

[NT] Trusting Domains Do Not Verify Domain Membership of SIDs in Authorization Data

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2002-01/0137.html>

From: support@securiteam.com

Date: 01/31/02

From: support@securiteam.com

To: list@securiteam.com

Date: Thu, 31 Jan 2002 23:09:38 +0100 (CET)

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

When was the last time you checked your server's security?

How about a monthly report?

<http://www.AutomatedScanning.com> -- Know that you're safe.

Trusting Domains Do Not Verify Domain Membership of SIDs in Authorization Data

SUMMARY

Trust relationships are created between Windows NT or Windows 2000 domains to allow users in one domain to access resources in other domains without requiring them to authenticate separately to each domain. When a user in a trusted domain requests access to a resource in a trusting domain, the trusted domain supplies authorization data in the form of a list of Security Identifiers (SIDs) that indicate the user's identity and group memberships. The trusting domain uses this data to determine whether to grant the user's request.

A vulnerability exists because the trusting domain does not verify that the trusted domain is actually authoritative for all the SIDs in the authorization data. If one of the SIDs in the list identified a user or security group that is not in the trusted domain, the trusting domain would accept the information and use it for subsequent access control decisions. If an attacker inserted SIDs of his choice into the authorization data at the trusted domain, he could elevate his privileges to those associated with any desired user or group, including the Domain Administrators group for the trusting domain. This would enable the attacker to gain full Domain Administrator access on computers in the

Securiteam: [NT] Trusting Domains Do Not Verify Domain Membership of SIDs in Authorization Data trusting domain.

Exploiting this vulnerability would be difficult, and require administrative privileges on the trusted domain, as well as the technical wherewithal to modify low-level operating system functions and data structures.

- Windows NT 4.0 provides no mechanism by which additional SIDs could be added to authorization data. To exploit the vulnerability, an attacker would need to develop and install custom operating system components to add the SIDs.

- Windows 2000 does provide a mechanism for introducing additional SIDs into authorization data, known as SIDHistory. However, there is no programming interface that would allow an attacker – even with administrative rights – to introduce a desired SID into the SIDHistory information; instead, an attacker would need to perform a binary edit of the data structures that hold the SIDHistory information.

Microsoft has developed a mechanism called SID Filtering that eliminates the vulnerability and adds further protection between trusting domains. When installed and enabled on the domain controllers of a trusting domain, SID Filtering causes the system to inspect all incoming authorization data and remove any SIDs that do not identify a user or security group that is defined in the trusted domain.

There are, however, tradeoffs associated with using the SID Filtering mechanism. These are summarized in the FAQ and Caveats sections below, and are discussed in detail in Microsoft Knowledge Base article <<http://www.microsoft.com/technet/support/kb.asp?ID=289243>> Q289243 and in a technical white paper (<<http://www.microsoft.com/windows2000/techinfo/administration/security/sidfilter.asp>> <http://www.microsoft.com/windows2000/techinfo/administration/security/sidfilter.asp>) that Microsoft strongly urges administrators to read before using SID Filtering. This is especially important in the case of administrators who are in the midst of migrating their networks from Windows NT 4.0 to Windows 2000.

DETAILS

Affected Software:

- * Microsoft Windows NT 4.0
- * Microsoft Windows 2000

Mitigating factors:

- * The attacker would need to have domain administrator privileges in the trusted domain in order to exploit the vulnerability.
- * The attacker's domain would need to already be trusted by the target domain, or the target domain's administrator would need to approve the establishment of a new trust relationship. There is no capability for the attacker to unilaterally initiate a trust relationship with another domain or cause it to trust the attacker's domain.
- * The attacker would need to modify operating system components and data.

Securiteam: [NT] Trusting Domains Do Not Verify Domain Membership of SIDs in Authorization Data

Patch availability:

Download locations for this patch

* Windows NT 4.0 Server and Windows NT 4.0 Server, Enterprise Edition:

Included in the

<<http://www.microsoft.com/downloads/release.asp?ReleaseID=31240>> Windows NT 4.0 Security Roll-up Package.

* Windows 2000 Server and Advanced Server:

The fix for this issue is included in

<<http://www.microsoft.com/windows2000/downloads/critical/q311401/default.asp>> Windows 2000 Security Roll-up Package 1

* Microsoft Windows 2000 Datacenter Server:

Patches for Windows 2000 Datacenter Server are hardware-specific and available from the original equipment manufacturer.

What is the scope of this vulnerability?

This is a privilege elevation attack. If an attacker had domain administrator privileges on a domain, and the domain were trusted by another domain, it would be possible for the attacker to gain privileges on the trusting domain, up to and including administrator privileges.

Exploiting this vulnerability would be difficult:

* The attacker would need to already have complete administrative control over the trusted domain.

* The vulnerability could only be exploited if there was a pre-existing trust relationship between the attacker's domain and the other one. The attacker would not be able to establish one by himself.

* On both Windows NT 4.0 and Windows 2000 systems, the attacker would need to possess the technical skills to modify low-level operating system functions and data.

Nevertheless, the worst case associated with this vulnerability is serious, and we encourage all Windows NT 4.0 and Windows 2000 administrators to consider deploying the fix if physical and personnel security considerations indicate sufficient risk.

What causes the vulnerability?

The vulnerability results because, when a trust relationship exists between two domains, the trusting domain will accept the SIDs specified within authorization data provided by the trusted domain – even if the SIDs are from domains other than the trusted one. If an attacker in a trusted domain were able to insert SIDs of his choice into authorization data, he could grant himself the privileges associated with a user in another domain, including the trusting domain itself.

What is a trust relationship?

Trust relationships allow users in one Windows NT or Windows 2000 domain to be authenticated and allowed access to resources in another domain.

There are two parties in a trust relationship:

* The trusting domain, also known as the resource domain. This is the domain where resources like file servers are located.

* The trusted domain, also known as the account domain. This is the domain

Securiteam: [NT] Trusting Domains Do Not Verify Domain Membership of SIDs in Authorization Data

where user accounts and security groups are defined.

The account domain authenticates users when they logon and connect to network servers, and provides identity information to the resource domain, which trusts the account domain to do this correctly. For this reason, the resource domain is usually referred to as the trusting domain, and the account domain is usually referred to as the trusted domain.

What is a SID?

A SID is a Security Identifier – a unique identifying value that is associated with a user account or group in Windows NT or Windows 2000. SIDs are used extensively in the Windows NT and Windows 2000 security architecture to support authentication and implement access control.

How do SIDs determine access to resources?

In Windows NT and Windows 2000, the owner of a resource regulates access to it using Access Control Lists (ACLs). An ACL consists of a table of Access Control Entries (ACEs), each of which specifies the SID for one user account or group, and the permissions that it has on the resource.

Whenever a user requests access to a protected resource, his account domain receives a request to authenticate the user. During this process, the account domain generates authorization data (listing the SIDs for the user's account and all the groups he belongs to) and sends it as part of the response. When the machine that hosts the resource receives the response, it uses the authorization data to build an access token. It then compares the SIDs in the token to the SIDs in the ACL and determines the privileges the user is entitled to.

To give an example, let us assume the following scenario:

- * A trust relationship exists between two domains, called Domain A and Domain B.
- * Domain B trusts Domain A.
- * Joe has a user account in Domain A, and is member of the Sales group there.
- * Joe logs into the system with his Domain A\Joe account.
- * Joe wants to access a file hosted on a computer in Domain B. The share and file name is \\ServerB\projects\plans.txt.
- * The owner of plans.txt has specified an ACL that grants full control to members of the Administrators group, read access to members of the Sales group in Domain A, and write access to user Joe in Domain A.

When Joe tries to open the file, one of the domain controllers in Domain A is contacted to authenticate Joe. While processing the authentication request, it generates the authorization data that includes the SIDs for Domain A \Joe and for Domain A \ Sales, because Joe is a member of the Sales group. When ServerB receives this authorization data, it adds SIDs for any local machine groups that Joe is a member of, and builds an access token. ServerB uses this token to impersonate Joe and open the file. The file system compares the SIDs in the token to those in the ACL for the file. The file system determines that Joe has read permission by virtue of

Securiteam: [NT] Trusting Domains Do Not Verify Domain Membership of SIDs in Authorization Data

his membership in the Domain A \Sales group, and write permission by virtue of being Domain A \Joe, and therefore grants him both read and write permission to the file.

What is wrong with the way authorization data is accepted?

There is a flaw in the way a trusting Windows NT or Windows 2000 domain accepts the authorization data from a trusted domain. Specifically, the trusting domain accepts the authorization data as correct, even if some of SIDs in the list are from domains for which the trusted domain is not authoritative.

But you said above that the trusting domain always accepts the trusted domain's authentication. This sounds like it is in keeping with that statement.

The problem is that the trusting domain is, in this case, too trusting. By design, it should trust the trusted domain's authentication – but only for accounts, the trusted domain is authoritative over. The trusted domain should not be able to make assertions about accounts that reside in other domains. This vulnerability could enable the trusted domain to make assertions about users and groups that are in some other domain.

Just to be clear, does the vulnerability lie in the trusted domain or in the trusting one?

The vulnerability lies in the mechanism by which the trusting domain uses authorization data. It is the trusting domain's responsibility to ensure that the trusted domain does not exceed the authority that the trusting domain granted it. Therefore, the trusting domain must check all the authorization information provided by the trusted domain. There is no vulnerability in the trusted domain.

What would this vulnerability enable an attacker to do?

If an attacker had sufficient privileges in a trusted domain, and he could modify the behavior of the security subsystem of the domain controller, he could insert SIDs of his choice into the authorization data and thereby elevate his privileges when connecting to computers in the trusting domain.

For instance, in the scenario we discussed above, suppose Joe only had read access to the file, plans.txt. If Joe were able to add a SID to the authorization data, he could identify himself as a member of a different group in a different domain – for example, Domain B/Domain Administrators. Not only would this grant him additional privileges on plans.txt, it also would give him full control of any computer in Domain B.

What privileges would the attacker need in order to modify the authorization data?

The attacker would need to have domain administrative privileges in the trusted domain.

But if the attacker has administrative privileges, all bets are off. Is not this the Sixth Immutable Law of Security?

Securiteam: [NT] Trusting Domains Do Not Verify Domain Membership of SIDs in Authorization Data

Actually, this situation goes a bit beyond the Sixth Immutable Law of Security ("A machine is only as secure as the administrator is trustworthy"). It is true that if the administrator of a machine (or, in this case, a domain) is untrustworthy, there is effectively no security on it. However, the damage should be limited to the domain that the administrator has authority over – he should not be able to extend his control to other domains. The fact that this issue would enable him to extend his control makes this a security vulnerability.

How hard would it be for an attacker to modify the authorization data? It would be extremely difficult. As we discussed above, the vulnerability means that if an attacker could introduce SIDs into the authorization data from a trusted domain, the trusting domain would accept them. However, nothing in the vulnerability provides a way for the attacker to do this.

Windows NT 4.0 lacks any mechanism for adding SIDs to authorization data. An attacker could only exploit this vulnerability if he first implemented such a mechanism, then installed it onto the system. This would require that the attacker have the technical skills to write low-level operating system components, and the administrative privileges needed to install the components onto a domain controller.

Windows 2000 does provide a mechanism for adding SIDs to authorization data, called SIDHistory. However, this is not as easily exploited as it might initially appear to be, because there are no functions by which the attacker – even with administrative privileges – could arbitrarily manipulate the SIDHistory data. To exploit the vulnerability, the attacker would need either to implement such functions, or manipulate data within low-level operating system data structures while the system was running.

What is SIDHistory and what does it do?

SIDHistory is mechanism that was introduced in Windows 2000 to aid in migrating user accounts from Windows NT 4.0 domains to Active Directory. Typically, when a Windows NT 4.0 network migrates to Windows 2000, the domain structure is revised to take advantage of the new efficiencies Windows 2000 offers. The number of domains is usually reduced, and user accounts are migrated into new, larger domains.

When a user account is migrated to a new domain, the account is assigned a new SID, but all of the ACLs on existing resources continue to refer to the account's old SID. As a result, a user whose account was migrated would lose access to all resources on the network. For instance, suppose Joe's domain, Domain A, was migrated to NewDomain. NewDomain\Joe would have a different SID than Domain A\Joe did, and NewDomain\Joe would not be able to access the resources he could as Domain A\Joe.

The SIDHistory mechanism retains the SID from the user's previous domain within the Active Directory, and includes it in any authorization data the domain generates. Therefore, if NewDomain used the SIDHistory mechanism and NewDomain\Joe requested access to \\ServerB\projects\plans.txt, NewDomain would generate authorization data containing not only the SIDs

Securiteam: [NT] Trusting Domains Do Not Verify Domain Membership of SIDs in Authorization Data

for NewDomain\Joe and any groups he belonged to there, but also the SIDs for Domain A\Joe.

Is the SIDHistory feature a security risk?

No. There is not a flaw in SIDHistory. It is simply a mechanism that could be misused, if the attacker had significant technical skills and complete administrative control over the machine.

Could the attacker simply add new SIDHistory data for himself?

No. There are neither system calls nor LDAP APIs that enable SIDHistory data to be modified. Instead, the attacker would need to either implement his own system calls, or use a debugger to modify the contents of the data structures that hold the SIDHistory data.

Could an attacker set up a rogue domain on a network and grant himself privileges on other domains in the network?

Simply setting up a machine in its own domain would not give the attacker any kind of trust in other domains. This does, however, reinforce the need for domain administrators to be very careful when establishing trust relationships with other domains, and only grant trust to domains that truly are trustworthy.

What is the solution to the vulnerability?

Microsoft has developed a mechanism called SID Filtering to eliminate the vulnerability. When SID Filtering is installed on the domain controllers in a trusting domain, and enabled for a specific trusted domain, it has the effect of establishing "quarantine" on the trusted domain. Thereafter, the trusting domain checks all incoming authorization data from that trusted domain and removes any SIDs that don't belong to it.

For instance, suppose SID Filtering were installed and enabled on the domain controllers in Domain B from the previous example. If an access token were received from Domain A that contained the SIDs for Domain A\Joe, Domain A\Sales and Domain B\Domain Administrators, SID Filtering would remove the SID for Domain B\Domain Administrators, as Domain A is not authoritative for users or groups in Domain B.

Once the patch is installed, is SID Filtering enabled by default?

No. As we will discuss in more detail below, SID Filtering should only be enabled on particular domain controllers, and even then only after careful consideration of how it will affect your network. Microsoft Knowledge Base article Q289243 provides details for enabling and configuring SID Filtering. At a high level, though, in Windows NT 4.0, SID Filtering is enabled and configured using a registry entry (HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Netlogon\Parameters\QuarantinedDomains) and in Windows 2000, it has enabled and configured using the NetDom command.

Should I enable SID Filtering on all machines in my network?

No. To protect a domain, you only need to enable SID Filtering on the domain controllers. Member servers and workstations in the domain do not use or implement SID Filtering. You do, however, need to ensure you have

Securiteam: [NT] Trusting Domains Do Not Verify Domain Membership of SIDs in Authorization Data

enabled SID Filtering on all domain controllers in the domains you want to protect. If you miss one domain controller server, it might be possible for an attacker to exploit the vulnerability via that computer.

Can I apply SID Filtering to domains within the same forest in a Windows 2000 network?

No. SID Filtering should only be applied to external trusts -- that is, trust relationships between domains that are not in the same forest. It should not be applied to trust relationships within a forest, as doing so will block replication and other functions that are essential to the proper operation of forest. If a domain is sufficiently untrustworthy to warrant applying SID Filtering to it, it really should not be a member of the forest.

Are there any drawbacks associated with using SID Filtering?

Yes. SID Filtering operates by simply removing all SIDs that do not belong to the domain sending them. This does effectively screen out all falsified SIDs, but it will also screen out legitimate SIDs that simply does not come from the originating domain. Two cases in which this can interfere with legitimate operation are:

- * Network migrations that use SIDHistory to mitigate the effects of migration until resources can be re-ACLED. SID Filtering will prevent the SIDHistory mechanism from working in quarantined domains.

- * Universal groups managed outside users' account domains.

Is there any other information I should consult before deploying SID Filtering?

There are two particularly useful references that we recommend reading:

*

<<http://www.microsoft.com/windows2000/techinfo/planning/activedirectory/addeladmin.asp>> Design Considerations for Delegation of Administration in Active Directory, which provides a detailed discussion of important Active Directory administration and security concepts. While this white paper does not discuss SID Filtering per se, it is very useful background reading.

*

<<http://www.microsoft.com/windows2000/techinfo/administration/security/sidfilter.asp>> Using SID Filtering to Prevent Elevation of Privilege Attacks, which provides additional technical detail about the security issue, including detailed discussions of the drawbacks discussed above and strategies for using SID Filtering most effectively. We strongly recommend that all system administrators read the white paper before deploying SID Filtering.

ADDITIONAL INFORMATION

The information has been provided by <<mailto:secnotif@MICROSOFT.COM>> Microsoft Product Security.

=====

This bulletin is sent to members of the SecuriTeam mailing list.

To unsubscribe from the list, send mail with an empty subject line and body to:

list-unsubscribe@securiteam.com

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@securiteam.com

=====
=====

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.

-
- ***Previous message:*** support@securiteam.com: "[\[UNIX\] ripMIME Mail Filter Remote Buffer Overflows](#)"
 - ***Messages sorted by:*** [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#) [\[attachment \]](#)