. Zone - Internet: Internet
1.3.1. All code: Same site Web
1.4. Zone - Untrusted: Nothing
1.5. Zone - Trusted: Internet
1.5.1. All code: Same site Web
1.6. ApplicationDirectory: Dev

Validate

1. If the *Security* permission of *Assert (Flags=”Assertion”)* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Security Permission (Assert any Permission that Has Been Granted)	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

5.25 APPNET0026: Performance Counter Permission

Description: The Performance Counter permission controls application access to system performance monitoring resources.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Diagnostics.PerformanceCounterPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
<IPermission class="**System.Diagnostics.PerformanceCounterPermission**, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1">
<Machine name="server1">
<Category name="admin" access="Administer" />
</Machine>
</IPermission>
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
  1.1. Zone - MyComputer: FullTrust
    1.1.1. StrongName -
00240000048000009400000006020000002400005253413100040000010001000 7D1FA57C4A
ED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DC
C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E82
1C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0
FC4963D261C8A12436518206DC093344D5AD293: FullTrust
    1.1.2. StrongName - 00000000000000000400000000000000: FullTrust
  1.2. Zone - Intranet: LocalIntranet
    1.2.1. All code: Same site Web
    1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
  1.3. Zone - Internet: Internet
    1.3.1. All code: Same site Web
  1.4. Zone - Untrusted: Nothing
  1.5. Zone - Trusted: Internet
    1.5.1. All code: Same site Web
  1.6. ApplicationDirectory: Dev

Validate:

1. If the *Performance Counter* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Performance Counter Permission	
Reference:	.NET Framework Security Guide pg	IA Control: ECLP-1

5.26 APPNET0027: Environment Variables Permission

Description: The Environment Variables permission controls application access to system environment variables and to other system resource names.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.EnvironmentPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
  <IPermission class="**System.Security.Permissions.EnvironmentPermission**, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1"

Unrestricted="true" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
  1.1. Zone - MyComputer: FullTrust
    1.1.1. StrongName - 00240000048000009400000006020000002400005253413100040000010001000 7D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DCC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
    1.1.2. StrongName - 00000000000000000400000000000000: FullTrust
  1.2. Zone - Intranet: LocalIntranet
    1.2.1. All code: Same site Web
    1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
  1.3. Zone - Internet: Internet
    1.3.1. All code: Same site Web
  1.4. Zone - Untrusted: Nothing
  1.5. Zone - Trusted: Internet
    1.5.1. All code: Same site Web
  1.6. ApplicationDirectory: Dev

Validate:

1. If the *Environment Variables* permission of *Grant assemblies access to all environment variables (unrestricted=”true”)* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Environment Variables Permission	
Reference:	.NET Framework Security Guide	IA Control: ECLP-1

5.27 APPNET0028: Event Log Permission

Description: The Event Log permission controls application access to event log resources defined on the system.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Diagnostics.EventLogPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
<IPermission class="**System.Diagnostics.EventLogPermission**, System, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
1.1. Zone - MyComputer: FullTrust
1.1.1. StrongName -
00240000048000009400000006020000002400005253413100040000010001000 7D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DCC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
1.1.2. StrongName - 00000000000000000400000000000000: FullTrust
1.2. Zone - Intranet: LocalIntranet
1.2.1. All code: Same site Web
1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
1.3. Zone - Internet: Internet
1.3.1. All code: Same site Web
1.4. Zone - Untrusted: Nothing
1.5. Zone - Trusted: Internet
1.5.1. All code: Same site Web
1.6. ApplicationDirectory: Dev

Validate:

1. If the *Event Log* permission of *Grant assemblies unrestricted access to all event logs*
2. *(unrestricted=”true”)* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Event Log Permission	
Reference:	.NET Framework Security Guide pg	IA Control: ECLP-1

5.28 APPNET0029: Registry Permission

Description: The Registry permission controls application access to the Windows registry.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.Security.Permissions.RegistryPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
<IPermission class="**System.Security.Permissions.RegistryPermission**, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Unrestricted="true" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
1.1. Zone - MyComputer: FullTrust
1.1.1. StrongName - 00240000048000009400000006020000002400005253413100040000010001000 7D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DCC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
1.1.2. StrongName - 00000000000000000400000000000000: FullTrust
1.2. Zone - Intranet: LocalIntranet
1.2.1. All code: Same site Web
1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
1.3. Zone - Internet: Internet
1.3.1. All code: Same site Web
1.4. Zone - Untrusted: Nothing
1.5. Zone - Trusted: Internet

1.5.1. All code: Same site Web
1.6. ApplicationDirectory: Dev

Validate:

1. If a Permission Set with the *Registry* permission of *Grant Assemblies unrestricted access to the registry (unrestricted=”true”)* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Registry Permission	
Reference:	.NET Framework Security Guide pg	IA Control: DCSL-1

5.29 APPNET0030: Directory Services Permission

Description: The Directory Services permission controls application access to the system Directory Service resources.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to review the non-default permission sets with the permission System.DirectoryServices.DirectoryServicesPermission.

Example:
7. Dev =
<PermissionSet class="System.Security.NamedPermissionSet" version="1" Name="Dev">
<IPermission class="**System.DirectoryServices.DirectoryServicesPermission**,
System.DirectoryServices, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a" version="1" />
</PermissionSet>

Use caspol to list the non-default code groups and their corresponding permission sets to determine which non default code groups use the permission sets in the previous step.

Example:
Level = Machine

Code Groups:

1. All code: Nothing
1.1. Zone - MyComputer: FullTrust
1.1.1. StrongName -
00240000048000009400000006020000002400005253413100040000010001000 7D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DC

C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E82
1C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0
FC4963D261C8A12436518206DC093344D5AD293: FullTrust

1.1.2. StrongName - 00000000000000000400000000000000: FullTrust

1.2. Zone - Intranet: LocalIntranet

1.2.1. All code: Same site Web

1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'

1.3. Zone - Internet: Internet

1.3.1. All code: Same site Web

1.4. Zone - Untrusted: Nothing

1.5. Zone - Trusted: Internet

1.5.1. All code: Same site Web

1.6. ApplicationDirectory: Dev

Validate:

1. If the *Directory Services* permission of *Grant assemblies unrestricted access to all directory service paths (unrestricted="true")* is assigned to a non-default Code Group that does not use a Strong Name as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.
2. If the *Directory Services* permission of *Write* () or Browse() is assigned to a non- default Code Groups that does not use a Strong Name as the membership condition and whose assignment criteria has not been reviewed and approved by the ISSO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Directory Services Permission	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

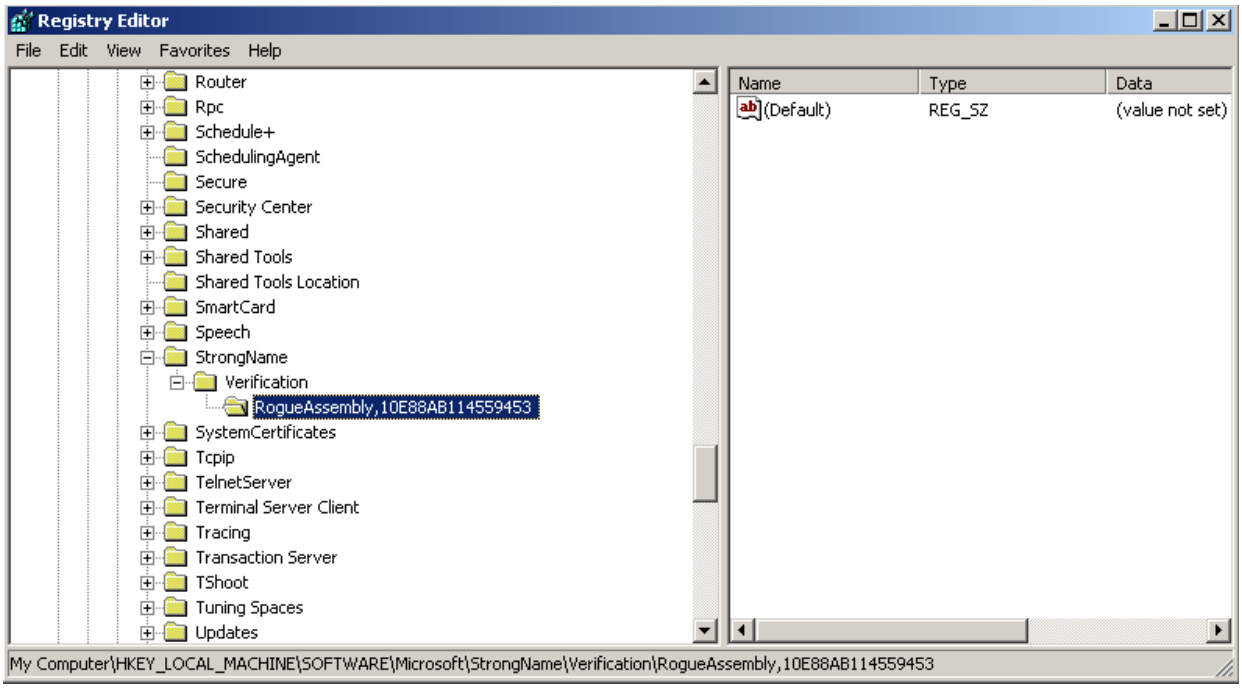
5.30 APPNET0031: No Strong Name Verification

Description: The Strong Name Membership Condition establishes the requirement for all code defined in the group to be configured with a Strong Name. Strong Name verification should not be omitted in a production environment.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use regedit to review the windows registry for assemblies omitting strong name verification in the registry entry HKLM\Software\Microsoft\StrongName\Verification. There should be no assemblies list under this registry key.

Figure 5-1: Sample Finding Example



Validate:

1. If any assemblies are listed as omitting Strong Name verification in a production environment then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	No Strong Name Verification	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

5.31 APPNET0032: First Match Code Groups

Description: The First Match Code Group is used to control the depth to which a branch of the code group tree is traversed when assigning membership to assemblies.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to list the code groups and for the First Match Code Group type.

Example:
Microsoft (R) .NET Framework CasPol 2.0.50727.42
Copyright (c) Microsoft Corporation. All rights reserved.

Security is ON
Execution checking is ON
Policy change prompt is ON

Level = Machine

Code Groups:

1. All code: Nothing
  1.1. Zone - MyComputer: FullTrust (LevelFinal)
    1.1.1. StrongName - 00240000048000009400000006020000002400005253413100040000010001000 7D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DCC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
    1.1.2. StrongName - 00000000000000000400000000000000: FullTrust
  1.2. Zone - Intranet: LocalIntranet
    1.2.1. All code: Same site Web
    1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
  1.3. Zone - Internet: Internet
    1.3.1. All code: Same site Web
  1.4. Zone - Untrusted: Nothing
  1.5. (First Match) Zone - Trusted: Internet
    1.5.1. All code: Same site Web
  1.6. Publisher - 30818902818100E47B359ACC061D70C237B572FA276C9854CFABD469DFB74E77D026630BEE2A0C2F8170A823AE69FDEB65704D7FD446DEFEF1F6BA12B6ACBDB1BFA7B9B595AB9A40636467CFF7C73F198B53A9A7CF177F6E7896EBC591DD3003C5992A266C0AD9FBEE4E2A056BE7F7ED154D806F7965F83B0AED616C192C6416CFCB46FC2F5CFD0203010001: FullTrust

Success

Validate:

1. If First Match Code Groups are used and the site does not have documentation regarding their use of First Match Code Groups then this is a finding.
2. Ask the System Administrator to verify the CAS policy is loaded is not the default policy. The CLR will load the default CAS policy when the policy file is corrupted. If the security policy is not loaded this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	First Match Code Groups	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

5.32 APPNET0033: File Code Groups, Net Code Groups

Description: The File Code Groups and Net Code Groups are used to establish directory access and web site connections respectively by the application

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Check for the CAS configuration files for any non-default File or Net Code groups. Verify .Net Configuration Files for Enterprise, Machine and User configurations for all installed versions of the .Net Framework. Search for any entries in the configuration file where the code group class is System.Security.Policy.NetCodeGroup or System.Security.Policy.FileCodeGroup

See Section 4.6 of the checklist for .Net Configuration File Locations

Examples:
<CodeGroup class="System.Security.Policy.NetCodeGroup
<CodeGroup class="System.Security.Policy.FileCodeGroup

Validate:

1. If File or Net Code Groups are used and the site does not have documentation regarding their use then this is a finding.
2. Ask the System Administrator to verify the CAS policy is loaded is not the default policy. The CLR will load the default CAS policy when the policy file is corrupted. If the security policy is not loaded this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	File Code Groups, Net Code groups	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

5.33 APPNET0035: Level Final Code Group Attribute

Description: The Level Final Code Group Attribute prevents permission sets farther down in the Code Group hierarchy from being applied to the assembly.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to list the code groups and and the Level Final code group attribute.

Example:
Microsoft (R) .NET Framework CasPol 2.0.50727.42
Copyright (c) Microsoft Corporation. All rights reserved.

Security is ON
Execution checking is ON Policy change prompt is ON

Level = Machine

Code Groups:

1. All code: Nothing
  1.1. Zone - MyComputer: FullTrust **(LevelFinal)**
    1.1.1. StrongName - 00240000048000009400000006020000002400005253413100040000010001000 7D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DCC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293: FullTrust
    1.1.2. StrongName - 00000000000000000400000000000000: FullTrust
  1.2. Zone - Intranet: LocalIntranet
    1.2.1. All code: Same site Web
    1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
  1.3. Zone - Internet: Internet
    1.3.1. All code: Same site Web
  1.4. Zone - Untrusted: Nothing
  1.5. (First Match) Zone - Trusted: Internet
    1.5.1. All code: Same site Web
  1.6. Publisher - 30818902818100E47B359ACC061D70C237B572FA276C9854CFABD469DFB74E77D026630BEE2A0C2F8170A823AE69FDEB65704D7FD446DEFEF1F6BA12B6ACBDB1BFA7B9B595AB9A40636467CFF7C73F198B53A9A7CF177F6E7896EBC591DD3003C5992A266C0AD9FBEE4E2A056BE7F7ED154D806F7965F83B0AED616C192C6416CFCB46FC2F5CFD0203010001: FullTrust

Success

Validate:

1. If Level Final Code Groups are used and the site does not have documentation regarding their use then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Level Final Code Group Attribute	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

5.34 APPNET0041: Zone Membership Condition

Description: The Zone Membership Condition determines policy level based on the URL zone of the application origin.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to list the code groups and the zone membership condition.

Example:
Microsoft (R) .NET Framework CasPol 2.0.50727.42
Copyright (c) Microsoft Corporation. All rights reserved.

Security is ON
Execution checking is ON
Policy change prompt is ON

Level = Machine

Code Groups:

1. All code: Nothing
  1.1. **Zone** - MyComputer: FullTrust (LevelFinal)
    1.1.1. StrongName -
00240000048000009400000006020000002400005253413100040000010001000 7D1FA57C4A
ED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DC
C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E82
1C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0
FC4963D261C8A12436518206DC093344D5AD293: FullTrust
    1.1.2. StrongName - 0000000000000000040000000000000: FullTrust
  1.2. **Zone** - Intranet: LocalIntranet
    1.2.1. All code: Same site Web
    1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
  1.3. **Zone** - Internet: Internet
    1.3.1. All code: Same site Web
  1.4. **Zone** - Untrusted: Nothing
  1.5. (First Match) Zone - Trusted: Internet
    1.5.1. All code: Same site Web
  1.6. Publisher -
30818902818100E47B359ACC061D70C237B572FA276C9854CFABD469DFB74E77D026630
BEE2A0C2F8170A823AE69FDEB65704D7FD446DEFEF1F6BA12B6ACBDB1BFA7B9B595
AB9A40636467CFF7C73F198B53A9A7CF177F6E7896EBC591DD3003C5992A266C0AD9F
BEE4E2A056BE7F7ED154D806F7965F83B0AED616C192C6416CFCB46FC2F5CFD020301
0001: FullTrust

Success

Validate:
1. If a *Zone* membership condition is used for a non-default code group this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Zone Membership Condition	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

5.35 APPNET0045: Administering CAS Policy

Description: The use of the CAS policy can be enabled or disabled on the system.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to list the code groups for the enterprise policy of Search for the occurrence of Security is OFF.

Example:
C:\WINDOWS\Microsoft.NET\Framework\v1.1.43\caspol -en -lg Microsoft (R) .NET Framework CasPol 1.1.4322.573
Copyright (C) Microsoft Corporation 1998-2002. All rights reserved.

Security is OFF
Execution checking is ON
Policy change prompt is ON

Level = Enterprise

Code Groups:

1. All code: FullTrust

Success

Validate:

1. If CAS Policy has been disabled then this is a finding.

Category:	CAT I	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Administering CAS Policy	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

5.36 APPNET0046: Software Publishing Certificate

Description: The Windows system may be configured to allow use of certificates. These certificates must be validated correctly.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use regedit as in Section 4.8 of the checklist to determine the software publishing state value for all users of the system.

Validate:

1. If the *Software Publishing State* is not set to hexadecimal *23C00*, then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Software Publishing Certificate	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

5.37 APPNET0048: Publisher Membership Condition

Description: The Publisher Member Condition requires member code to be certified using certificates originating from a trusted source.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5
Use caspol to list the code groups and the publisher membership condition.

Example:
Microsoft (R) .NET Framework CasPol 2.0.50727.42
Copyright (c) Microsoft Corporation. All rights reserved.

Security is ON
Execution checking is ON
Policy change prompt is ON

Level = Machine

Code Groups:

1. All code: Nothing
  1.1. Zone - MyComputer: FullTrust (LevelFinal)
    1.1.1. StrongName -
00240000048000009400000006020000002400005253413100040000010001000 7D1FA57C4A
ED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DC
C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E82
1C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0
FC4963D261C8A12436518206DC093344D5AD293: FullTrust
    1.1.2. StrongName - 00000000000000000400000000000000: FullTrust
  1.2. Zone - Intranet: LocalIntranet
    1.2.1. All code: Same site Web
    1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
  1.3. Zone - Internet: Internet
    1.3.1. All code: Same site Web
  1.4. Zone - Untrusted: Nothing
  1.5. (First Match) Zone - Trusted: Internet
    1.5.1. All code: Same site Web
  1.6. Publisher -

30818902818100E47B359ACC061D70C237B572FA276C9854CFABD469DFB74E77D026630
BEE2A0C2F8170A823AE69FDEB65704D7FD446DEFEF1F6BA12B6ACBDB1BFA7B9B595
AB9A40636467CFF7C73F198B53A9A7CF177F6E7896EBC591DD3003C5992A266C0AD9F
BEE4E2A056BE7F7ED154D806F7965F83B0AED616C192C6416CFCB46FC2F5CFD020301
0001: FullTrust

Success

Validate:

1. If the *Publisher Membership Condition* is used on a Non-default Code Group and the use of that Publishers certificate is not documented and approved by the ISSO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Publisher Membership Condition	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

5.38 APPNET0052: Strong Name Membership Condition

Description: The Strong Name Membership condition requires that member assemblies be defined with Strong Names.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Use caspol to list the code groups and the strong name membership condition.

Example:
Microsoft (R) .NET Framework CasPol 2.0.50727.42
Copyright (c) Microsoft Corporation. All rights reserved.

Security is ON
Execution checking is ON
Policy change prompt is ON

Level = Machine

Code Groups:

1. All code: Nothing
  1.1. Zone - MyComputer: FullTrust (LevelFinal)
    1.1.1. **StrongName** -
00240000048000009400000006020000002400005253413100040000010001 0007D1FA57C4A
ED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DC
C1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E82
1C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0

FC4963D261C8A12436518206DC093344D5AD293: FullTrust
1.1.2. **StrongName** - 00000000000000000400000000000000: FullTrust
1.2. Zone - Intranet: LocalIntranet
1.2.1. All code: Same site Web
1.2.2. All code: Same directory FileIO - 'Read, PathDiscovery'
1.3. Zone - Internet: Internet
1.3.1. All code: Same site Web
1.4. Zone - Untrusted: Nothing
1.5. (First Match) Zone - Trusted: Internet
1.5.1. All code: Same site Web
1.6. Publisher -
30818902818100E47B359ACC061D70C237B572FA276C9854CFABD469DFB74E77D026630BEE2A0C2F8170A823AE69FDEB65704D7FD446DEFEF1F6BA12B6ACBDB1BFA7B9B595AB9A40636467CFF7C73F198B53A9A7CF177F6E7896EBC591DD3003C5992A266C0AD9FBEE4E2A056BE7F7ED154D806F7965F83B0AED616C192C6416CFCB46FC2F5CFD0203010001: FullTrust
Success

Validate:

1. If a Strong Name membership condition is assigned to a non-default Code Group ensure the private key is adequately protected by the software developer. Ask the System Administrator how the private keys are protected. If the private key is not adequately protected then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Strong Name Membership Condition	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

5.39 APPNET0054: Administering CAS Policy for Group Names

Description: The use of duplicate code group names within a level of the CAS policy can lead to mis-assignment of permissions.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Review the caspol.exe listing for all code groups. Review all code group names. The code group names follow a sequential number at the far left of the file or screen. Code attributes for the code group are numbered and indented below each code group name.

Validate:

1. If non-unique Code Group names are used within a CAS security policy level (Enterprise, Machine, User) then this is a finding.

Category:	CAT III	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Administering CAS Policy for Group Names	
Reference:	.NET Framework Security Guide	IA Control: DCBP-1

5.40 APPNET0055: Administering CAS Policy and Policy Configuration File Backups

Description: CAS Policy and CAS Policy Configuration files are required for a complete system baseline and disaster recovery event.

Applies to Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Validate:

1. Ask the System Administrator if all CAS policy and policy configuration files are included in the system backup. If they are not then this is a finding.
2. Ask the System Administrator if the policy and configuration files are backed up prior to migration, deployment, and reconfiguration. If they are not then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Administering CAS Policy and Policy Configuration File Backups	
Reference:	.NET Framework Security Guide	IA Control: CODB-1, CODB-2

5.41 APPNET0060: Remoting Services Authentication and Encryption

Description: The *typefilterlevel="Full"* attribute allows unfiltered code to access system resources.

Applies to: Versions 1.0, 1.1, 2.0. 3.0 & 3.5

Check for the CAS configuration files for *typefilterlevel="Full"* This allows references to custom client objects to be passed as parameters.

Verify .Net Configuration Files for Enterprise, Machine and User configurations for all installed versions of the .Net Framework. Search for any entries in the configuration file where typeFilterLevel="Full" Ask the System Administrator what encryption and authentication methods are in place for the remoting channels.

See Section 4.6 of the checklist for .Net Configuration File Locations

Example:
<application name="remoteserver">
<service>
<activated type="sample.my.object, myobjects"/>
</service>
<channels>

<channel ref=“http server” port=“80”/>
</channels>
</application>

<serverProviders>
<provider ref="wsdl" />
<formatter ref="soap" typeFilterLevel="Full" />
<formatter ref="binary" typeFilterLevel="Full" />
</serverProviders>

Validate:

1. In Version 1.0 of the .NET Framework if authentication and encryption are not used for all remoting channels then this is a finding.
2. In Version 1.1 or above of the .NET Framework if authentication and encryption are not used for all remoting channels when the *typefilterlevel=”Full”, then this is a finding.*

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Remoting Services Authentication and Encryption	
Reference:	.NET Framework Security Guide	IA Control: DCSL-1

5.42 APPNET0061: Unsupported .Net Framework Versions

Description: Verify the installed .Net Frameworks are still supported by Microsoft.

Applies to: All Versions

Determine which versions of the .NET Framework are installed by opening the directory %systemroot%\Microsoft.NET\Framework. The following folders contain the released versions of the .NET Framework:
v3.5 v3.0 v2.0.50727 v1.1.4322 v1.0.3705

Search for all the Mscorlib.dll files in %systemroot%\Microsoft.NET\Framework.. Click on each of the files and view properties and click Version tab to determine the version installed. If there is no Mscorlib.dll, there is no installed version of .Net Framework in that directory.

More specific information on determining versions of .Net Framework installed can be found at the following link. http://support.microsoft.com/kb/318785

Verify that extended support is available for the installed versions of .Net Framework. Verify the .Net Framework support dates with Microsoft Product Lifecycle Search link.
http://support.microsoft.com/lifecycle/search/?sort=PN&alpha=.NET+Framework

.NET Framework 1.0	7/14/2009
.NET Framework 1.1	10/8/2013
.NET Framework 2.0	4/12/2016

.NET Framework 3.0 4/11/2017
.NET Framework 3.5 4/10/2018

Validate:

1. If the any versions of the.Net Framework are installed and extended support is no longer available this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-
SDID :	Unsupported .Net Framework Versions	
Reference:	DoDI 8500.2 IA Controls	IA Control: COMS-1 COMS-2