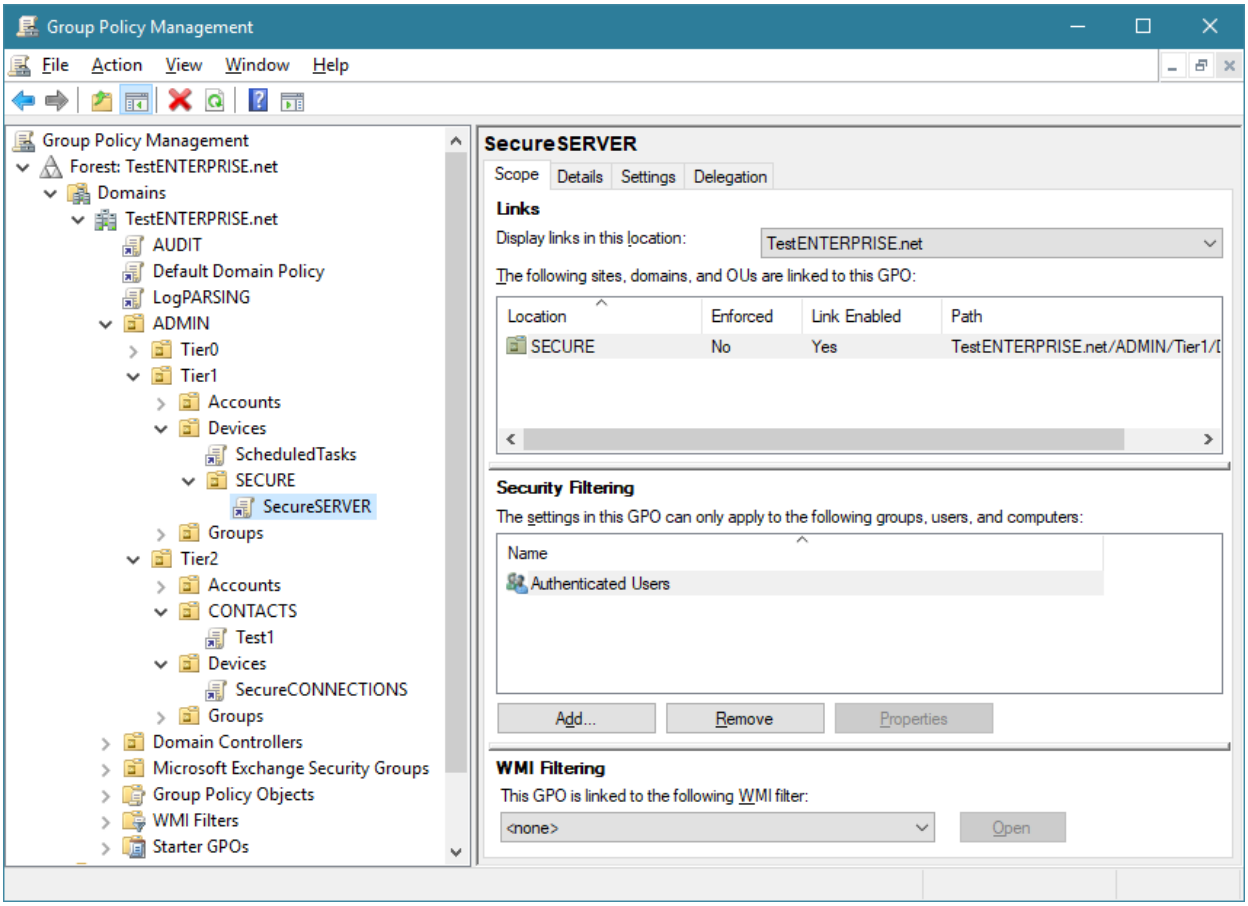
Implemeting IPSec in Windows Domain – part 2

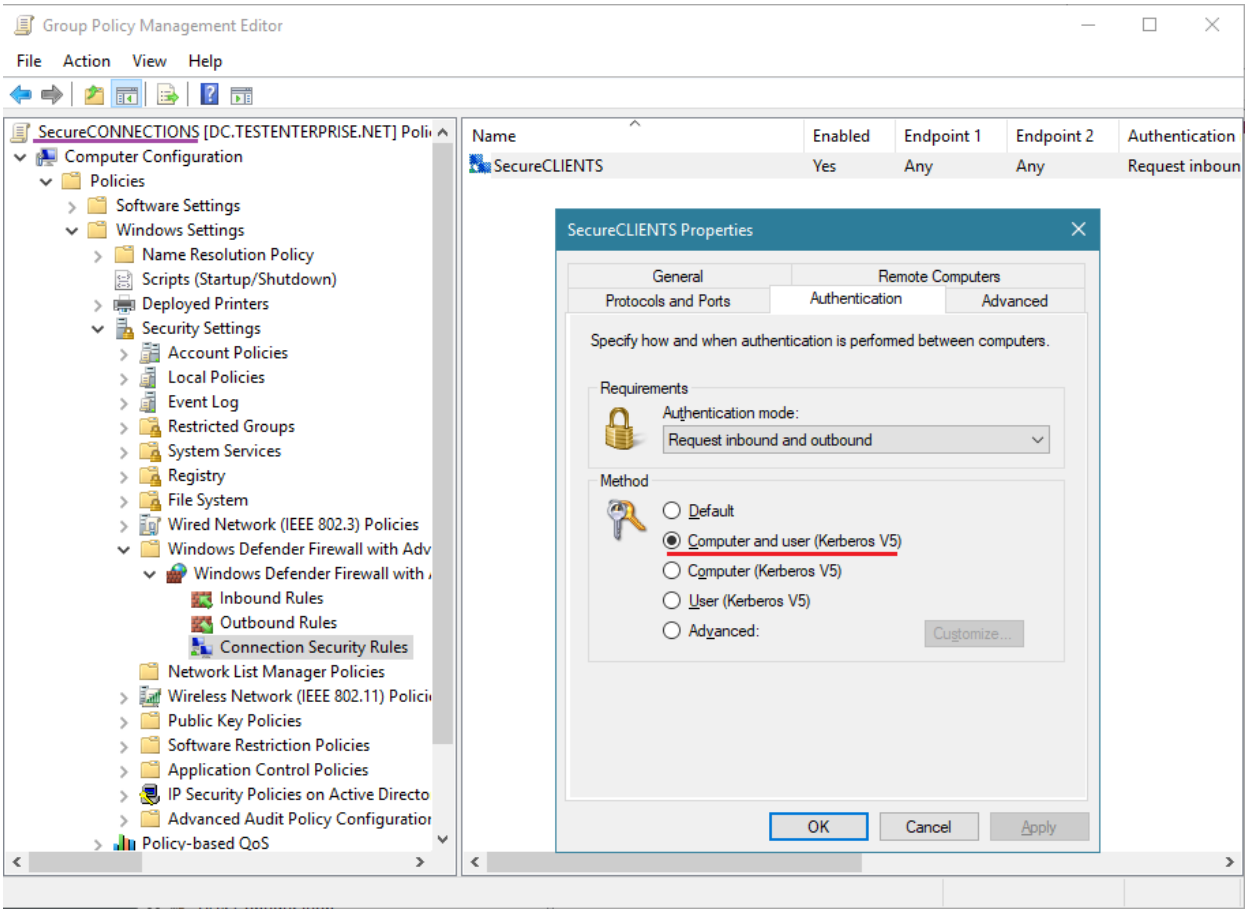
michaelfirsov.wordpress.com/implemeting-ipsec-in-windows-domain-part-2

November 26, 2019

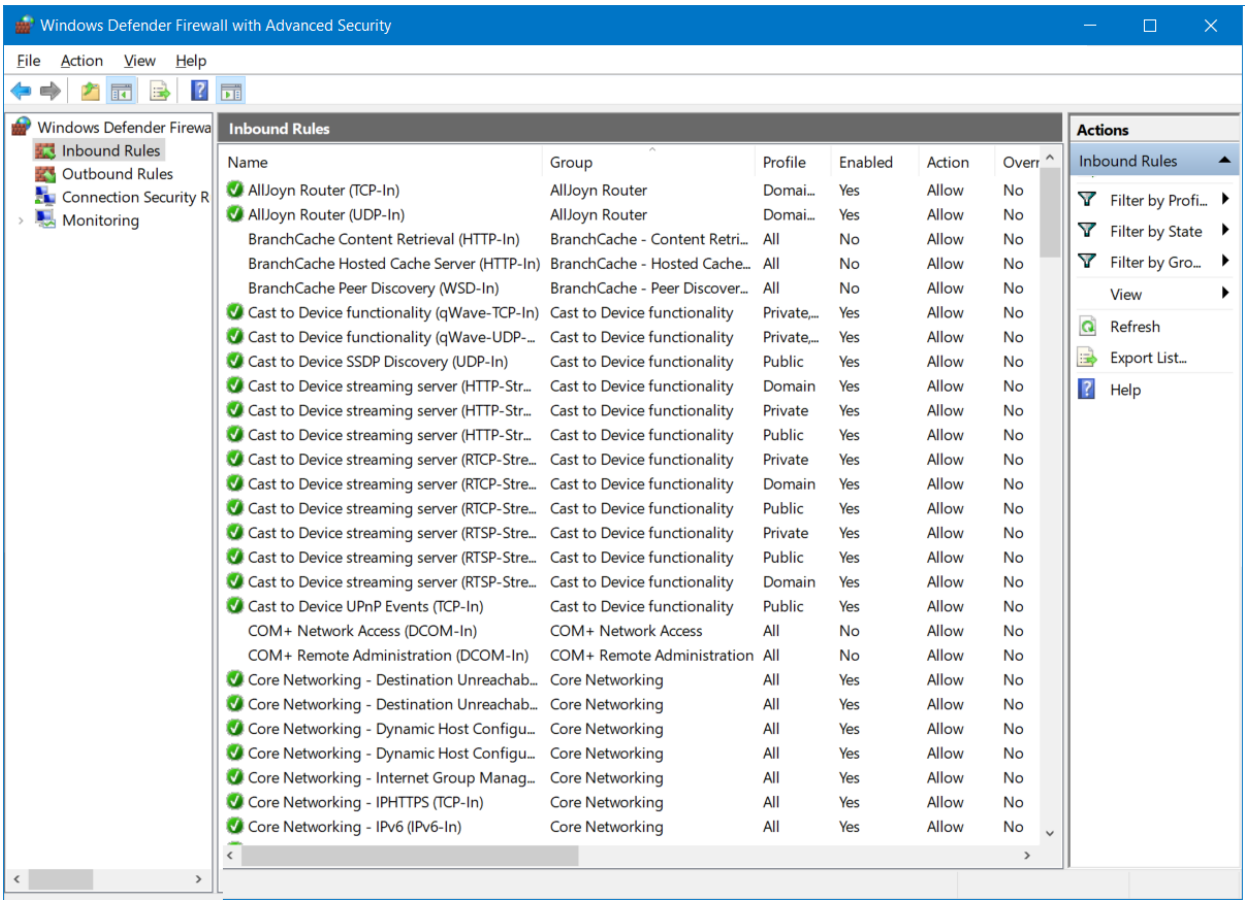
In part 1 we've seen how we can isolate a server by applying the appropriate connection security rules to the specific OUs only – in this article I'd like to show how we can add either an additional security layer to the configuration described in part 1 or use this new layer instead of the previous configuration. Let's recap how the ipsec have been deployed in part 1: there's the SecureServer gpo that requires all inbound traffic to be authenticated and encrypted (it applies to the OU containing the secure SQL server machine) and the SecureCONNECTIONS gpo that only requests the ipsec protection (it appiles to the OU with client computer accounts).



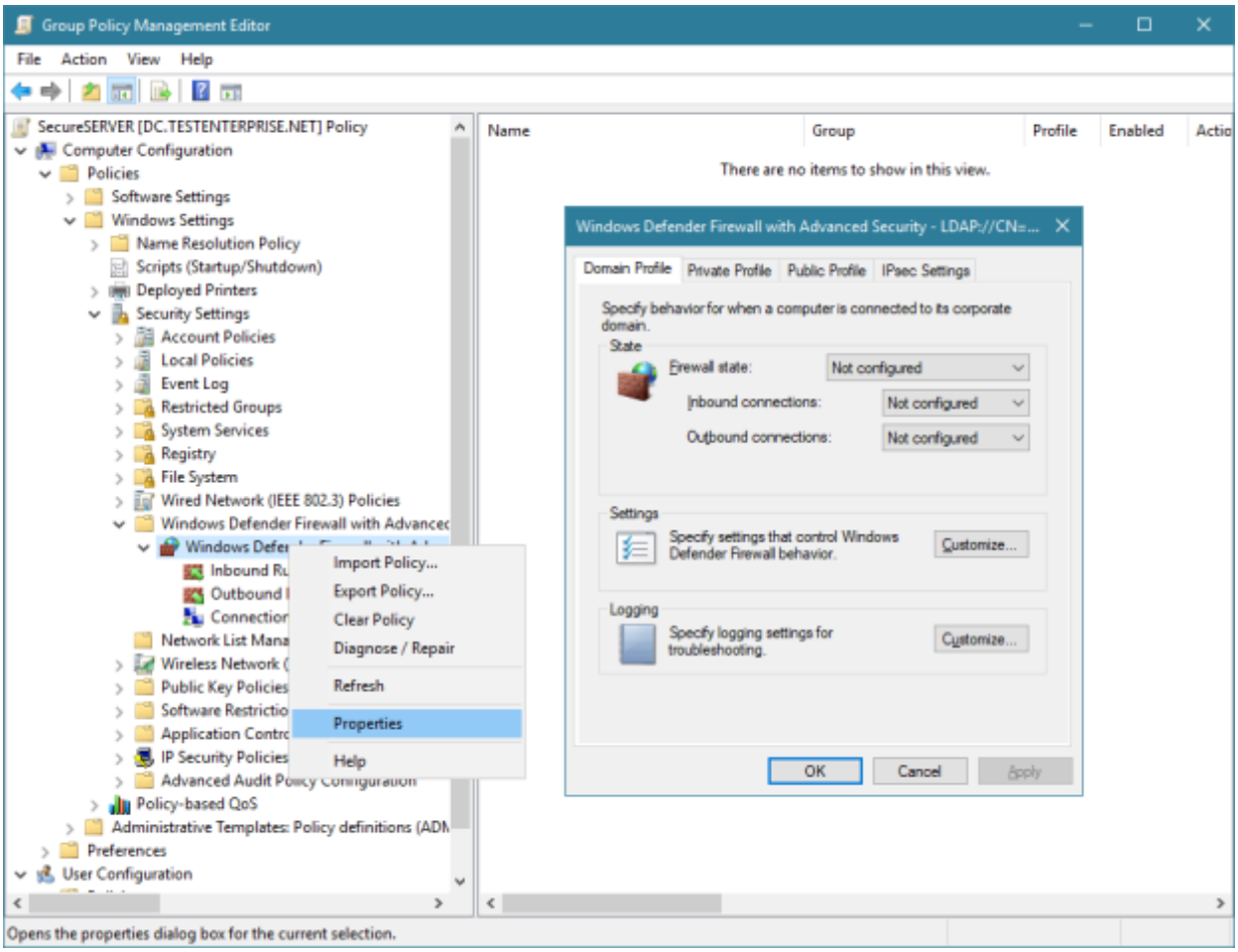
Both GPOs contains the connection security rules that uses Kerberos authentication for computer and user accounts. It is this connection security rule setting – **Computer and User (Kerberos V5)** – that will allow us to tighten the network security by leveraging per-rule computer or/and user authentication:



Before delving into the rules specifics I'd like to draw your attention to the following fact: we can configure ipsec communications between the hosts without creating any ipsec-specific firewall rules – if you've read the part 1 you'd have noticed that the only type of rules that have been created are the connection security rules. Windows Firewall – even in default configuration – does not contain the rules that would permit traffic on UDP 500/4500 ports (neither in the Inbound rules no in the Outbound rules) – it differs WF from other vendors' firewalls that may require creating the rules for the "raw" IPSsec packets (udp 500/4500) prior to configuring any IPSec associations. Of course, for packets to go in/out of a network interface the "ordinary" rules (the rules for the protocols/ports that are being used by various applications – icmp, tcp 1433 for the SQL Server connections and etc. – all that can be named "IPSec payload" when an IPSec session is already established) must exist for both inbound and outbound traffic and these rules do exist by default on any Windows host:



Since the goal is to control network communications with the secure server as mush as possible (at least the incoming communications), let's first configure the Windows Firewall's section of the server gpo in such a way that the WF on the target server will use only the rules added to this gpo and local administrators will not be able to apply the local rules. For this navigate to Computer ConfigurationPolices\Windows Settings\Security Settings\Windows Defender Firewall with Advanced Security, right-click the *Windows Firewall with Advanced Security – LDAP://cn={GUID},cn=policies,cn=…,DC=…,DC=….* and select Properties:

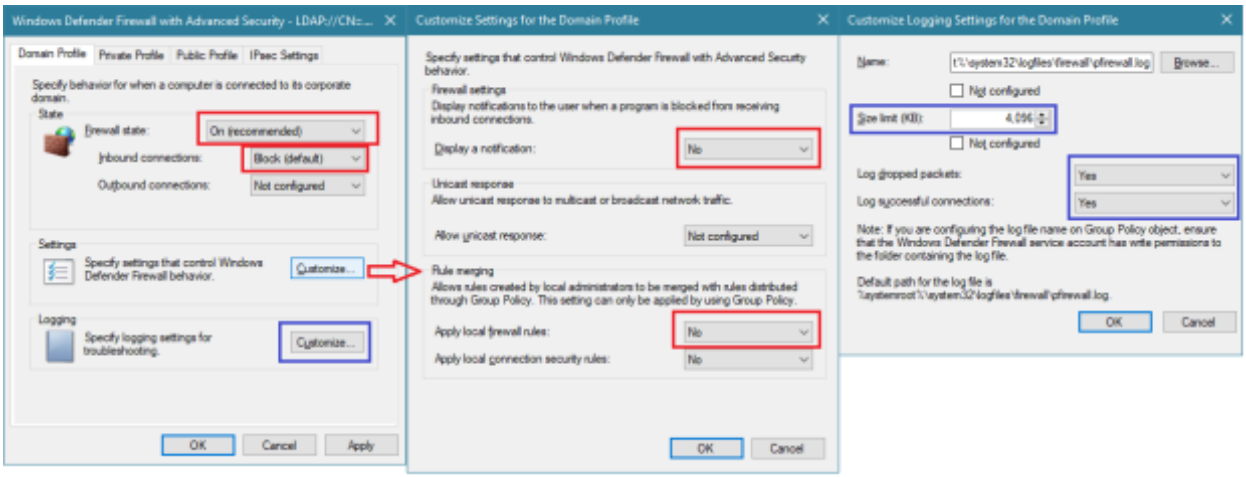


Advertisements

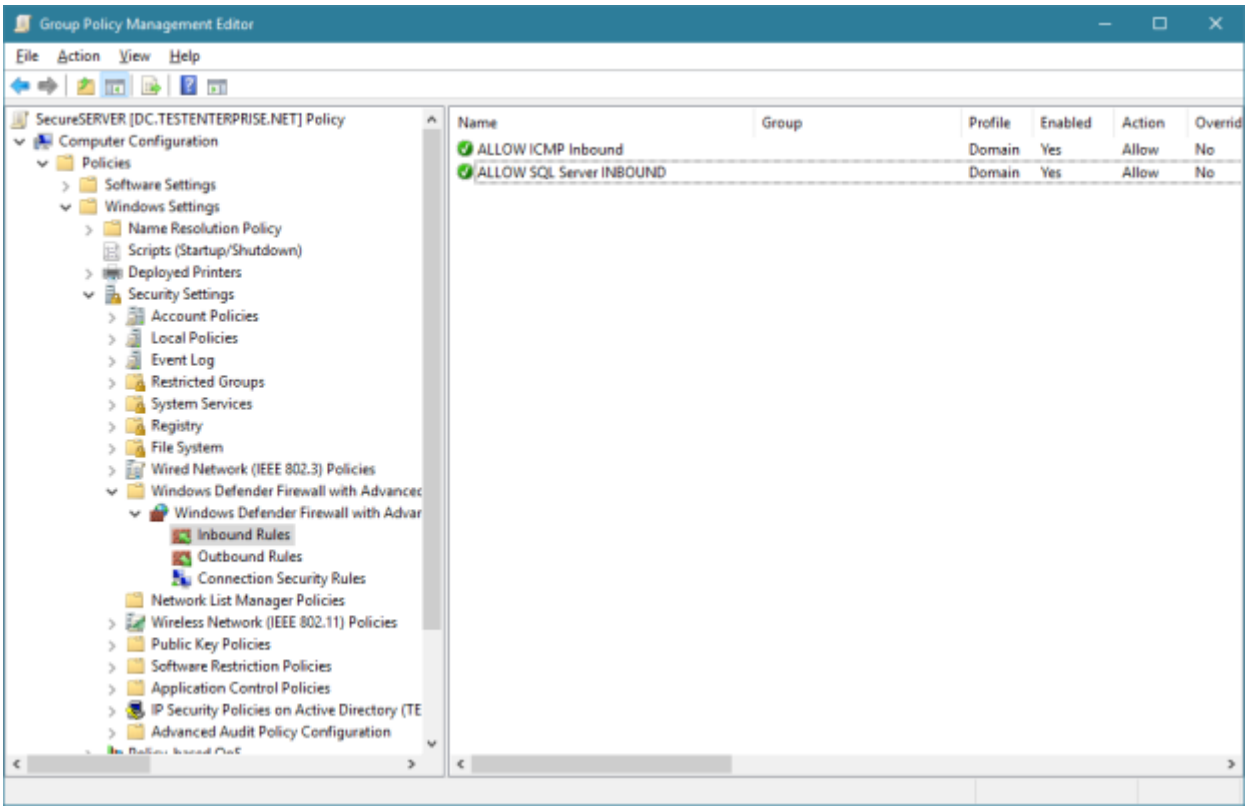
Report this adPrivacy

The configuration that follows will serve the following puposes:

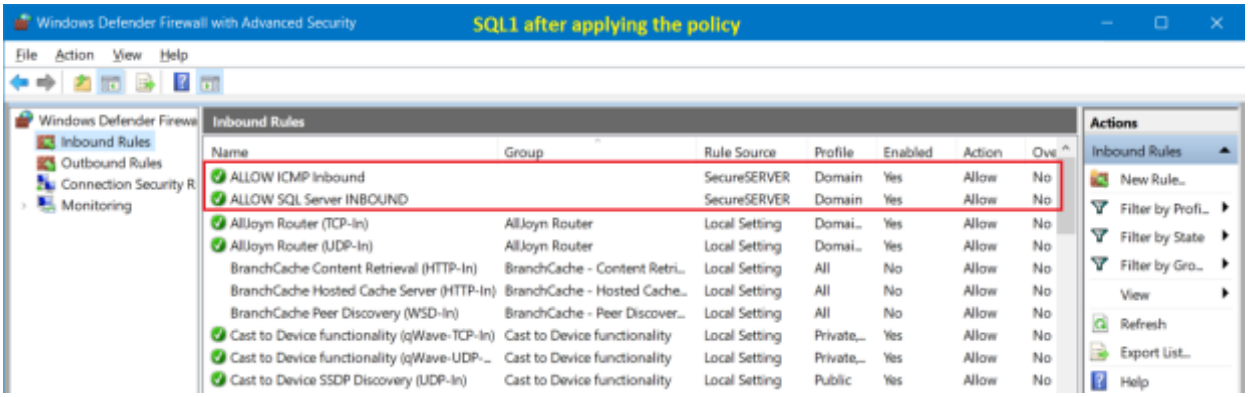
1) Firewall is turned on
2) All incoming connections are blocked
3) No questions arise when a new application tries a connection (*Display a notification* = No)
4) Local rules do not apply (*Apply local firewall rules* = No)
*5) logging for dropped and allowed packets is turned on – for testing porpuses only



Here's the minimal rule set that would allow to 1) ping the secure server 2) connect to the MS SQL Server hosted on SQL1:

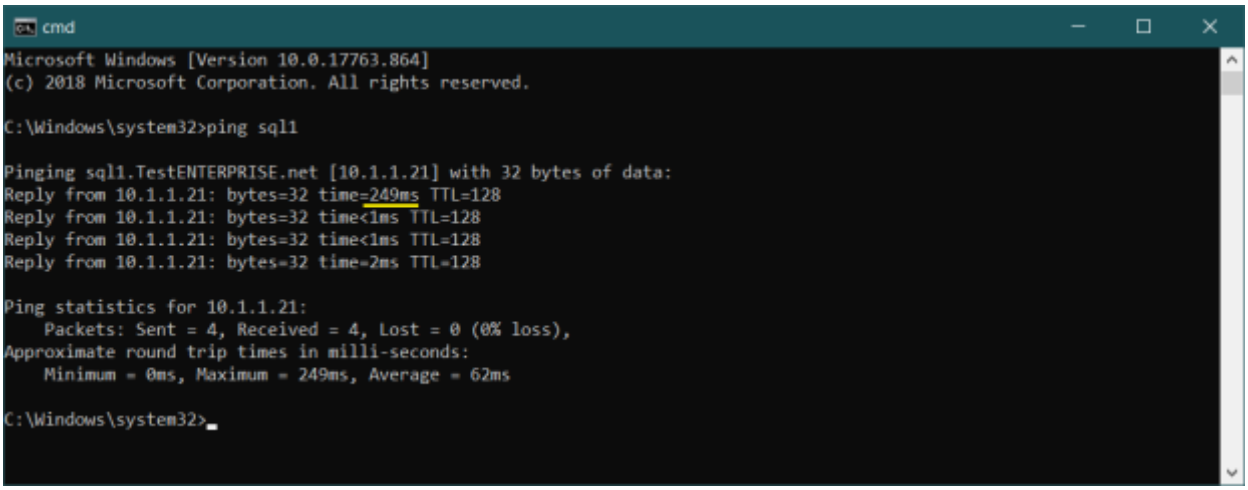


Once the SecureSERVER policy is applied to the SQL1 host, the new rules can be seen on top of the rule set in the local SQL1's WF console if ordered by the Rule Source field (you can add this field in the View menu) – only these two rules will affect the network traffic as the *Apply local firewall rules* has been set to No.

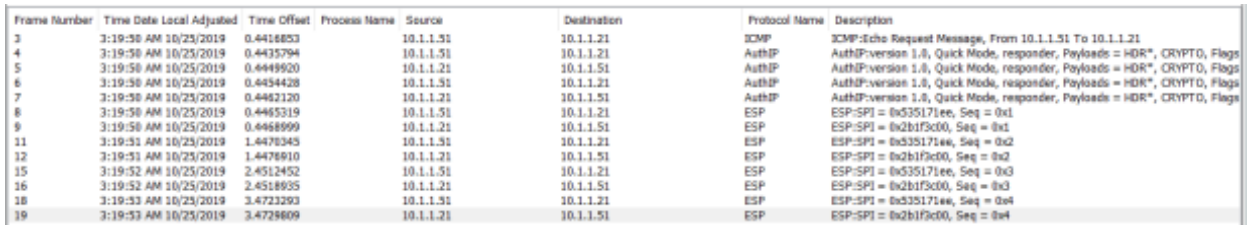


If we now conduct the same tests as in part 1 – pinging SQL1 and connecting to the MS SQL Server hosted on SQL1 from Client1 – we'll get exactly the same results because the same connection security rules from both GPOs still apply to SQL1 and Client1 and there are firewall rules that permit both types of traffic – icmp and tcp 1433, the only difference here is that Windows Firewall on SQL1 is now being ruled by the SecureSERVER gpo:

Test 1-1: On Client1 – Pinging SQL1

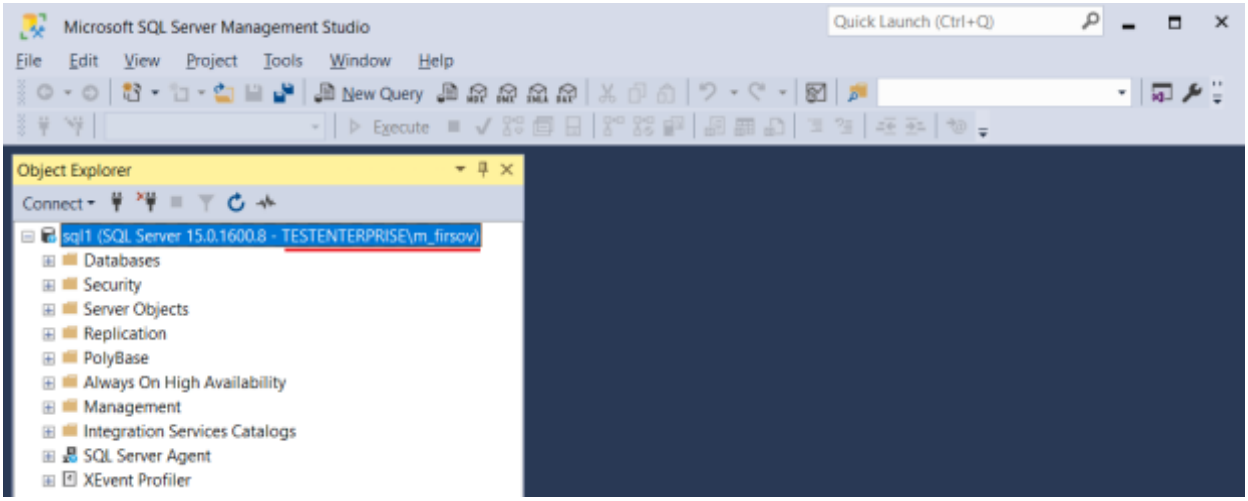


Again, the first response time of 249 ms means the IPSec communication has been just established.



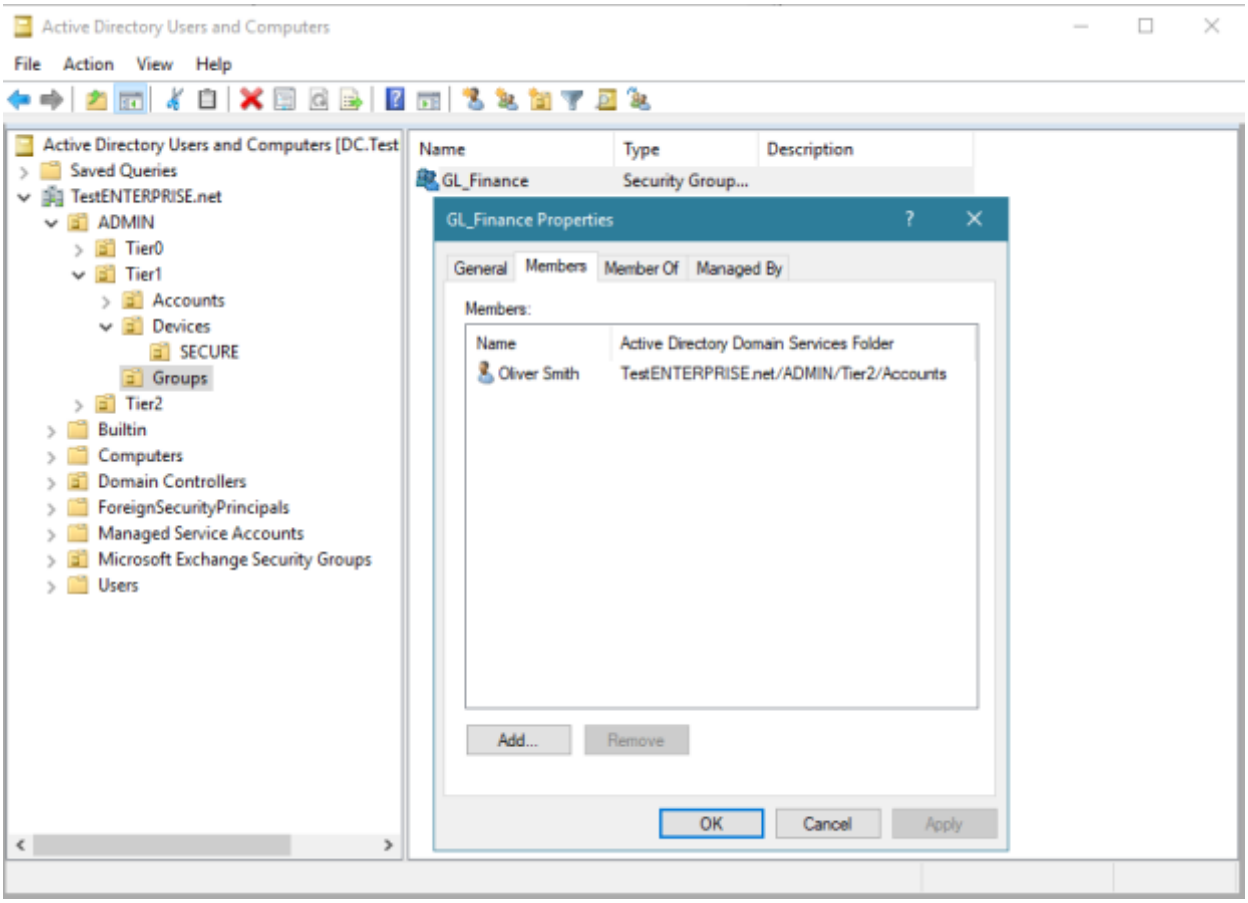
Frame Number	Time Date Local Adjusted	Time Offset	Process Name	Source	Destination	Protocol Name	Description
3	3:19:50 AM 10/25/2019	0.4416853		10.1.1.51	10.1.1.21	ICMP	ICMP:Echo Request Message, From 10.1.1.51 To 10.1.1.21
4	3:19:50 AM 10/25/2019	0.4435794		10.1.1.51	10.1.1.21	AuthIP	AuthIP:version 1.0, Quick Mode, responder, Payloads = HDR*, CRYPTO, Flags
5	3:19:50 AM 10/25/2019	0.4449920		10.1.1.21	10.1.1.51	AuthIP	AuthIP:version 1.0, Quick Mode, responder, Payloads = HDR*, CRYPTO, Flags
6	3:19:50 AM 10/25/2019	0.4454428		10.1.1.51	10.1.1.21	AuthIP	AuthIP:version 1.0, Quick Mode, responder, Payloads = HDR*, CRYPTO, Flags
7	3:19:50 AM 10/25/2019	0.4462120		10.1.1.21	10.1.1.51	AuthIP	AuthIP:version 1.0, Quick Mode, responder, Payloads = HDR*, CRYPTO, Flags
8	3:19:50 AM 10/25/2019	0.4465319		10.1.1.51	10.1.1.21	ESP	ESP:SPI = 0x535171ee, Seq = 0x1
9	3:19:50 AM 10/25/2019	0.4468999		10.1.1.21	10.1.1.51	ESP	ESP:SPI = 0x2b1f3c00, Seq = 0x1
11	3:19:51 AM 10/25/2019	1.4470345		10.1.1.51	10.1.1.21	ESP	ESP:SPI = 0x535171ee, Seq = 0x2
12	3:19:51 AM 10/25/2019	1.4476910		10.1.1.21	10.1.1.51	ESP	ESP:SPI = 0x2b1f3c00, Seq = 0x2
15	3:19:52 AM 10/25/2019	2.4512452		10.1.1.51	10.1.1.21	ESP	ESP:SPI = 0x535171ee, Seq = 0x3
16	3:19:52 AM 10/25/2019	2.4518935		10.1.1.21	10.1.1.51	ESP	ESP:SPI = 0x2b1f3c00, Seq = 0x3
18	3:19:53 AM 10/25/2019	3.4723293		10.1.1.51	10.1.1.21	ESP	ESP:SPI = 0x535171ee, Seq = 0x4
19	3:19:53 AM 10/25/2019	3.4728609		10.1.1.21	10.1.1.51	ESP	ESP:SPI = 0x2b1f3c00, Seq = 0x4

Test 1-2: On Client1 – Connecting to SQL Server

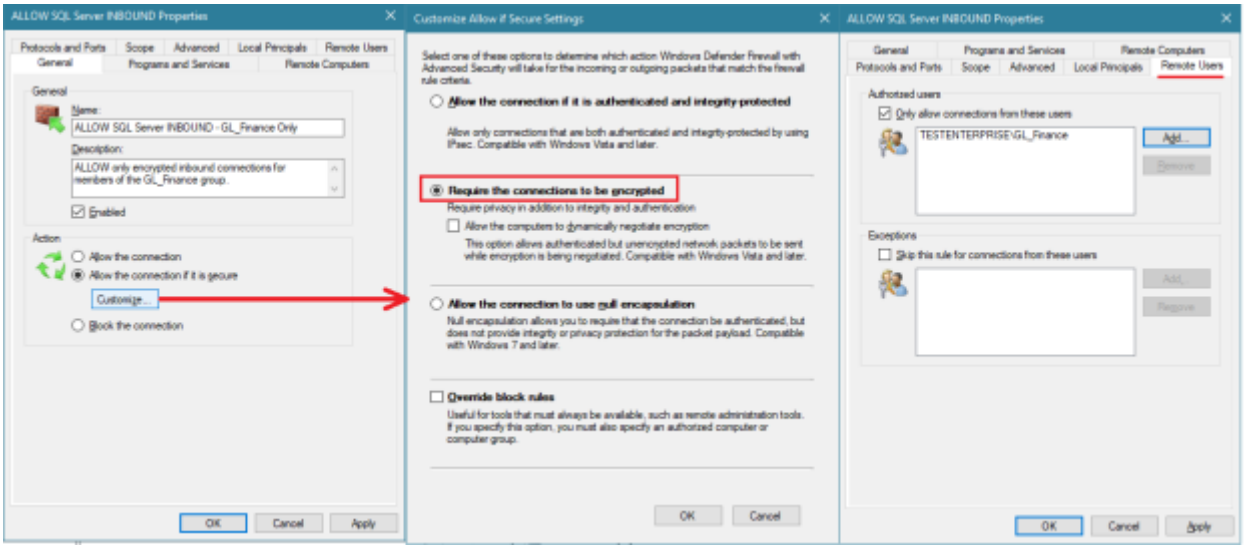


So far so good. Now we can proceed to configuring the additional security layer – restricting access to the secure server to the specific users or/and computers only. Suppose our goal is to permit only the members of the **GL_Finance** global group to connect to the SQL Server – to fulfill this requirement the following steps must be taken:

1) Create the group (if not already exists) and populate it with user accounts eligible to connect to the server:

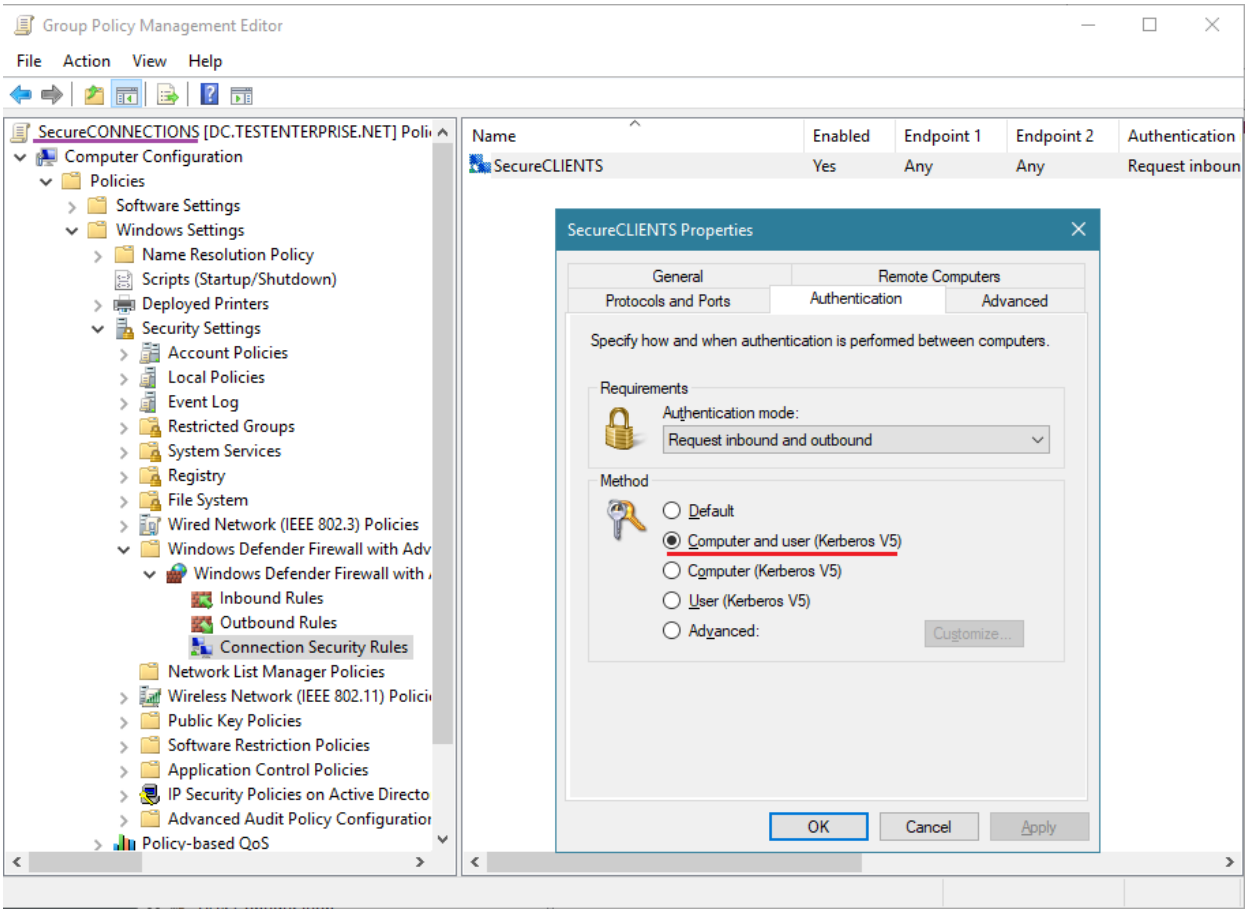


Advertisements
Report this adPrivacy
2) Modify the **ALLOW SQL Server INBOUND** rule as follows:



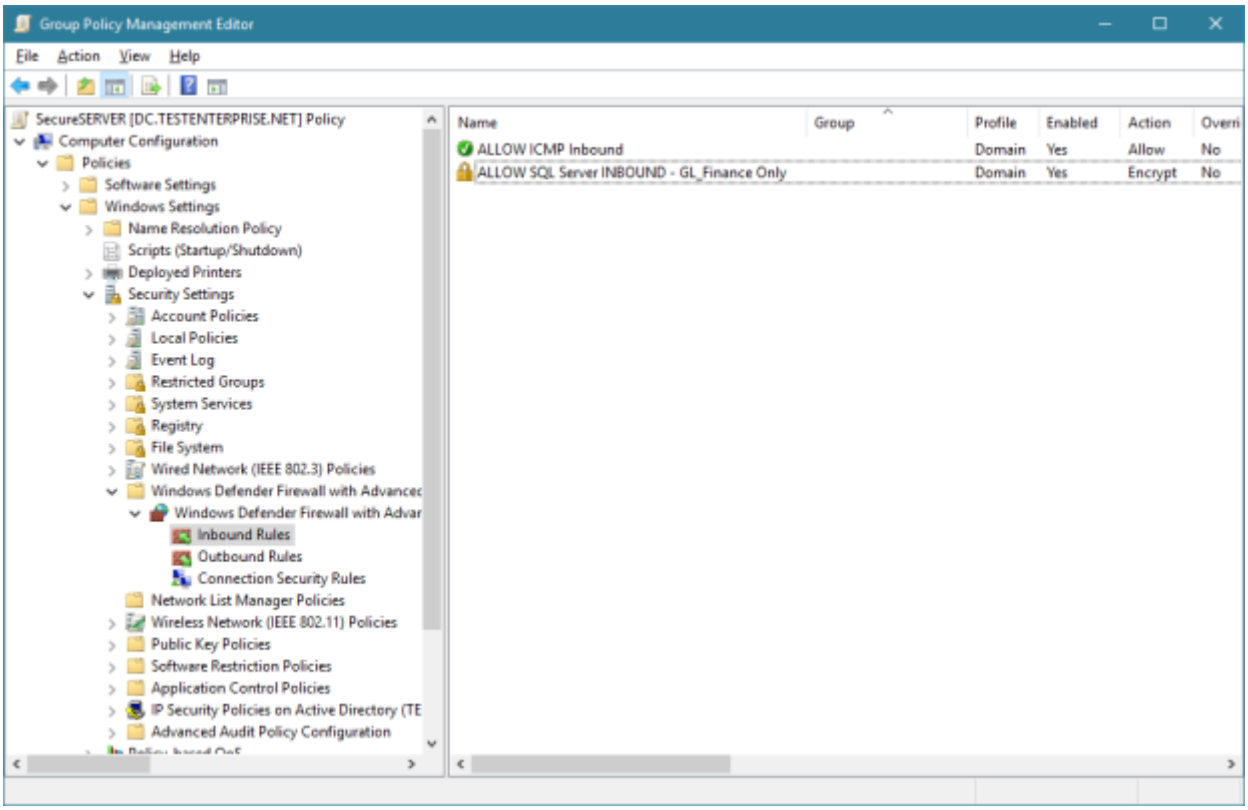
Select *Allow the connection if it secure*, then *Require the connections to be encrypted* and add the required user group.

Similarly, the Remote Computers tab may be used to restrict access to a certain computer group(s) – either alone or together with the Remote Users tab. The ability to use the user and/or computer authentication in Windows Firewall rules depends on the security rule's authentication type as mentioned earlier:



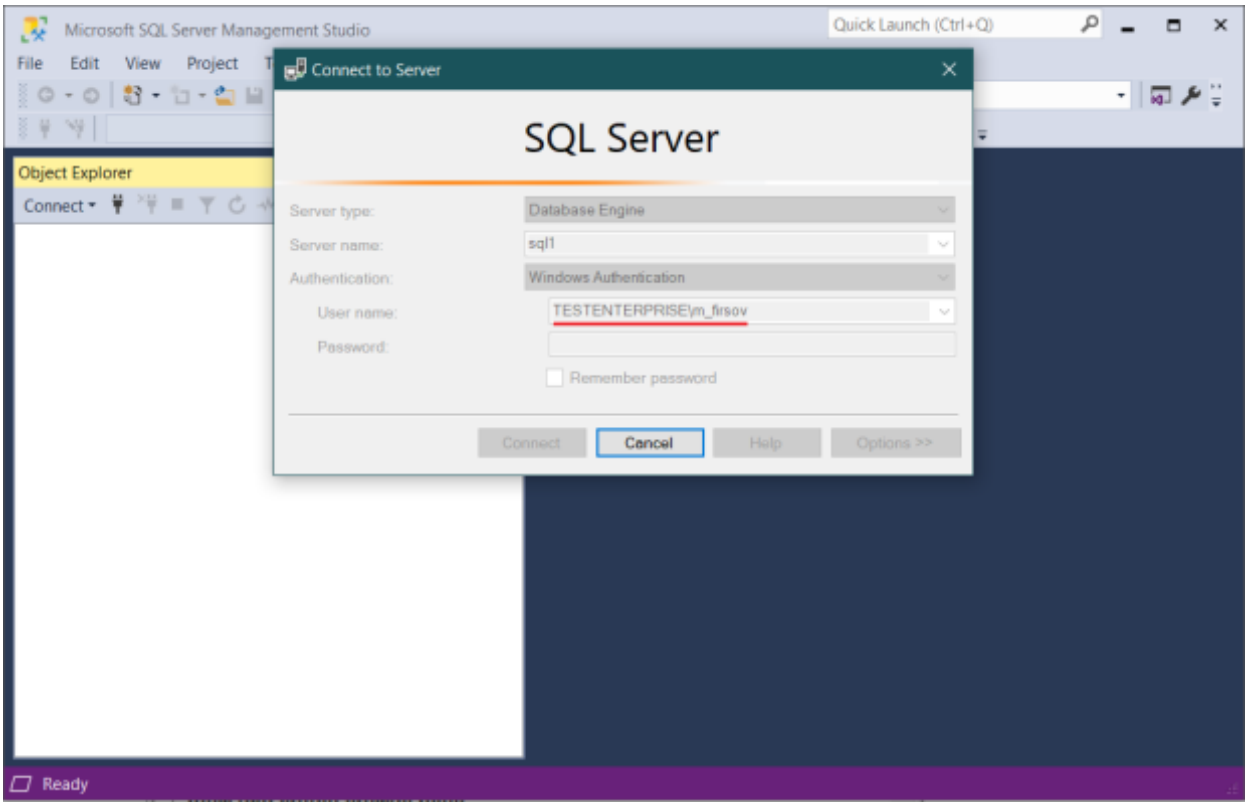
If, for example, I selected here the **Computer (Kerberos V5)** authentication type instead of the default **Computer and user (Kerberos V5)** I'd be left with the single option to use the Remote Computers tab. Access restriction in this case would be based on the client computer account(s) rather than user account(s).

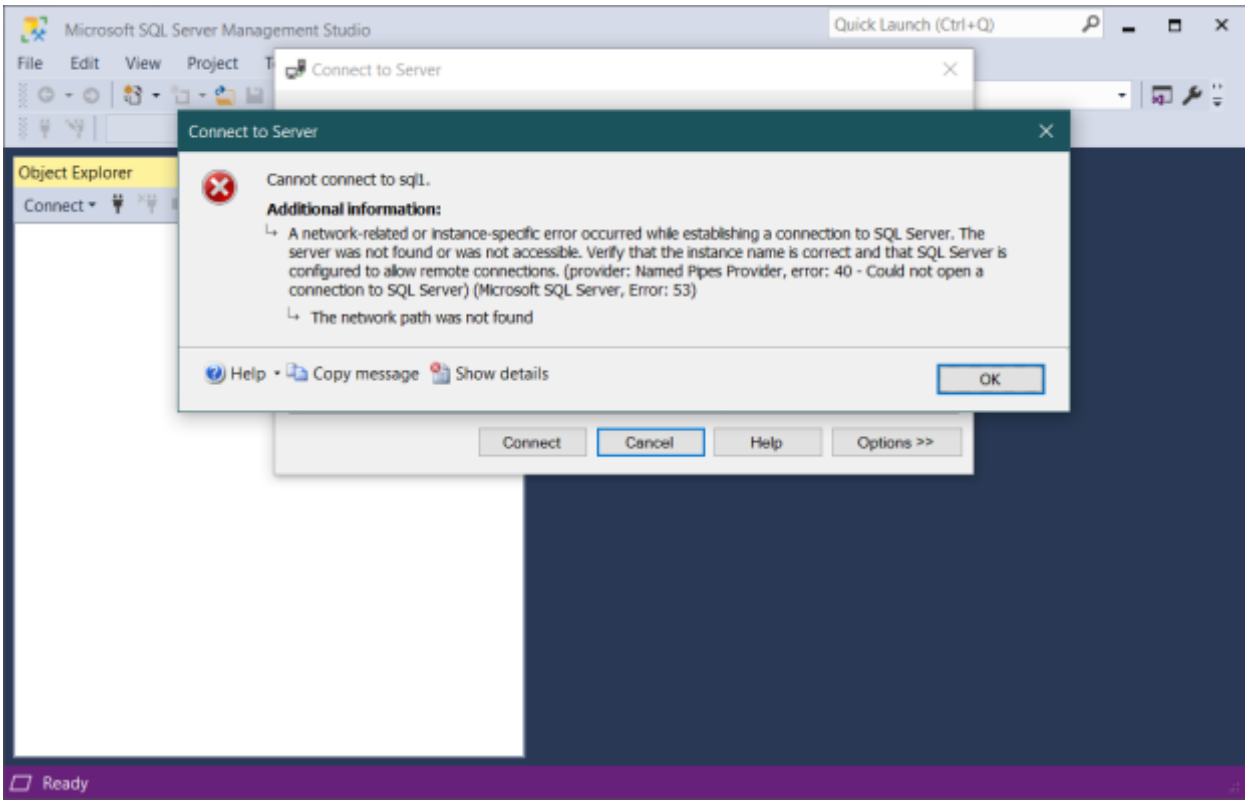
Once this modification is completed, the ALLOW SQL Server INBOUND rule will change its icon to the yellow lock:



Now let's test this rule – for this I'll first repeat the Test 1.2 and once again try to connect to SQL1 as Michael Firsov (recall that Michael Firsov is not a member of the GL_Finance group) and then access SQL1 as Oliver Smith who is a member of the GL_Finance group.

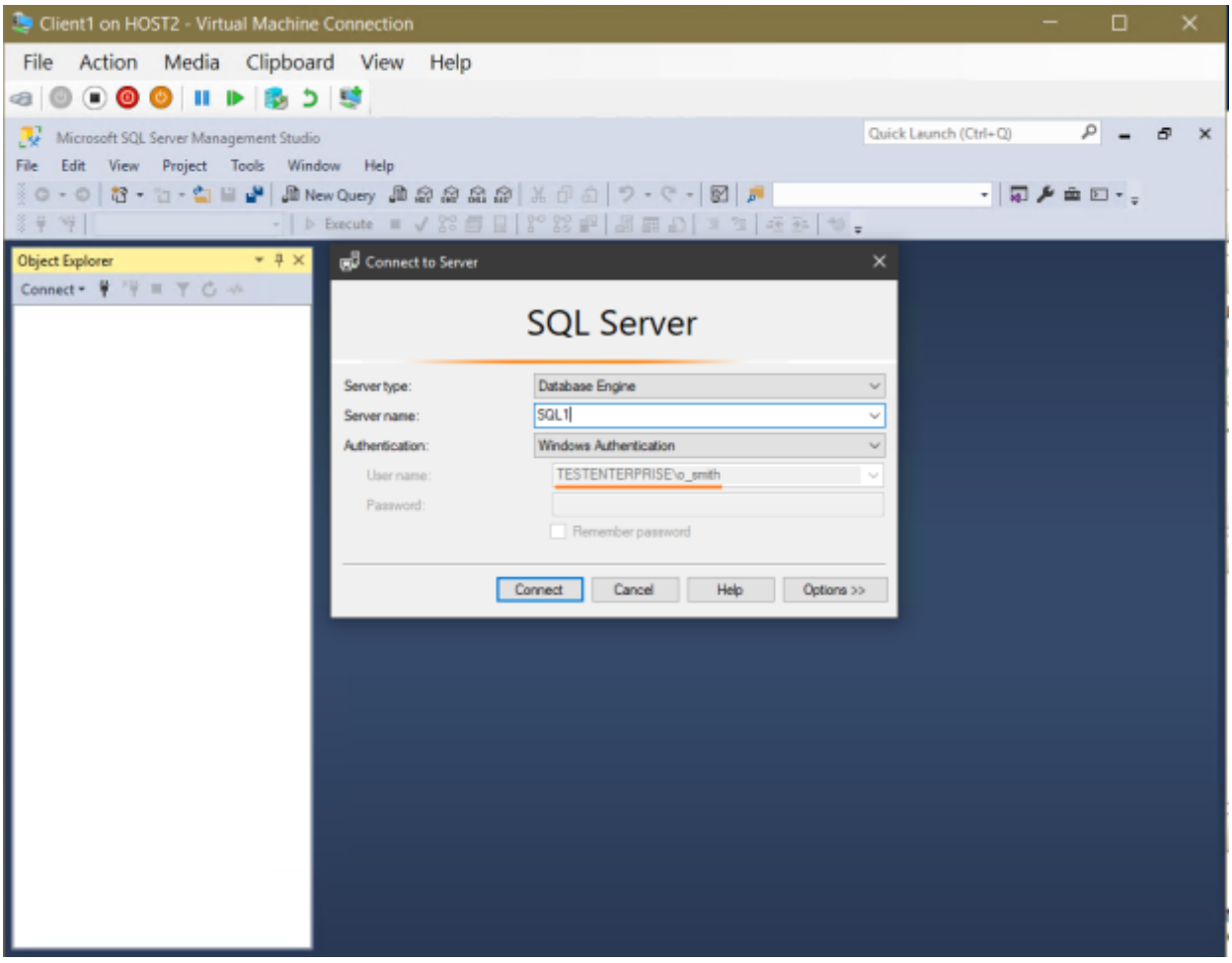
Test 2.1: On Client 1 – Connecting as Michael Firsov

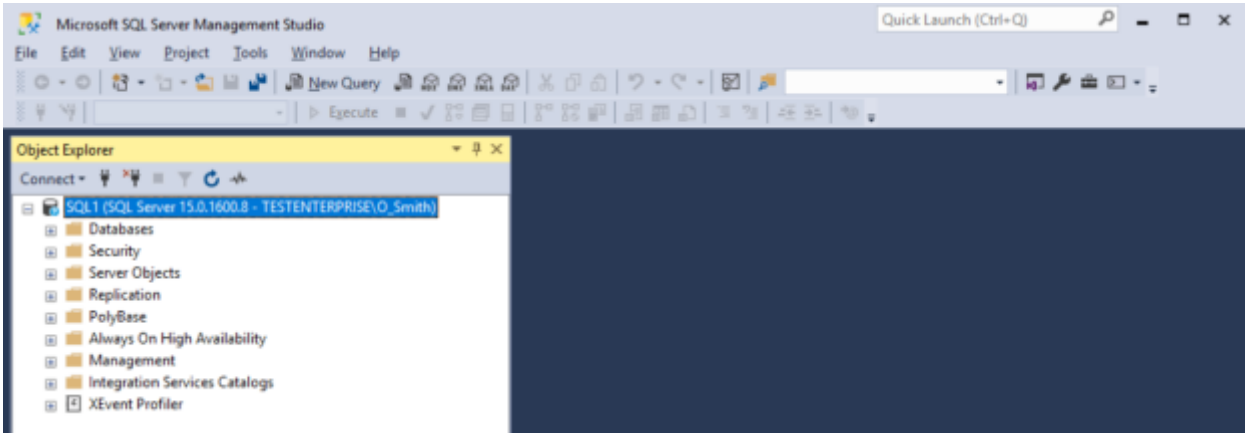




This behaviour is by design as Michael Firsov is not a member of the GL_Finance group.

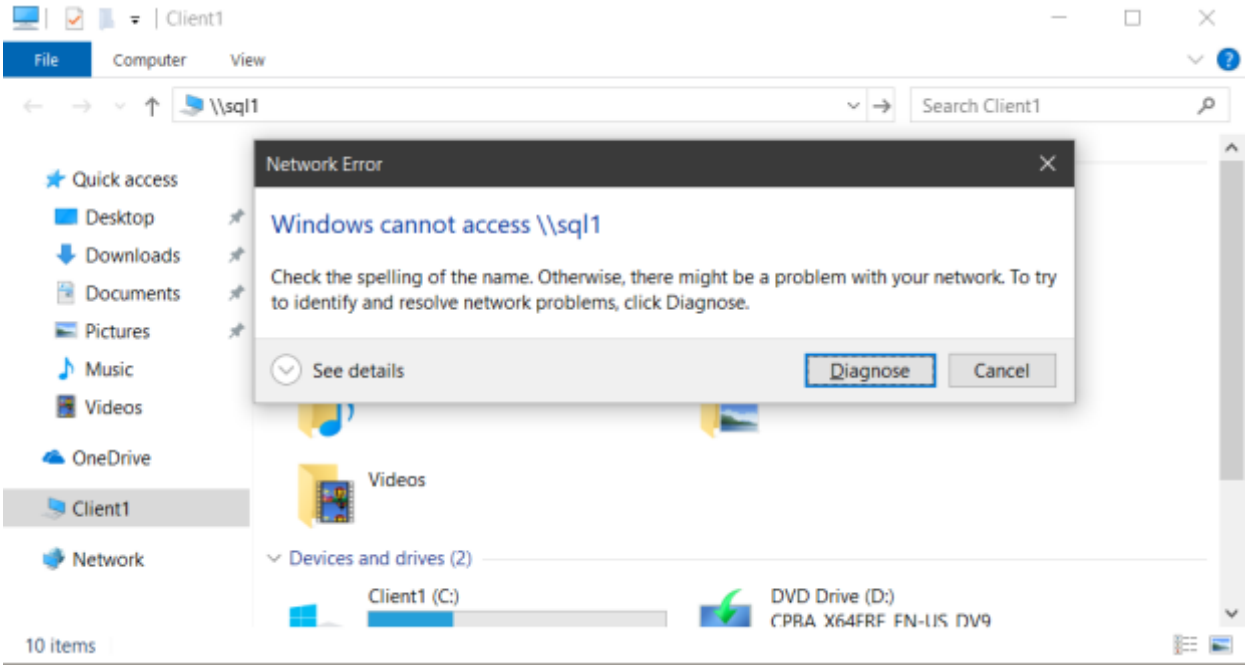
Test 2-2: On Client 1 – Connecting as Oliver Smith



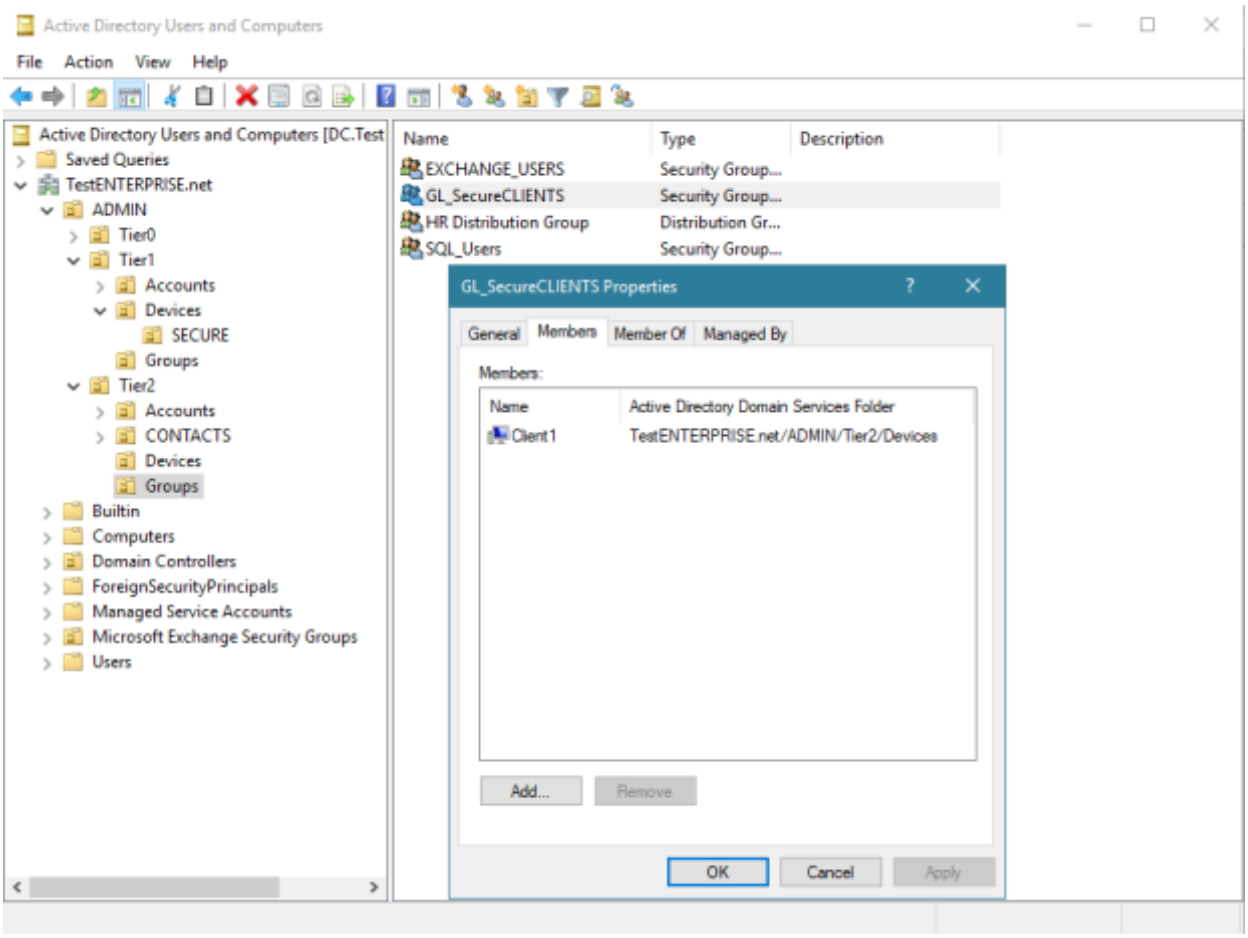


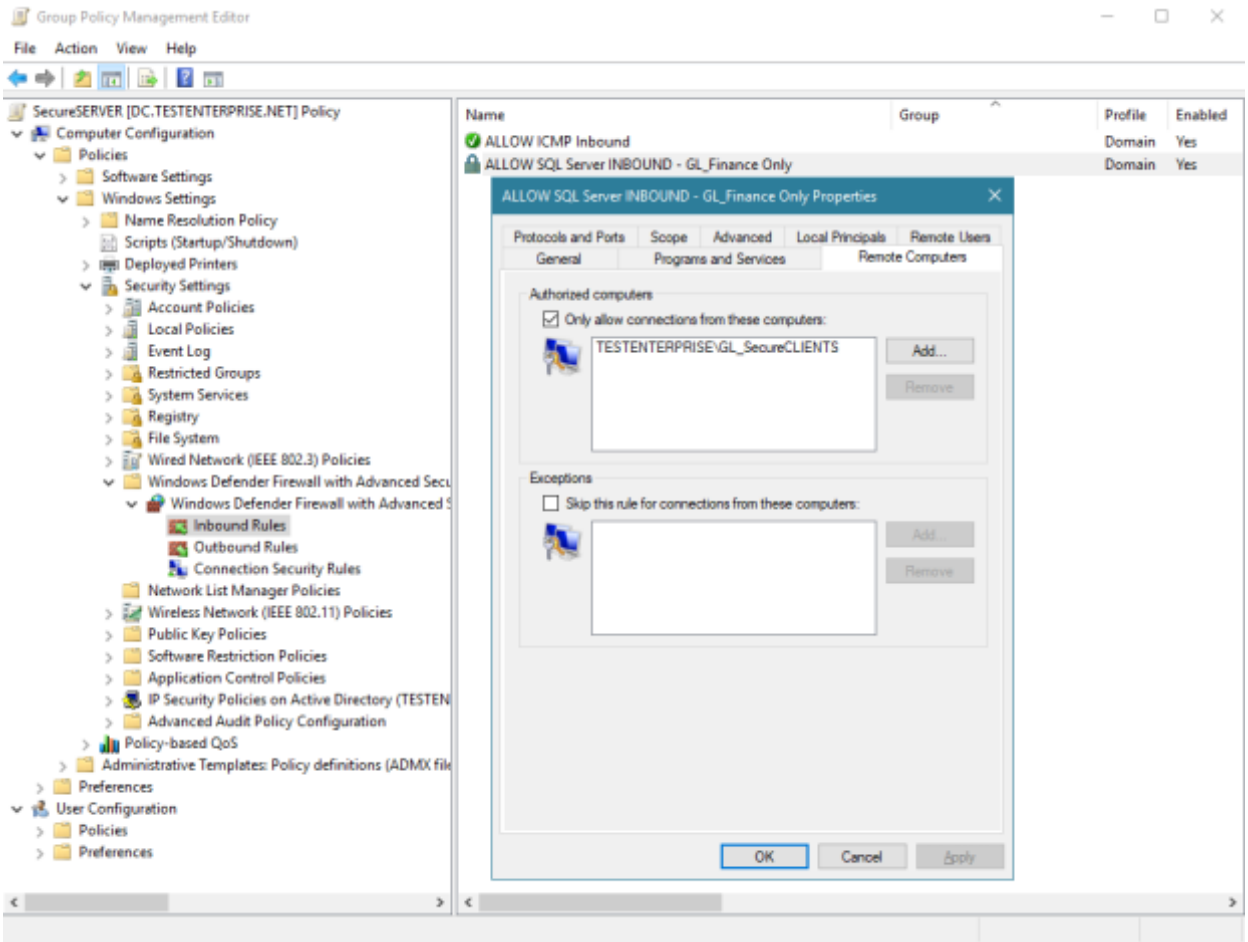
As you see the rule works as expected – currently only Oliver Smith may connect to the SQL Server from any domain-joined computer. Even Oliver Smith will not be able to connect to the server from a client that does not have the required connection security rules applied.

All other access to SQL1 (except icmp) is blocked:



If for some reason members of the GL_Finance global group should be bound to use specific computers to connect to SQL1 we can further restrict the **ALLOW SQL Server INBOUND** rule. Suppose only computers which computer accounts are members of the GL_SecureClients global group are allowed to access the server – in this case make sure the client computer account is in the OU for which the connecton securty rules are being applied, add the required computer account(s) to the group and add this group to the Remote computers tab:

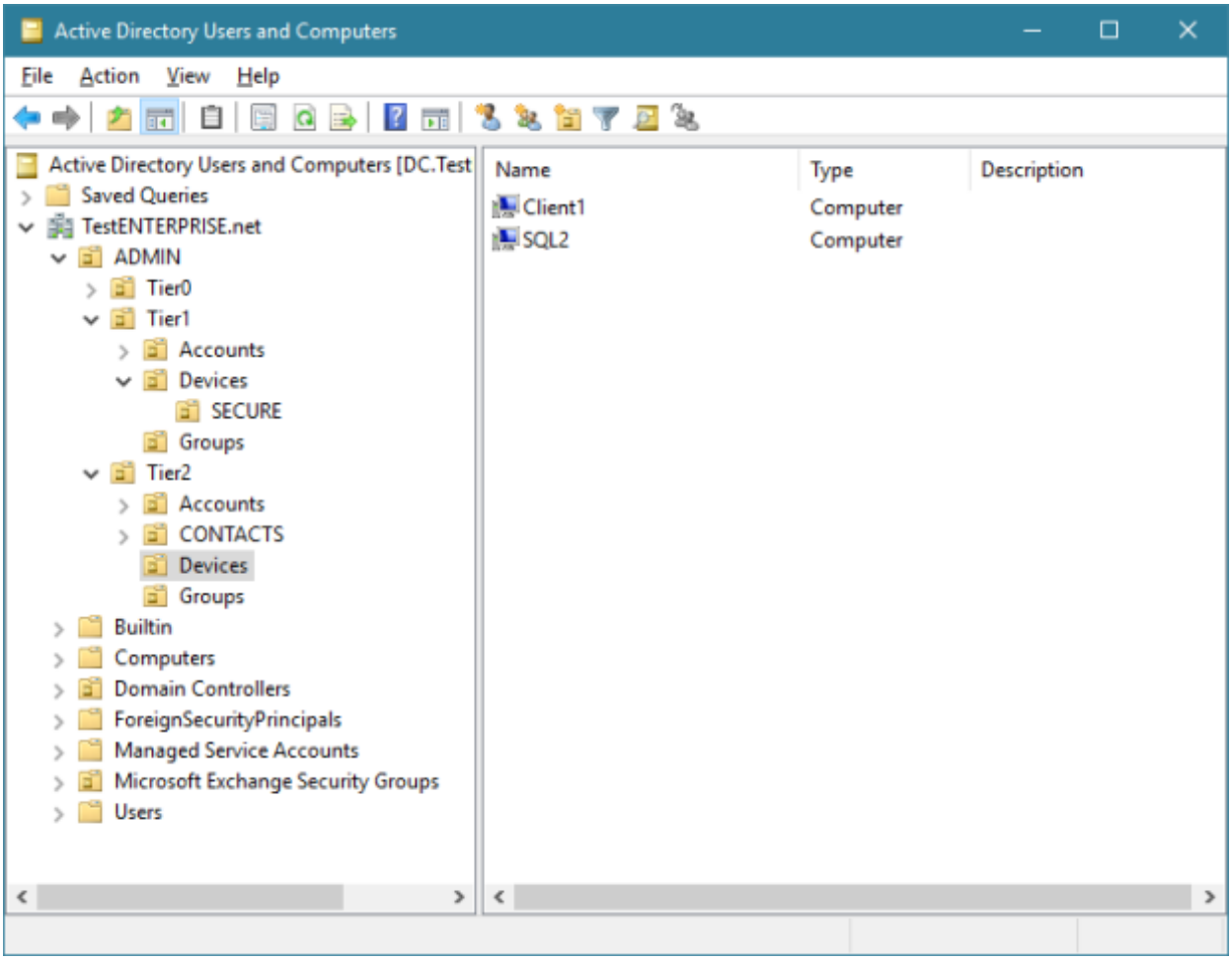




Advertisements

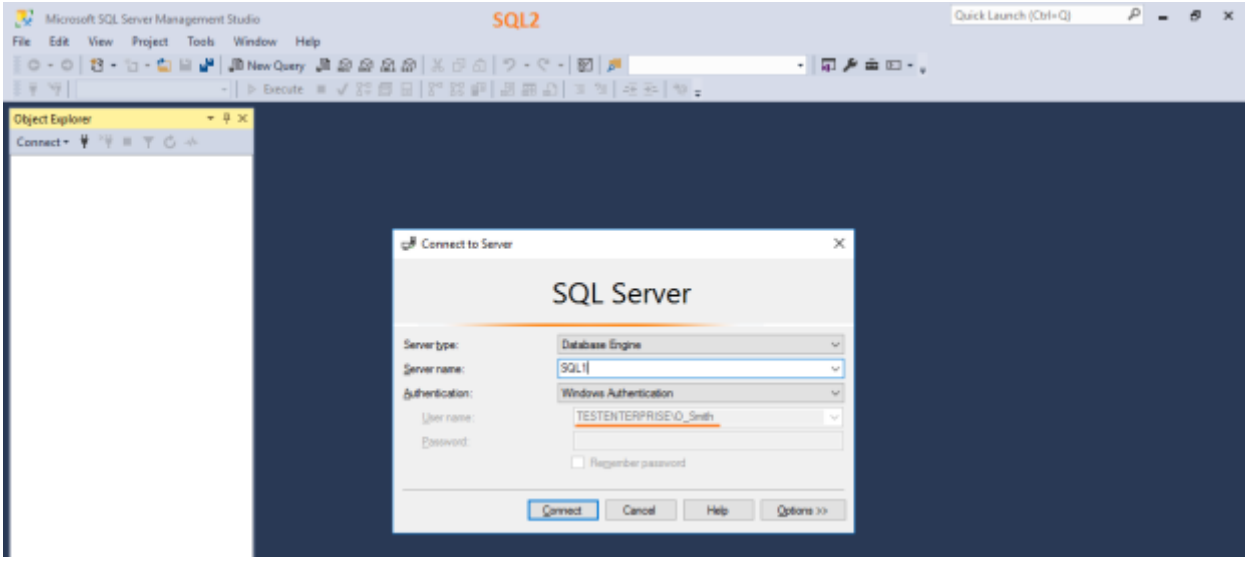
This time Oliver Smith (or any member of the GL_Finance group) will be able to connect to SQL1 only from the computers that are members of the GL_SecureClients group (currently only from Client1).

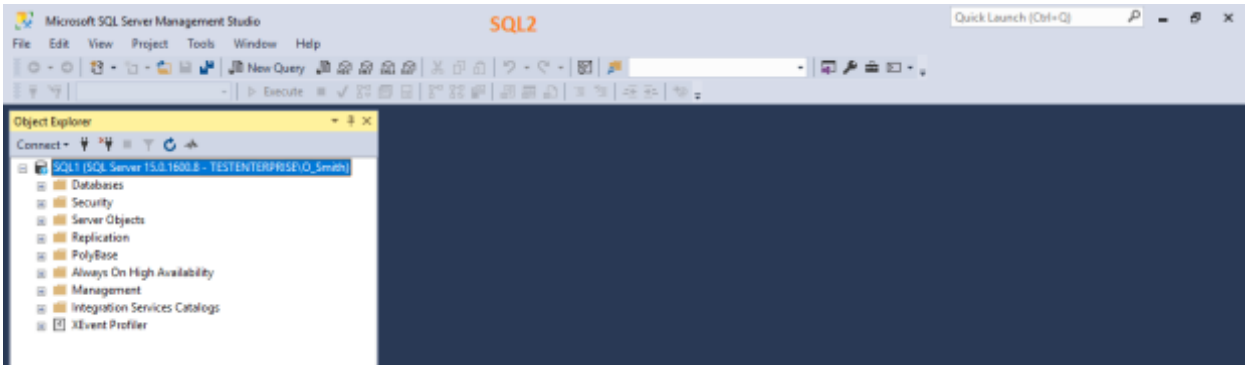
I've moved the SQL2 computer account to the Devices OU which has the SecureCONNECTIONS gpo applied…



…and made a couple of the new test connections:

a) connecting to SQL1 from SQL2 as Oliver Smith before modifying a rule:

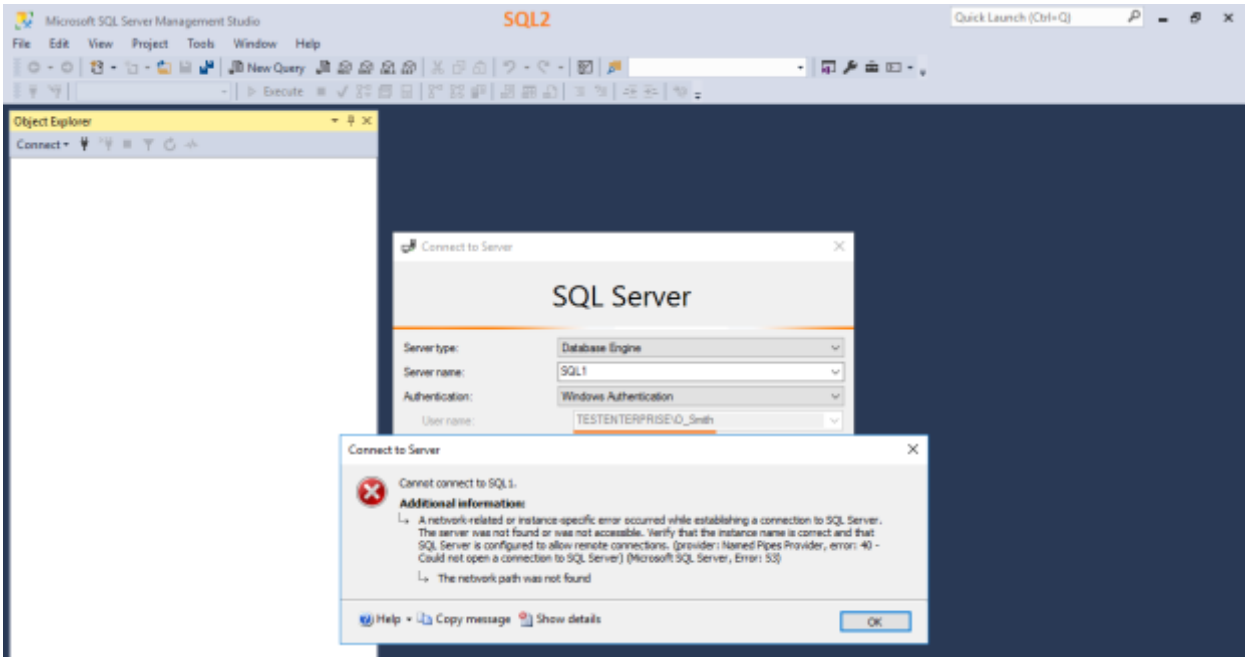




Advertisements

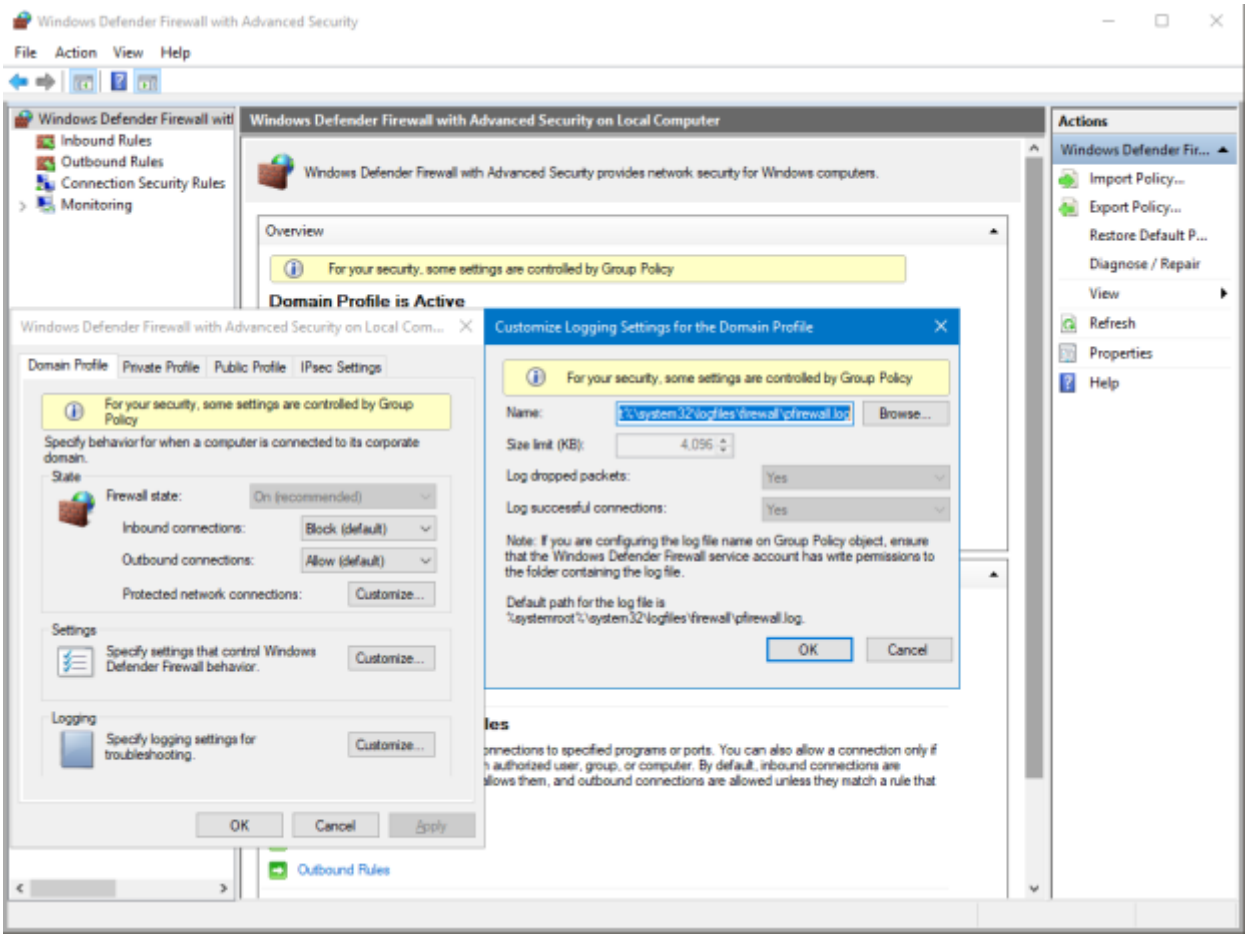
Report this adPrivacy

b) …and after the rule modification:

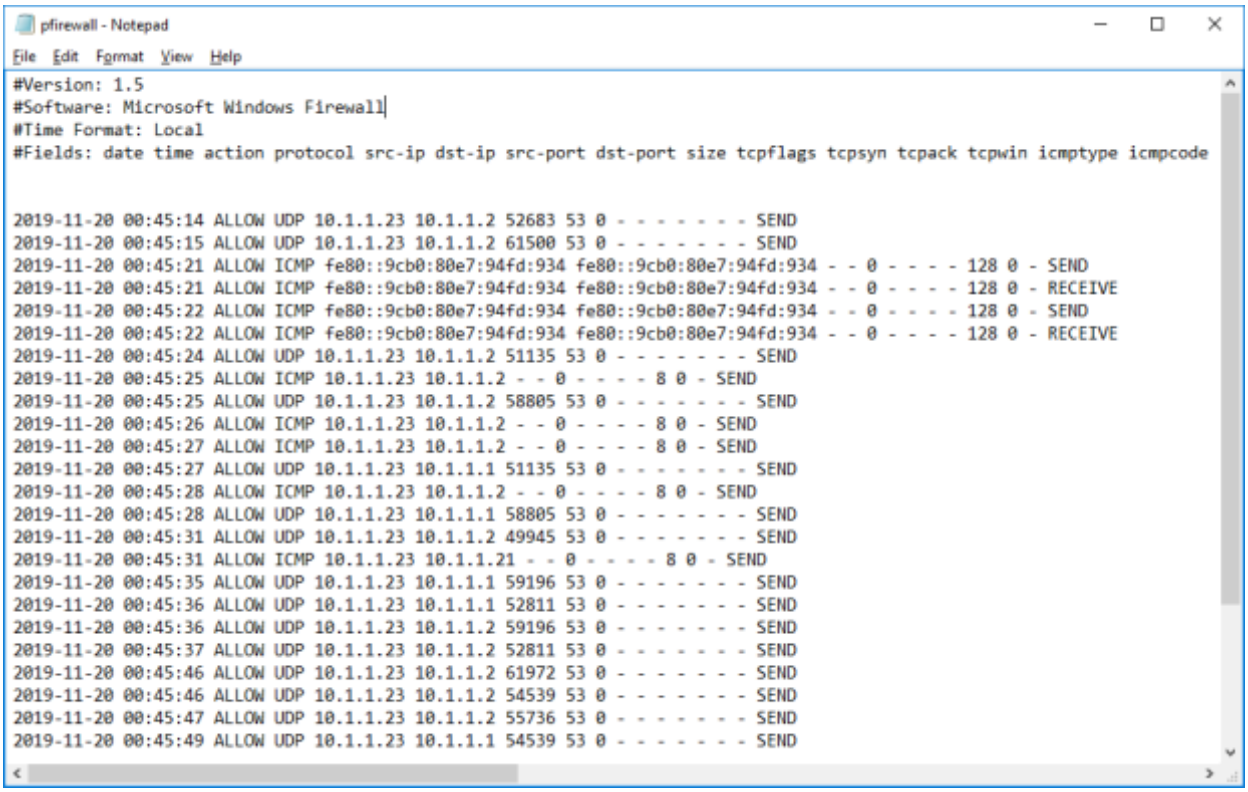


As you see when Oliver Smith accesses SQL1 from the computer which is not a member of the GL_SecureClients global group the connection is dropped.

* – *5) logging for dropped and allowed packets is turned on* – during firewall configuration administrators are supposed to use the firewall log which resides here: %systemroot%system32logfilesfirewallpfirewall.log, but I wouldn't count on it too much as this log may 1) not get created at all 2) not get populated with log records, consisting only of the field names 3) stop adding new records – and all of this can happen in spite of the fact that all firewall-related settings get applied successfully:



Here's the example of the log:



Lot's of complaints can be seen on technet regarding this issue and I myself has contacted MS on the matter so there may be some improvements over time.

Summary:

In this blog post series we've seen how IPSec can be used for securing servers that contain highly sensitive data – either by leveraging the connection security rules alone or the connection security rules in conjunction with Windows Defender Firewall rules.