Global Information Assurance Certification Paper

ADVACED INCIDENT HANDLING AND HACKER EXPLOITS

GCIH PRACTICAL ASSIGNMENT
Version 1.5c
Option 2 - Document an exploit, vulnerability or malicious program

The Vulnerability of Microsoft System Management Server Remote Tools

By: Roman Kulbashny

Introduction

SMS is a powerful and widely used system management tool. It is fair to say that most organizations or companies that have Windows NT systems will also have SMS for management of its Windows NT and Windows 9x hosts.
Microsoft SMS has many power features like:

- Software and hardware inventory;
- Software metering;
- Software distribution;
- Network monitoring and
- Remote control tools.

This makes SMS a very desirable platform to take control of a network.
It is also accepted knowledge that SMS Remote control tools can be used as hacker's tools. Some comments on the matter can be found in "Don't Worry Windows Users, Everything Will BO2K" by a well known hacker group The Cult of the Dead Cow.[1]

SMS allows you to gain access to a very large number of computer systems by breaking the security of just one component.

SMS security has many levels.[2]
Unfortunately, SMS has as many vulnerabilities as it has security levels.
To break SMS security we don't need to break all levels of SMS security, one is enough.

To illustrate this we will exploit a vulnerability of the remote tools security.
The core SMS is based on the WBEM/SMS Site database/SQL database.
By getting control of SMS Site database, the attacker can gain full control of SMS Server, and from there the whole network.

By default, the SMS site server will use the SQL "sa" account for access to the SMS Site database.

The Microsoft SQL Server has many vulnerabilities that allow someone to obtain control of the SQL "sa" account: for example "SQL Server 7.0 Service Pack Password" Vulnerability[3], "the MS SQL blank "sa" password" vulnerability[4]. One vulnerability, which can be used to obtain a password for SQL "sa" login, has been described at "GCIH PRACTICAL ASSIGNMENT, SQLEXEC v 1.0 exploit", by Chuck Crawford, May 28, 2001.[5]

What makes SQL for SMS Site database easy to exploit is that it is not very well monitored and secure. Yes, Windows NT logs maybe are monitored and security policies will be put in place by network administrators, but because network administrators will be busy dealing with network security they can overlook the security of the SQL database that runs in the background of SMS infrastructure. "As with most other IS projects, configuring security, considering internal controls, and addressing for any specific department requirements are perceived to require

additional administrative overhead and are therefore often neglected-- unless IS Security or Audit is part of the SMS implementation and review team,"[6] Gary Wong writes.

After gaining access to the SMS site database, an attacker gets virtually unlimited control of the SMS server. Not only can he or she view information collected by SMS including: TCP/IP, NetBIOS, MAC addresses of SMS clients and query for information about software installed on SMS clients for exploring known vulnerabilities, the attacker can distribute malicious code to SMS clients, and get remote control for SMS clients.

Exploit/Vulnerability Details

Operating system

Server platform: Windows NT 4.0, Windows 2000
Client platform: Windows NT, Windows 2000, Windows 98, and Windows 95.

Protocol/Services

UDP ports 1761, 1762, 1763, 1764, 2701, 2702, 2703, 2704
TCP ports 1761, 1762, 1763, 1764, 2701, 2702, 2703, 2704
SQL server TCP port 1433
DCOM/RPC

Brief Description

This vulnerability allows access to the SMS Remote Tools client, bypassing Windows NT and DCOM security of SMS Server by altering the SMS Site database on Microsoft SQL Server.

Protocol Description

The SMS employs the OSI model for network communications. Because vulnerability lies in the application level of OSI we will review this level of SMS communication with more detail. The transport level of communication - in our case, TCP - will be used for its direct purpose to establish communication between the "ATTACKER" host and "VICTIM" host and can by used to detect attack based on the IP address of the attacker and port number associated with the vulnerability.

The Microsoft SMS Server uses many protocols for communication between different components of SMS, in the data flow of SMS and between computers in SMS infrastructure.[7] MS Security employs the security of different technologies that it uses:

- Windows NT security
- SMS Provider security
- Windows Management Instrumentation (WMI) security, which is Microsoft's implementation of Web-Based Enterprise Management (WBEM)
- DCOM security
- Microsoft SQL Server Security

Windows NT Security

It is commonly held that Windows NT security is the key element of SMS security. "SMS is very dependent on Windows NT and Windows NT security," according to Microsoft TechNet. [8]
SMS uses authentication, pass-through authentication and domain security of Windows NT. The Windows NT security environment has access control lists (ACL) that are associated with objects on the Windows NT operating system such as files and directories.
Windows NT user and user group are used to control access to objects of SMS security.
Windows NT rights are given to accounts, and they allow processes created for those accounts to perform specific functions on a computer. The ability to use rights that are granted to an account is stored in a token when the token or process is created.
Permissions are assigned to objects of Windows NT. They are used when the Windows NT security system must determine whether a process is allowed to access an object for the kind of access the process is requesting. Windows NT compares the user or group name and domain with the object's access control list. If a match is found, then Windows NT determines what kind of access requested is permitted. Privileges are a combination of rights and permissions granted to users or groups.
Windows NT hosts do not trust tokens from other hosts, because the other computers might not be running a valid version of the operating system. A Windows NT host that attempts to connect to another host must provide credentials that can be authenticated against a security database, in the same way that credentials are authenticated during the creation of a process or token.
When a process provides credentials for a network connection, it is common for the process to provide the credentials that were used when the process was created. It is also valid for the process to provide credentials different from the originals. This allows the process to run with one

set of privileges locally, but to connect to another host with a different set of permissions.
The ability of a process to use one account locally and another for its network connections allows greater control of security, because the process's account that specifies the privileges can be limited to a specific Windows NT host.
User accounts or user groups can be used to grant SMS administrators only the rights required for the administrators to perform tasks in the SMS Administrator console.

WMI security

The WMI is used for a communication interface between the Common Informational Model (CIM) repository and components of SMS like SMS hardware and software inventory, SMS Administration Console and SMS Site database.
WBEM security provides another level of security for SMS. To gain access to SMS security objects through the SMS Provider, users must log on to the WBEM namespace.
Each WMI namespace has its own security descriptor, which allows the namespace to have its own security permissions.
The security descriptor is used to control access to WMI services. The security descriptor is a standard Windows NT security descriptor, and it contains an ACL, a list of access control entries. Each of them grants permission to perform a specific operation, such as logons, remote access, or reading or writing the CIM Repository (the WBEM database). The WMI security descriptors are stored in the WBEM database.

DCOM security

DCOM is a protocol used for communication between WMI components. The DCOM is a distributed version of Component Object Model (COM) that provides communication between objects on the computer network. The DCOM has been designed for easy communication in distributed component environments. As result, WMI does not rely on the security of DCOM, but utilizes its own security for WMI namespaces.
DCOM utilizes the security directory provided by Windows NT. The Windows NT user directory stores the necessary information to validate a user's credentials.
Internally, DCOM stores access control lists for components. These lists indicate which users or groups of users have the right to access a component of a certain class.
When a client makes a DCOM call, DCOM obtains the client's current user name associated with the current process. Windows NT provides information that this user credential is authentic.
DCOM then passes the user name to the computer or process where the component is running.
If the client's user name is not found, DCOM rejects the call.
DCOM is built on lower levels of network protocols: remote procedure call (RPC) and named pipes protocols. The SMS does not use RPC or named pipes levels of security.

SMS Provider security

The SMS Provider enforces a security model that creates SMS security objects (for example, Packages, Collections, Advertisements, Queries, Sites, and Status Messages) and creates specific SMS security rights. SMS security objects are objects in the SMS site database that have security

rights administered through the SMS Administrator console. Users and user groups are granted specific SMS rights to SMS security objects. SMS security objects are created for the SMS site database.

Microsoft SQL server security

The SQL server provides secure access to its own objects based on logon ID credentials.
A few types of SQL Security exist:

- On SQL 7.0, Windows NT Authentication and SQL Server Authentication.
- On SQL 6.5, Integrated security, Standard security and Mixed security.

SMS uses the SQL Server account for data exchange with the SMS site database.
Access to the SMS site database is handled by SMS Provider. No direct database access to the SMS site database is used for the SMS operation.

Description of variants

This exploit is proof of concept and at this time no variants have been found.

This paper reviews the ability to gain remote assess by altering the SMS Site database on MS SQL server and utilizing the SMS Remote Tools. The same concept can be used not only by having remote access to one or to many hosts on the network, but also by utilizing the Software Distribution function of the SMS Server in order to distribute the malicious software to network hosts on an enterprise-wide scale.
By gaining access to SMS Server configuration thru the SMS Site database, the attacker can gain remote access to the Windows NT server hosting SMS Server itself; create an SMS software package containing the malicious software; and assign this package on a mandatory basis to specific or to all Windows hosts with SMS Client installed. If the attacker finds that not all network hosts have the SMS Client software installed, he or she can modify the configurations of the SMS Client Installation component of SMS server and force the installation of SMS clients on any Windows hosts on the network. This can create, overnight, a zombie network, one that will be ready to execute a Denial of Service attack. It can also create a network of hosts that can be used for distributed encryption cracking. And it can do all this while using off-the-shelf Microsoft products readily accessible worldwide, without the need to write code or indeed exert much effort or creativity.

How the exploit works

Remote Tools is normally launched and used from the SMS Administrator Console.
To prevent unauthorized Remote Tools access, SMS employs SMS Remote Tools Security, which is a subset of the SMS Administrator Console security. This exploit takes advantage of vulnerabilities in the configuration of SMS that allow access to the SMS Remote Tools client, bypassing Windows NT and DCOM security of SMS Server by altering the SMS Site database on Microsoft SQL Server.

SMS Remote Tools security

The security of SMS Remote Tools is based on the Collection level of security and a security list of Permitted Viewers.[9] The Collection level of SMS Remote Tools Security prevents the user of SMS from starting Remote Tools against the victim host as a member of the Collection of SMS clients in SMS Administrator Console. The Permitted Viewers list is used to check if the user of Remote Tools does have permissions for remote control of the host. The list of Permitted Viewers is stored on the SMS Site database at Microsoft SQL Server at the table "SiteControl" in the binary data field "BinaryData".

SQL Server security

The default database "System Administrator" login for Microsoft SQL Server with "SQL Server Authentication" or "Standard security" is "sa".
The login "sa" also will be the default SMS Site database connection login in case the Microsoft SQL Server configured with "SQL Server Authentication" or "Standard security" is used for SMS Site database.

By utilizing one of the exploits for Microsoft SQL Server, the attacker can gain the "sa" login to the SQL server and right away get access to a repository of configuration of SMS server that resides in the database usually named <SMS_"Sitecode">. After that, an attacker can modify a whole variety of settings for SMS server, in our case a database record related to the configuration of the SMS Remote Tools.

Work around of SMS Remote Tools security.

To go around SMS Administrator Console security, the exploit uses the fact that a user of Remote Tools doesn't need to be a member of a local Administrators group.

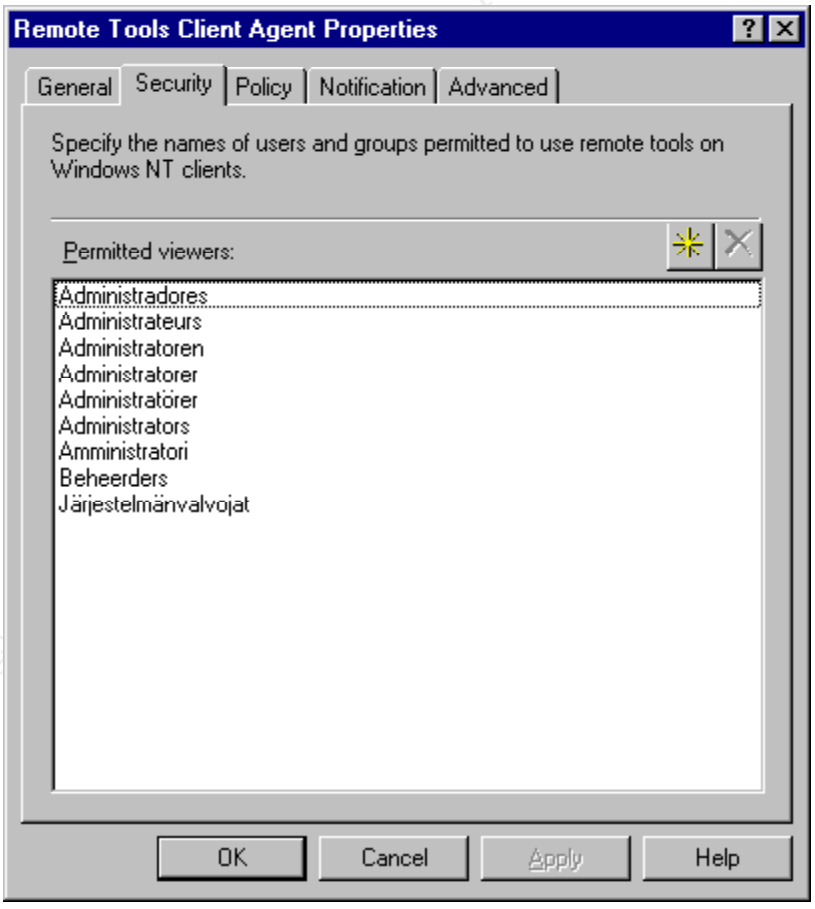
"Local administrator rights are not required for user to able to use Remote Tools," according to Microsoft TechNet.[10]
The SMS 2.0 Administrator Guide misleads SMS administrators by stating that administrator privileges are required for Remote Tools.[11]

The members of the Permitted Viewers list not only don't need to have administrator privileges, they also are not required to be a member any of special local and global group that was created at the time of installation of SMS server.

The "official" way to modify the list of Permitted Viewers is by going to SMS Administrator Console to:

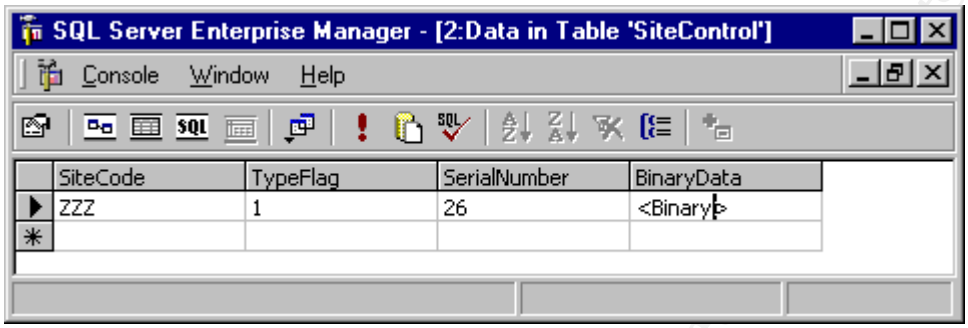
Systems Management Server
· Site Database (site code - site name)
· Site Hierarchy
· Site code - site name
· Site Settings
· Client Agents
· Remote Tools Client Agent Property sheet
· Security tab:

(See the screen capture below)



The list of Permitted Viewers can be modified in the database field "BinaryData" at the table

"SiteControl" at SMS Site database. (See the screen capture below)
This is how the attacker can modify permissions without access to the SMS Administrator Console that would require an attacker to pass the authentication as the permitted SMS Administrator.



The configuration parameters set in the Remote Tools Client Agent Property sheet are propagated to the SMS clients, and stored under the Registry key:

HKLM\Software\Microsoft\SMS \Client\Sites
System
<Site code>
Client Components
Remote Control

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\SMS\Client\Client Components\Remote Control]
"Control Level"=dword:00000002
"Permission Required"=dword:00000000
"Allow Takeover"=dword:00000001
"Allow Remote Execute"=dword:00000001
"Allow File Transfer"=dword:00000001
"Allow Reboot"=dword:00000001
"Allow Chat"=dword:00000001
"Visible Signal"=dword:00000000
"PermittedViewers"=hex(7):41,64,6d,69,6e,69,73,74,72,61,64,6f,72,65,73,00,41,
64,6d,69,6e,69,73,74,72,61,74,65,75,72,73,00,41,64,6d,69,6e,69,73,74,72,61,
74,6f,72,65,6e,00,41,64,6d,69,6e,69,73,74,72,61,74,6f,72,65,72,00,41,64,6d,
69,6e,69,73,74,72,61,74,f6,72,65,72,00,41,64,6d,69,6e,69,73,74,72,61,74,6f,
72,73,00,41,6d,6d,69,6e,69,73,74,72,61,74,6f,72,69,00,42,65,68,65,65,72,64,
65,72,73,00,4a,e4,72,6a,65,73,74,65,6c,6d,e4,6e,76,61,6c,76,6f,6a,61,74,00,
74,65,73,74,61,63,63,6f,75,6e,74,00,00
"Allow Ping Test"=dword:00000001
"CommandLine"="-TCP"
"Always Visible"=dword:00000000

"IndicatorType"=dword:00000000
"Audible Signal"=dword:00000000
"CompressionType"=dword:00000001
"Use IDIS"=dword:00000001
"System Tray Visible Signal"=dword:00000001
"System Tray Always Visible"=dword:00000000

Where Permitted Viewers are:

Administradores
Administrateurs
Administratoren
Administratorer
Administratörer
Administrators
Amministratori
Beheerders
Järjestelmänvalvojat
testaccount

It is possible to launch Remote Tools from the command line by providing the IP address, IPX address, NetBIOS name or SMS system ID of the SMS client host. In our case we will use an IP address that an attacker can find at the SMS Site database in the table "System_IP_Address_ARR".
At the time of the launch, Remote Tools will try to get information from the Site server if the user of Remote Tools has permission to access a remote host. The fact that the user belongs to the list of Permitted Viewers will be checked at the remote host. After that, the user of Remote Tools is granted permission to access the remote host.

Diagrams

Network communication diagram

In our variant of attack, the Server “SMSSERVER” will host both SMS Server and SQL Server.

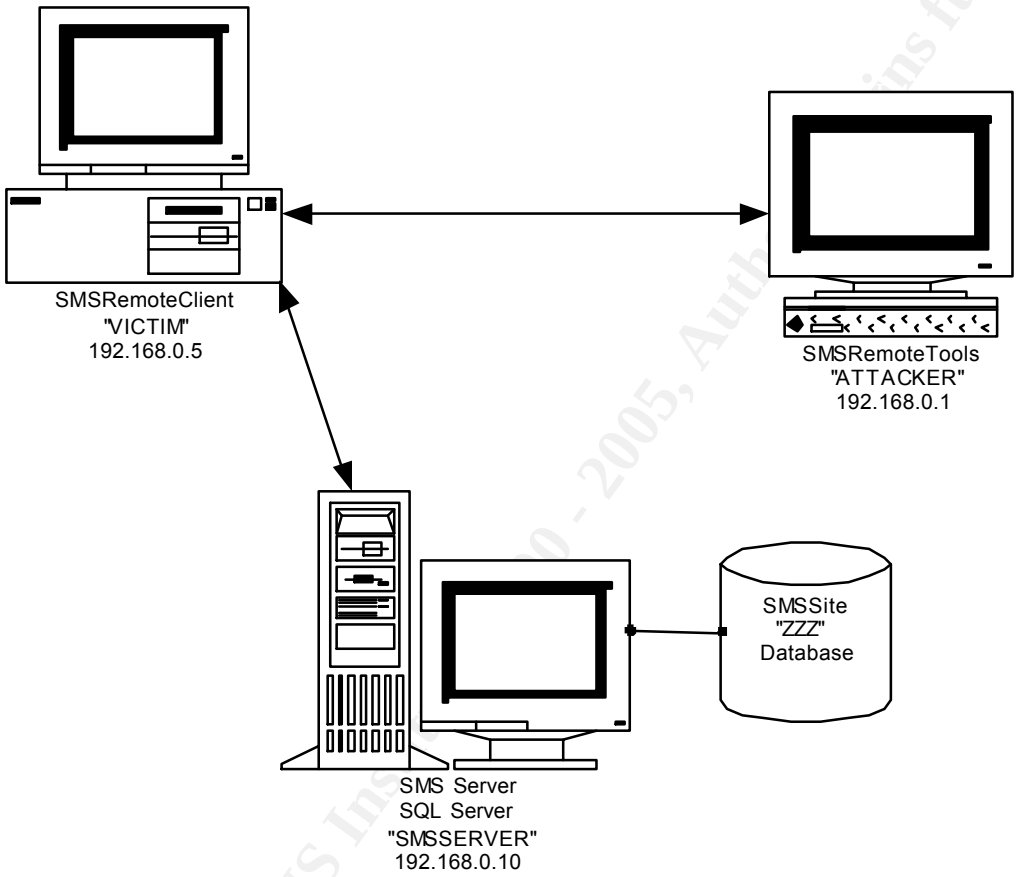
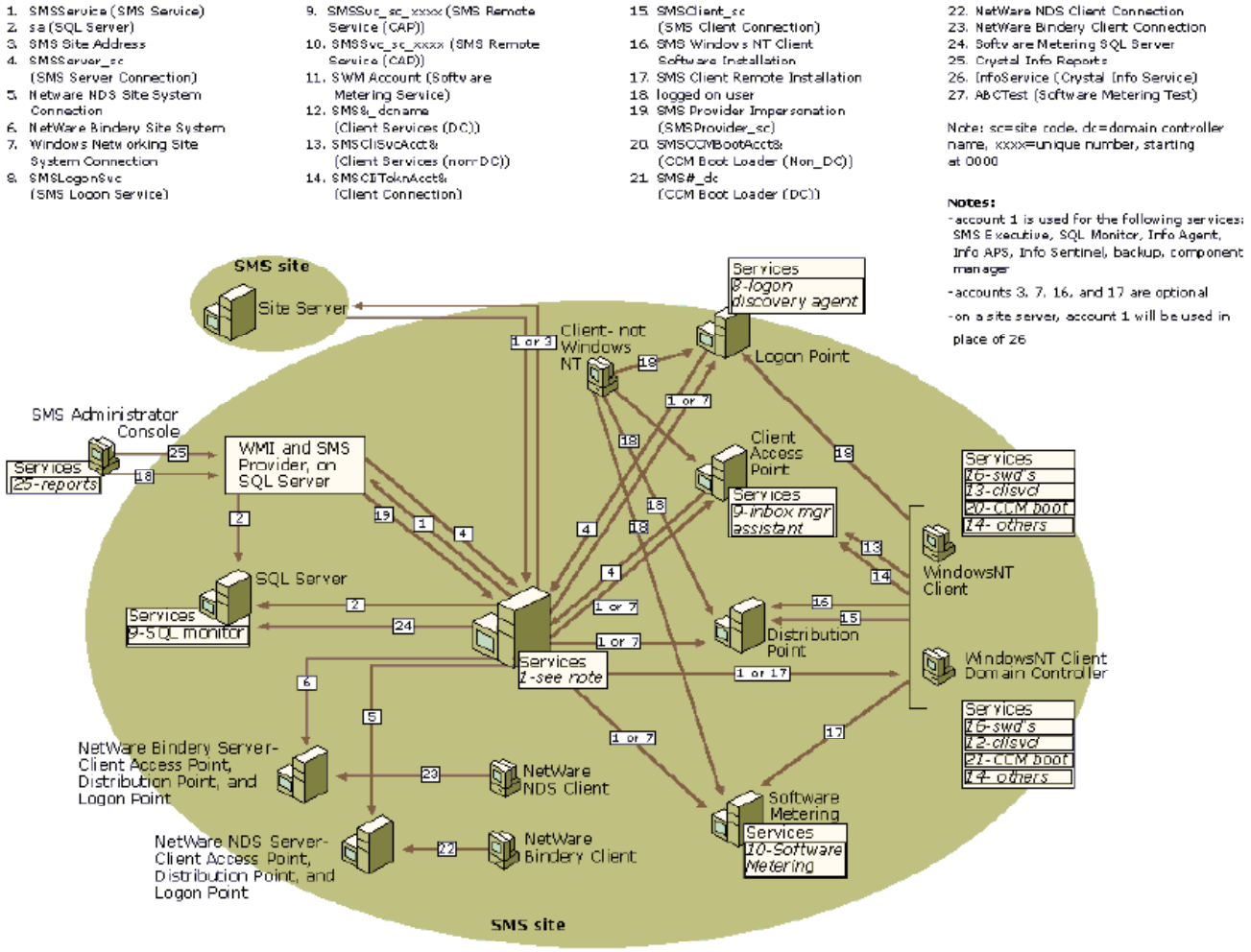


Diagram of SMS 2.0 Server Communication



The diagram is from Microsoft Corporation, TechNet, Paul Thomsen, Technical Writer, MS Systems Management Server Security Essentials.[10]

How to use the exploit

Reconnaissance

In order to use this exploit, some reconnaissance and scan must be conducted to see if SMS server is being used on the network.
The easiest way to discern the presence of SMS server and SMS clients is by sniffing network traffic, using standard tools such as TCPDUMP and MS Network Monitor. If the ports that are used by SMS Remote Tools are detected in the network communications between network hosts, the attacker can use this discovery to complete this exploit.

For example, below is output from TCPDUMP where we can see that the host 192.168.0.1 establishing connection to the host "VICTIM", port 2701/tcp:

22:21:48.042693 192.168.0.1.1046 > VICTIM.2701: S 1767769180:1767769180(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
22:21:48.042989 VICTIM.2701 > 192.168.0.1.1046: S 194987:194987(0) ack 1767769181 win 8760 <mss 1460> (DF)
22:21:48.043116 192.168.0.1.1046 > VICTIM.2701: . ack 1 win 17520 (DF)
22:21:48.043563 192.168.0.1.1046 > VICTIM.2701: P 1:17(16) ack 1 win 17520 (DF)
22:21:48.046696 VICTIM.2701 > 192.168.0.1.1046: P 1:198(197) ack 17 win 8744 (DF)
22:21:48.078066 192.168.0.1.1046 > VICTIM.2701: P 17:33(16) ack 198 win 17323 (DF)
22:21:48.078594 VICTIM.2701 > 192.168.0.1.1046: P 198:395(197) ack 33 win 8728 (DF)
22:21:48.105134 192.168.0.1.1046 > VICTIM.2701: P 33:51(18) ack 395 win 17126 (DF)
22:21:48.106169 VICTIM.2701 > 192.168.0.1.1046: P 395:419(24) ack 51 win 8710 (DF)
22:21:48.135820 192.168.0.1.1046 > VICTIM.2701: P 51:67(16) ack 419 win 17102 (DF)
22:21:48.136239 VICTIM.2701 > 192.168.0.1.1046: P 419:435(16) ack 67 win 8694 (DF)
22:21:48.136540 VICTIM.2701 > 192.168.0.1.1046: F 435:435(0) ack 67 win 8694 (DF)
22:21:48.136651 192.168.0.1.1046 > VICTIM.2701: . ack 436 win 17086 (DF)
22:21:48.167049 192.168.0.1.1046 > VICTIM.2701: F 67:67(0) ack 436 win 17086 (DF)
22:21:48.167261 VICTIM.2701 > 192.168.0.1.1046: . ack 68 win 8694 (DF)

The fact that SMS Server has been installed on the network can be detected by the presence of NetBIOS shares (SMSLogon, CAP_*sitecode,* SMSPKG*x*$, SMS_SITE, SMS_CPS*x*$) [12]

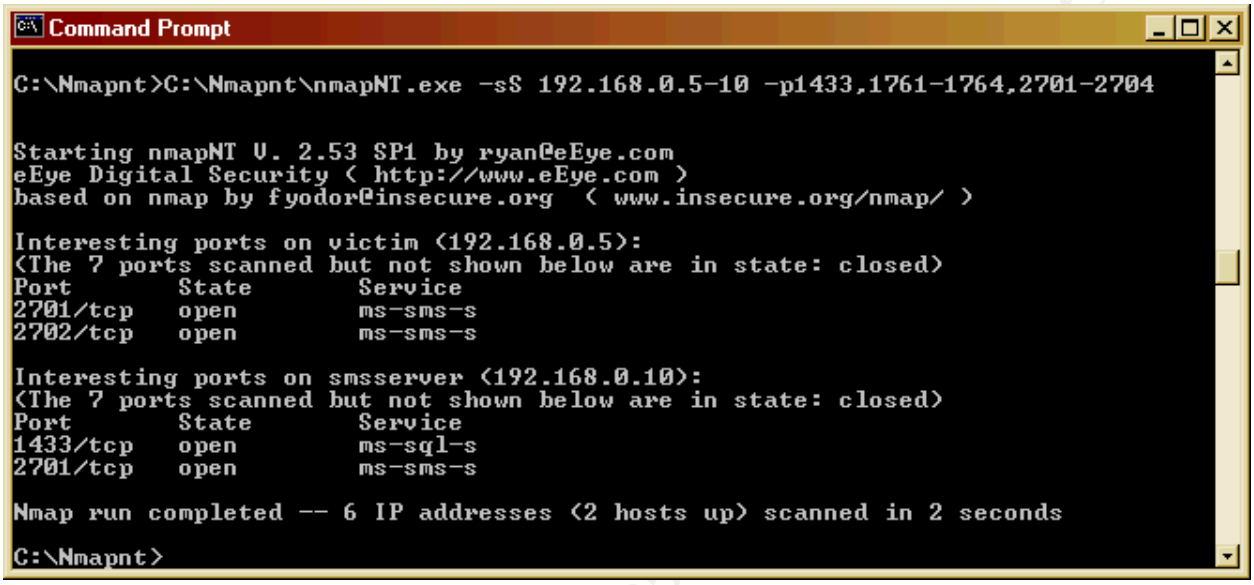
Of course, if the attacker gains access to the SQL database server, he or she can check if the database SMS_Site name exists. This will be a direct indication that SQL server hosts an <SMS_"Sitecode"> database.

Scan

The scan for open ports 1443/tcp (SQL Server), 1761/tcp, 1762/tcp, 1763/tcp, 1764/tcp, 1761/udp, 1762/udp, 1763/udp, 1764/udp, 2701/tcp, 2702/tcp, 2703/tcp, 2704/tcp, 2701/udp, 2702/udp, 2703/udp and 2704/udp ports (SMS Remote Tools clients) can be conducted in order to find

hosts with SQL server, SMS Server or Remote Tools Client Agent installed.

Below is an example of the NMAP port scan. As we can see the host "VICTIM" has opened ports 2701/tcp and 2702/tcp, the host "SMSSERVER" has opened ports 1433/tcp and 2701/tcp.

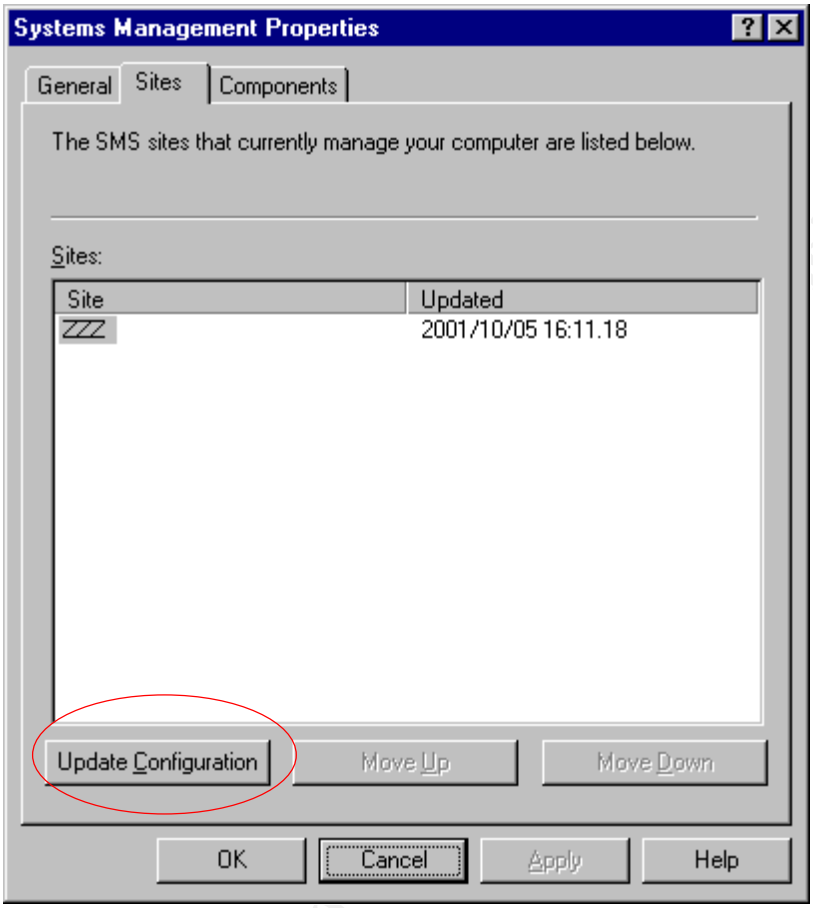


Exploit

The first step is to use one of the existing exploits of Microsoft SQL Server such as "SQLExec" [4,5] to get access to the SMS Site database. By utilizing one these exploits, an attacker can get a password for the "sa" login.

The second step is to modify the field "BinaryData" at the table "SiteControl" at the SMS Site database to include the account or group that the attacker will use to get remote access to the victim host. The user name can be one of the test accounts that exist in almost every network environment with passwords very easy to guess. The group name can be "Users" and by default all domain users will get permissions to use SMS Remote Tools.

The third step is to propagate the new configuration for Remote Tools to the victim host. That normally will happen at the time of the next update of the configuration of SMS client on the victim host. The scheduled update of configuration occurs when Client Component Installation Manager (CCIM) runs. That usually happens every 23 hours, or whenever the computer is booted up, or when the user clicks the "Update Configuration" button in the "Control Panel" – "Systems Management" applet. So, a relatively patient attacker can simply change the SMS Site database and wait for 23 hours until all clients of SMS Site have received their new, altered configuration of Remote Tools from the compromised SMS server. With the corrupted configuration in place, the attacker is set to do remote control of network hosts with a relatively low risk of detection.



The fourth step involves the actual remote access to the victim host. To access the remote SMS client, the attacker can start a Remote Control session from the command line. The syntax of remote control is[13]:

remote.exe <address-type ><name> [/SMS:Server=Site Server Name],

where <address-type> 1 for Novell; 2 for TCP and 3 for NetBIOS

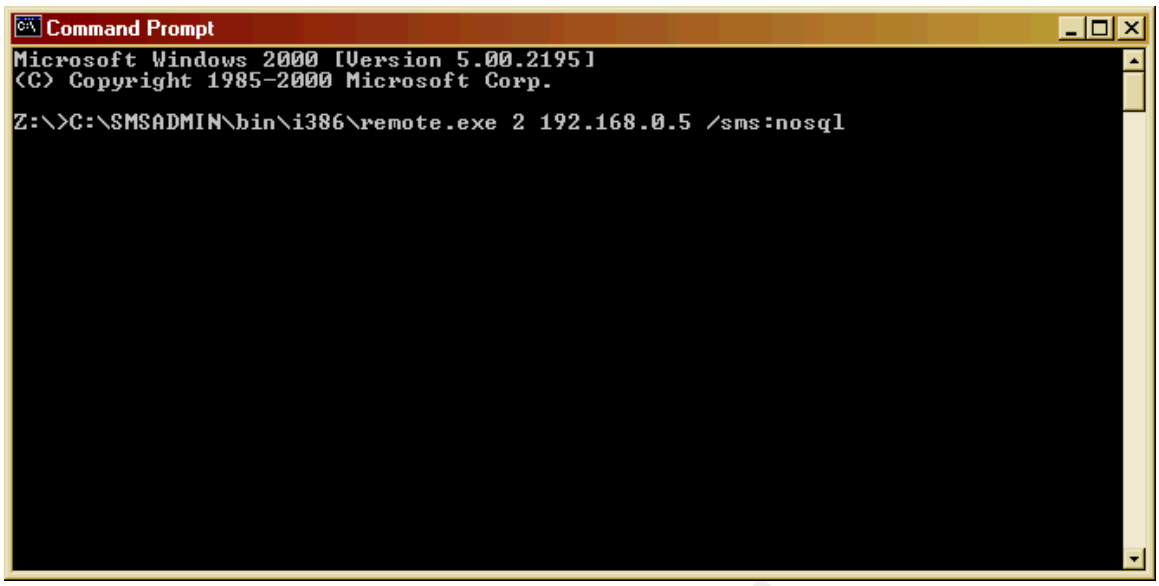
In our case the command will be:

C:\SMSADMIN\bin\i386\remote.exe <Victim IP Address> /SMS:NOSQL

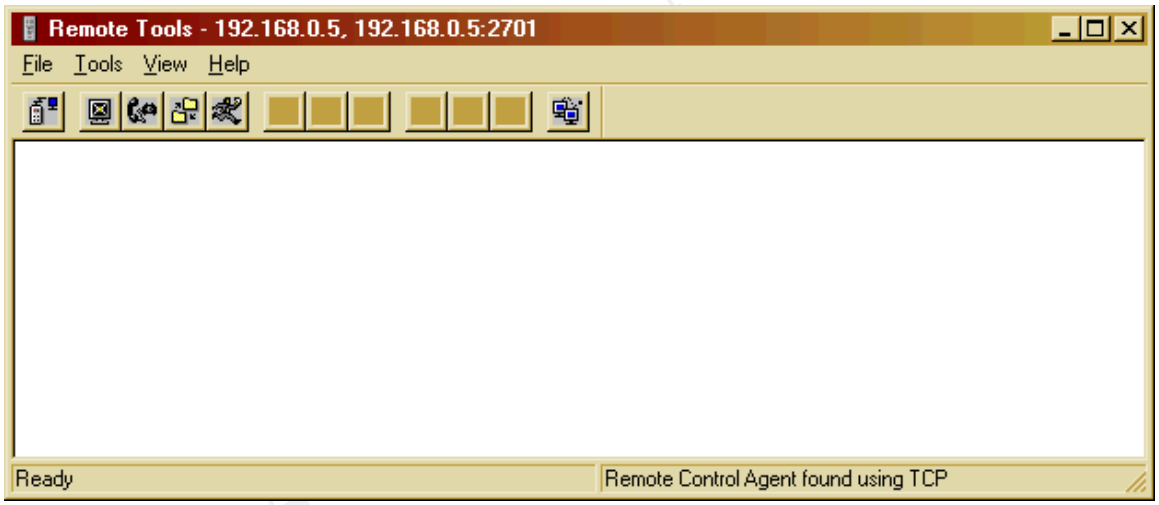
You can use the switch */SMS:NOSQL* with SMS Remote Tools only if TCP/IP is used as the network protocol for SMS communications.

The switch */SMS:NOSQL* will allow the attacker to bypass querying the SMS Site server whether the IP address that being is used is indeed the correct address of the SMS client, and that the user of Remote Tools has the proper permissions to gain remote control to the collections where the victim SMS client resides.

This option will allow attackers to make fewer changes to the SMS Site database than would ordinarily be required. Also, this will make the process of establishing a Remote Tools session sorter and generate less of network traffic. Together, these two aspects make it more likely that the exploit will remain hidden from network administrators and security staff.



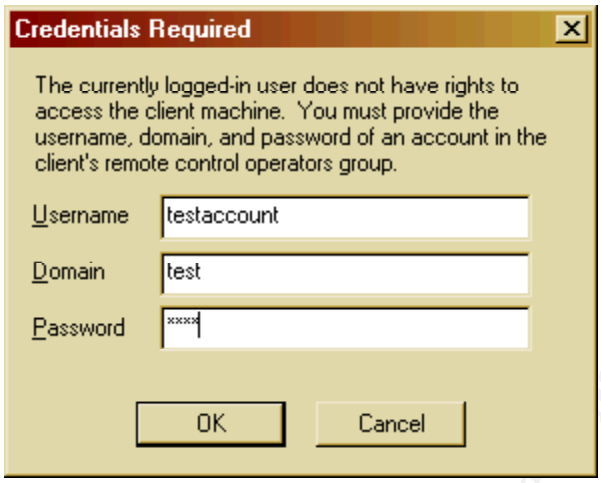
If the connection has been permitted the attacker will get the window:



This window shows that a connection has been established to the IP address 192.168.0.5 on TCP port 2701.

By going to the menu "Tools" – "Remote control," the attacker will initiate the interactive remote control session with the victim host.

If the attacker's local account is different from the account that the attacker planning to use for remote control sessions, he or she will get a prompt for the "valid account": in our case the test account "testaccount" from domain "test" for which the attacker had a password or was able to find a password. Once again, the supplied account doesn't need to be an account from the victim's local Administrators group, or an account that has administrator privileges on the victim host.



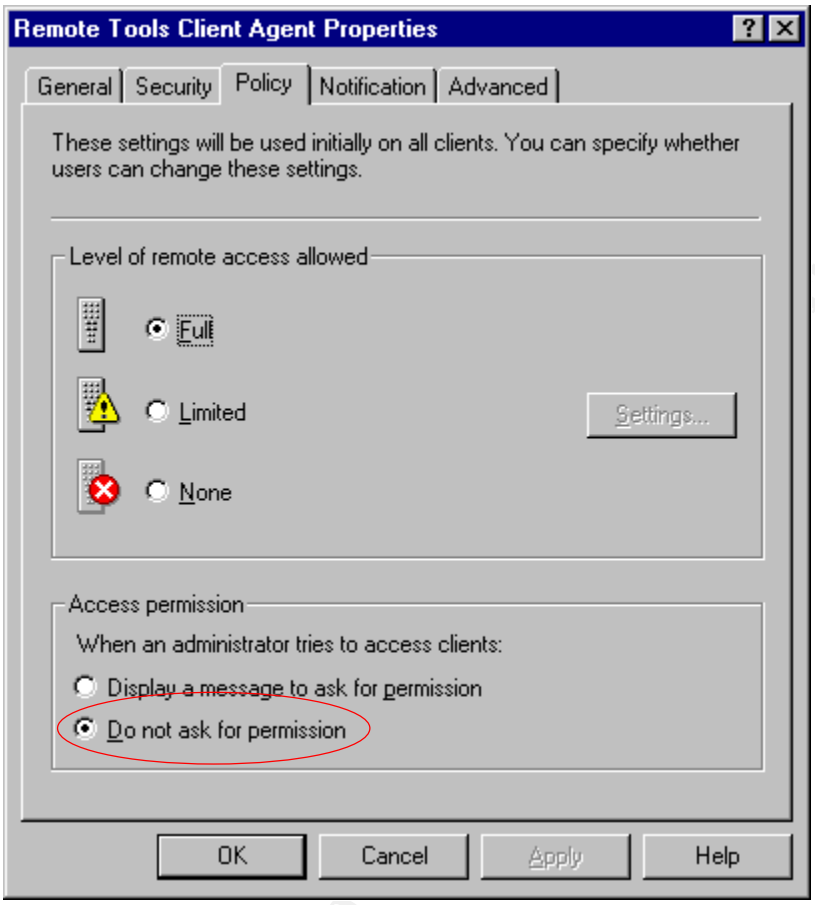
After the victim host verifies that the account the attacker supplied is the correct account, it will an accept a request for remote control, and the attacker will get the window:

The attacker has gained remote control for victim host!

By using Remote Tools, the attacker also can transfer files to the victim SMS client host, and run an executable file remotely on the victim host. This is a good opportunity for the attacker to install back doors other then SMS Remote Tools Client. The attacker can also take the opportunity to clean event logs indicating attackers remote connection, and otherwise cover up tracks.

SMS allows SMS client to be configured to request permission of the user of local host to accept or reject a remote connection. This is an important moment because the attack could be detected. To insure that the use of Remote Tools will not be detected, the attacker can use the fact that SMS Client will accept a request for a remote session without asking the user if the session is being initialized in less than two minutes from the last remote session. Microsoft has fixed this problem in SMS 2.0 Service Pack 3. [14] Or an attacker can modify another configuration option for SMS Remote Tools. The option "Do not ask for permissions" can be changed at:

Systems Management Server
· Site Database (site code - site name)
· Site Hierarchy
· Site code - site name
· Site Settings
· Client Agents
· Remote Tools Client Agent Property sheet
· Policy tab:



Or it also can be changed in the database field "BinaryData" at the table "SiteControl" at SMS Site database.

Keeping access

After gaining remote access, an attacker can keep access by installing back doors, creating new accounts or modifying existing accounts, user rights and polices. The attacker can modify configuration of SMS Remote Tools Client Agent in a way that will preserve configurations that the attacker originally created in SMS Site database and that have been propagated to the remote host. This will allow preserving access to the victim host even in case the SMS configuration has been changed to pre attack stage.

Covering tracks

One way to cover the fact that the configuration of SMS Remote Tools Client Agent has been modified is to change the SMS field in the SMS Site database right after the change has been propagated to the clients and SMS Remote Tools has been used.

After the attacker has gained access to the remote host, he or she can create a local account that is part of the local Administrators group, or crack the password of a local Administrator account and use those accounts to establish Remote Tools sessions. At that point, the user or group

names that the attacker added to the list of Permitted Viewers are no longer needed and can be removed. After this, the SMS Site database will look the same as before the attack, and no intrusion can be detected by comparing the two stages of the SMS Site database.

Also if the list of the Permitted Viewers still has the default members:

Administradores
Administrateurs
Administratoren
Administratorer
Administratörer
Administrators
Amministratori
Beheerders
Järjestelmänvalvojat

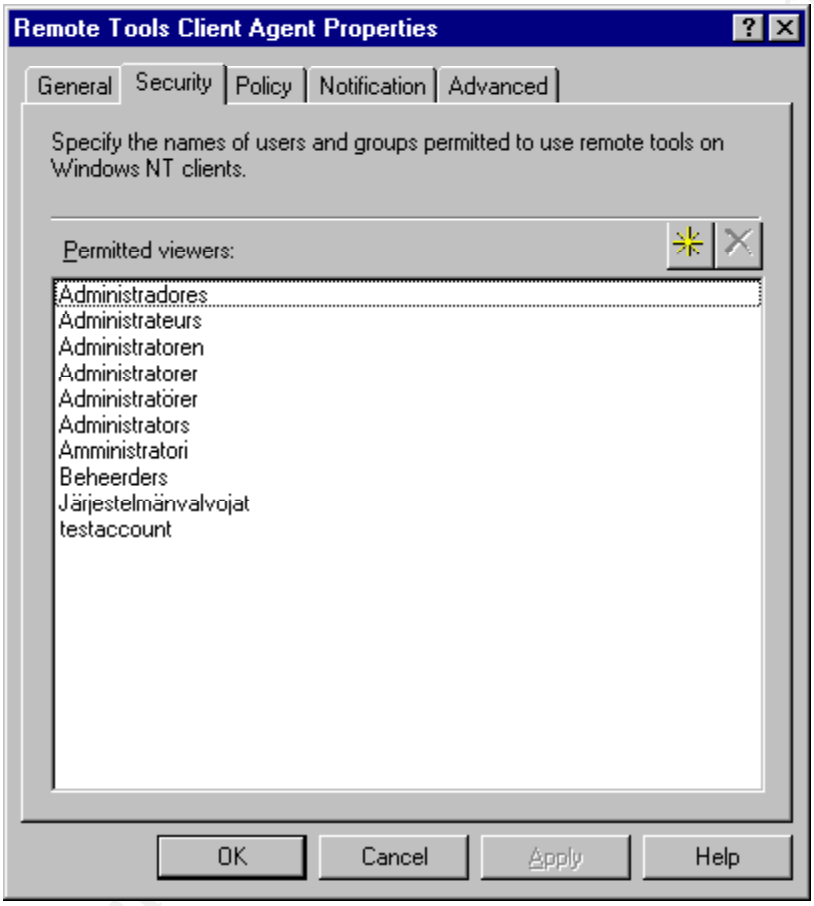
The attacker can create a local or global account or user group with one of these names and use it for remote access. If the attacker doesn't have permissions to create accounts, he or she can ask Help Desk personnel to create an account "for a new intern with name Beheerders, with just read permissions".

On the victim host, the attacker can delete events in the Windows NT security event log that indicate that the attacker established the remote control session. The attacker can delete the entire Security event log or use tools like WinZapper (http://ntsecurity.nu/cgi-bin/download/winzapper.zip.pl).

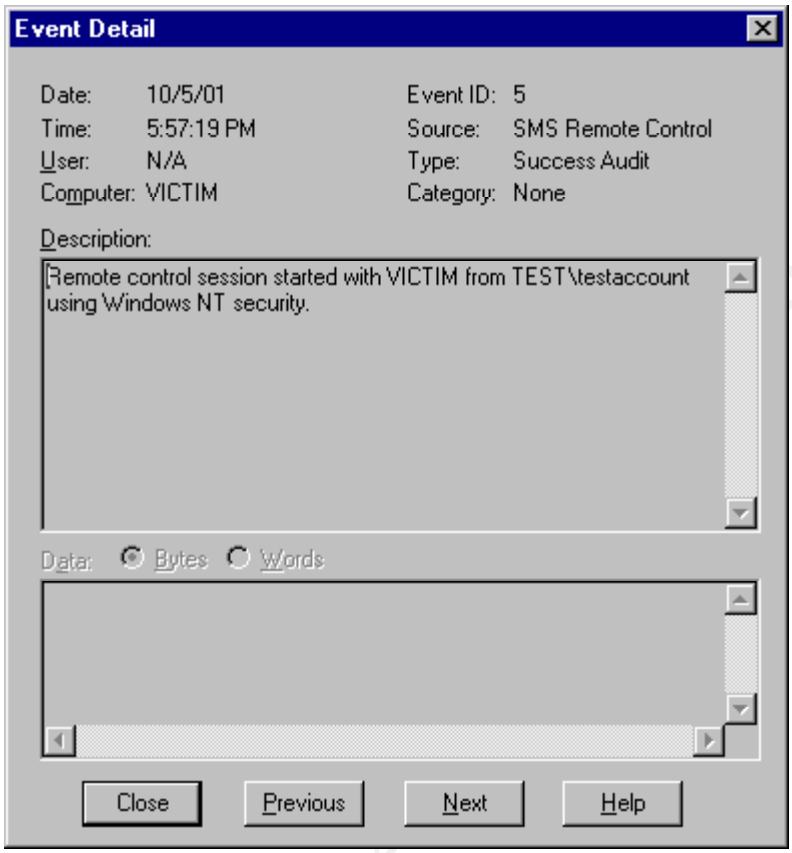
Obviously, the remote control of the victim host by the attacker will be noticeable by an interactive user of the victim host. In order to avoid detection, the attacker will need to access the Remote Tools client host at a time when no interactive users will be present on the computer, most likely nights and weekends.

Signature of the attack

An indication that someone is using this exploit will be the fact that the list of Permitted Viewers has been modified. Below we can see the inappropriate account "testaccount" included in the list of Permitted Viewers:



Also, Security Event Logs on the victim host will have records of remote access to the host. For example, below is the Event Detail for the remote control session with host "Victim" and account "testaccount" used by the attacker:



Also if you see connections to your hosts established on one of the posts associated to this exploit from unexpected source as in TCPDUMP output below, you can expect that someone is using SMS Remote Tools without authorization.

02:48:58.205952 attacker.1058 > VICTIM.2701: S 3547720291:3547720291(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
02:48:58.206209 VICTIM.2701 > attacker.1058: S 57253:57253(0) ack 3547720292 win 8760 <mss 1460> (DF)
02:48:58.206401 attacker.1058 > VICTIM.2701: . ack 1 win 17520 (DF)
02:48:58.206851 attacker.1058 > VICTIM.2701: P 1:17(16) ack 1 win 17520 (DF)
02:48:58.210074 VICTIM.2701 > attacker.1058: P 1:198(197) ack 17 win 8744 (DF)
02:48:58.236247 attacker.1058 > VICTIM.2701: P 17:33(16) ack 198 win 17323 (DF)
02:48:58.236818 VICTIM.2701 > attacker.1058: P 198:395(197) ack 33 win 8728 (DF)
02:48:58.264911 attacker.1058 > VICTIM.2701: P 33:51(18) ack 395 win 17126 (DF)
02:48:58.265915 VICTIM.2701 > attacker.1058: P 395:419(24) ack 51 win 8710 (DF)
02:48:58.295832 attacker.1058 > VICTIM.2701: P 51:67(16) ack 419 win 17102 (DF)
02:48:58.296256 VICTIM.2701 > attacker.1058: P 419:435(16) ack 67 win 8694 (DF)
02:48:58.296554 VICTIM.2701 > attacker.1058: F 435:435(0) ack 67 win 8694 (DF)
02:48:58.296719 attacker.1058 > VICTIM.2701: . ack 436 win 17086 (DF)
02:48:58.327033 attacker.1058 > VICTIM.2701: F 67:67(0) ack 436 win 17086 (DF)
02:48:58.327228 VICTIM.2701 > attacker.1058: . ack 68 win 8694 (DF)

There remains one continued risk; that the attacker will be detected by users of the local console of the Windows NT host that is being attacked. The simplest way to reduce this risk is to try to use the remote control features of SMS client at the time that is out normal business hours. But this approach also gives systems administrators an opportunity to filter suspicious Remote Tools sessions and detect attacks. For example in TCPDUMP above, the host ATTACKER established the remote session with host VICTIM at 2:48 am. It is very doubtful that a legitimate network administrator with remote control rights would try to do it at this time.

How to protect against this exploit

The system administrator

It is a good practice to assign a good strong password, alphanumeric and random-generated. It is also good practice to review the configuration of SMS Site on a regular basis. Modify the list of Permitted Viewers to keep only the required accounts and groups and review it on a regular basis. It is also possible to monitor network traffic for which hosts are trying to initiate remote control sessions and to see if remote sessions are being initialized from computers not assigned to the permitted user of SMS Remote Tools.

The vendor

Microsoft can rethink the practice of storing the critically important SMS configuration information in an unencrypted form in the SQL database. The fact that the SMS service accounts passwords are not stored there does not guarantee that the integrity of SMS will not be compromised if the intruder can get access to the SMS Site database.
The option to use the "sa" login for SMS Site database should be disallowed by default.
Also it would be a good idea to change passwords for the SMS Site database connection account programmatically on a regular basis with randomly generated passwords.

Source code/Pseudo code

Because the exploit does not have an automated program at this time, and because to get remote access an off the shelf product (Microsoft SMS Server 2.0) has been used, no code can be presented.

Additional Information

The additional vulnerabilities of Microsoft SMS server

Microsoft Corporation, Microsoft Security Bulletin FAQ (MS00-012)

SecuriTeam.com, Default security permissions of SMS 2.0 Remote Control opens a security hole, 3/1/2000, http://www.securiteam.com/windowsntfocus/5IP010U0AU.html

SecuriTeam.com, Patch Available for the Remote Agent Permissions Vulnerability, http://www.securiteam.com/windowsntfocus/5QP0C000BM.html

SecuriTeam.com, Buffer overflow in Network Monitor allows code execution, 25/2/2000, http://www.securiteam.com/windowsntfocus/6U0020K0AU.html

Microsoft Corporation. TechNet, SMS: List of Bugs Fixed in Systems Management Server 2.0 SP3 [Q280756], 3/7/2001, http://support.microsoft.com/support/kb/articles/Q280/7/56.ASP

The additional vulnerabilities of Microsoft SQL server

SANS Security Reading Room, Stephen Arehart, **SQL Server Security,** 11/10/2000, http://www.sans.org/infosecFAQ/win/SQL_sec.htm

SecuriTeam.com, Patch Available for the "SQL Server 7.0 Service Pack Password" vulnerability, 10/05/2001, http://www.securiteam.com/windowsntfocus/5RP080A1VM.html

SecuriTeam.com, Patch Available for the SQL Query Abuse Vulnerability, 10/05/2001, http://www.securiteam.com/windowsntfocus/5UP0B000GW.html

SecuriTeam.com, Microsoft releases safeguard guide for the MS SQL blank 'sa' vulnerability, 21/8/2000, http://www.securiteam.com/windowsntfocus/Microsoft_releases_safeguard_guide_for_the_MS_SQL_blank__sa__vulnerability.html

SecuriTeam.com, SQL Query Method Enables Cached Administrator Connection to be Re-used, 13/6/2001, http://www.securiteam.com/windowsntfocus/5NP0G154KI.html

Resources and References

[1] The Cult of the Dead Cow, "Don't Worry Windows Users, Everything Will BO2K". Release, July 19, 1999. URL: http://www.cultdeadcow.com/news/pr19990719.html, 11/20/2000

[2] Microsoft Corporation, TechNet, Paul Thomsen, Technical Writer, MS Systems Management Server Security Essentials, http://www.microsoft.com/smsmgmt/techdetails/secessentials.asp, 11/20/2000

[3] Microsoft Corporation, "SQL Server 7.0 Service Pack Password" Vulnerability, 5/10/2001, http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/ms00-035.asp

[4] SecuriTeam.com, SQLExec allows easy exploitation of default SQL passwords, 10/2/2001, http://www.securiteam.com/exploits/5YP0D003FQ.html

[5] GCIH PRACTICAL ASSIGNMENT, SQLEXEC v 1.0 exploit, by Chuck Crawford, 5/28/2001, http://www.sans.org/y2k/practical/Chuck_Crawford_GCIH.zip

[6] SANS Security Reading Room, Gary A. Wong, Microsoft Systems Management Server (SMS) 2.0 Features: Security Policy and Controls Implications Within an Enterprise, 11/21/2000, http://www.sans.org/infosecFAQ/win/SMS_features.htm

[7] Microsoft Corporation, TechNet, Using SMS Object Security to Control the Use of SMS 2.0 Features, September 2000, http://www.microsoft.com/TechNet/prodtechnol/sms/maintain/optimize/colsec.asp

[8] Microsoft Corporation, TechNet, Paul Thomsen, Technical Writer, MS Systems Management Server Security Essentials, 3/22/2000, http://www.microsoft.com/smsmgmt/techdetails/secessentials.asp

[9] Microsoft Corporation. TechNet, Systems Management Server 2.0 Resource Guide, Part 4 - Administrating and Maintaining, Chapter 9 - Remote Tools for the Advanced User, 2001, http://www.microsoft.com/technet/prodtechnol/sms/reskit/sms2res/part4/smc09.asp

[10] Microsoft Corporation, TechNet, Paul Thomsen, Technical Writer, MS Systems Management Server Security Essentials, 3/22/2000, http://www.microsoft.com/smsmgmt/techdetails/secessentials.asp,

[11] Microsoft Corporation, Systems Management Server 2.0 Administrator's Guide, Part 3, chapter 15, Configuring the Remote Tools Client Agent, 1999, http://www.microsoft.com/technet/prodtechnol/sms/proddocs/smsadm/part3/smsad15.asp?frame=true.

[12] Microsoft Corporation, Systems Management Server 2.0 Administrator's Guide, Part 1, Chapter 4, Using Windows NT File and Directory Security, 1999, http://www.microsoft.com/technet/prodtechnol/sms/proddocs/smsadm/part1/smsad04.asp?frame=true.

[13] Microsoft Corporation. Microsoft Knowledge Base. SMS: Version 2.0 Remote Control Command Line Options [Q201793], 7/27/2001, http://support.microsoft.com/support/kb/articles/Q201/7/93.ASP

[14] Microsoft Corporation, Microsoft Knowledge Base. Remote Control Residual Permissions Override Permission Required [Q248452], 2/7/2001, http://support.microsoft.com/support/kb/articles/Q248/4/52.ASP