

[AppSecInc Team SHATTER Security Advisory] Microsoft Windows Improper Token Validation

Source: <http://www.derkeiler.com/Mailing-Lists/securityfocus/bugtraq/2005-01/0098.html>

From: Team SHATTER (Application Security, Inc.) (*vrathod_at_appsecinc.com*)

Date: 01/10/05

Date: Mon, 10 Jan 2005 17:12:17 -0500

To: bugtraq@securityfocus.com, full-disclosure@lists.netsys.com, vulnwatch@vulnwatch.org, ntbugtr

Microsoft Windows Improper Token Validation

AppSecInc Team SHATTER Security Advisory

<http://www.appsecinc.com/resources/alerts/general/06-0001.html>

January 10, 2005

Credit: This vulnerability was discovered and researched by Cesar Cerrudo of Application Security, Inc.

Risk Level: High

Summary:

A local privilege elevation vulnerability exists on the Windows operating systems. This vulnerability allows any user to take complete control over the system and affects Windows 2000, Windows XP, and Windows 2003 (all service packs).

Versions Affected:

Microsoft Windows 2000, Windows XP, and Windows 2003 (all service packs).

Details:

According to MSDN:

"An access token is an object that describes the security context of a process or thread. The information in a token includes the identity and privileges of the user account associated with the process or thread. When a user logs on, the system verifies the user's password by comparing it with information stored in a security database. If the password is authenticated, the system produces an access token. Every process executed on behalf of this user has a copy of this access token.

The system uses an access token to identify the user when a thread interacts with a securable object or tries to perform a system task that requires privileges. Access tokens contain the following information:

- The security identifier (SID) for the user's account
- SIDs for the groups of which the user is a member
- A logon SID that identifies the current logon session
- A list of the privileges held by either the user or the user's groups
- An owner SID
- The SID for the primary group
- The default DACL that the system uses when the user creates a securable object without specifying a security descriptor
- The source of the access token
- Whether the token is a primary or impersonation token
- An optional list of restricting SIDs
- Current impersonation levels
- Other statistics

Every process has a primary token that describes the security context of the user account associated with the process. By default, the system uses the primary token when a thread of the process interacts with a securable object. Moreover, a thread can impersonate a client account. Impersonation allows the thread to interact with securable objects using the client's security context. A thread that is impersonating a client has both a primary token and an impersonation token."

Microsoft introduced a new user right called "Impersonate a client after authentication" in Windows 2000 SP4, Windows 2003, and Windows XP SP2. This right allows or limits the processes ran by a user from being able to impersonate. For instance, if a process thread running in the security context of a user without proper rights tries to impersonate, then it gets an Identity Token instead of an Impersonation Token. An Identity Token only identifies the user account under which the target process is running and can not be used for impersonation. An Identity Token can also be retrieved by a thread in order to identify the user account under which a process is running. Under certain circumstances this Identity Token can be used to impersonate any process thread running under any user account.

The attack vector identified is to impersonate a victim using Identity Tokens to access network shares using UNC. For instance, after a thread gets an Identity Token for the Local System account or an administrative account, the token can be used to impersonate and access administrative shares such as `\\computername\c$` and to replace system files such as `.exe`, `.dll`, etc... This allows an attacker to elevate privileges or to read arbitrary files bypassing permissions. Also, network shares on other computers can be accessed in the same way. For instance, user JohnDoe's Identity Token can access `\\remotepc\someshare\` for which the user JohnDoe has permissions but the attacker does not. The attack succeeds because apparently that user's credentials are cached by the LSASS (Local Security Authority Subsystem Service) after successfully authenticating to a network share by standard methods. Then when the share is accessed again, the LSASS assumes an Identity Token is an Impersonation token and uses the cached credentials to authenticate.

This vulnerability is critical for servers using Terminal Services (or Citrix) because a user could impersonate any other user to access network shares.

Links:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/secauthz/security/client_impersonation.asp
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/secauthz/security/access_tokens.asp
<http://support.microsoft.com/kb/821546/en-us>
<http://www.microsoft.com/resources/documentation/WindowsServ/2003/standard/proddocs/en-us/Default.asp?url=/res>

Workaround:

None.

Fix:

<http://www.microsoft.com/technet/security/bulletin/MS04-044.msp>

Application Security, Inc.
www.appsecinc.com

AppSecInc is the leading provider of database security solutions for the enterprise. AppSecInc products proactively secure enterprise applications at more than 200 organizations around the world by discovering, assessing, and protecting the database against rapidly changing security threats. By securing data at its source, we enable organizations to more confidently extend their business with customers, partners and suppliers. Our security experts, combined with our strong support team, deliver up-to-date application safeguards that minimize risk and eliminate its impact on business.
