Knowledge Base Article

KB01154 - 排除 PI System 连接故障

Product: PI AF / PI Data Archive
Version(s): All

中文(简体) ▼

（2016 年 12 月）本文提及了现已被更安全的身份验证形式取代的 PI Trust。AL00309 讨论了从 PI Trust 向 PI API 的 Windows 集成安全性 (WIS) 的转移。

问题

用户难以将客户端/接口连接到 PI System。解决此问题的最佳方法是什么?

背景

要了解 PI System 连接的更多背景，请查阅关于 Troubleshooting PI client connection problems（排除 PI 客户端连接问题）的白皮书。

解决方案

有多种问题可导致无法连接到 PI System。在解决问题之前，知道客户端的以下信息会很有用:

1. 应用程序基于哪个编程层（PI API、PI SDK、AF SDK）？
2. 初始化连接时，多长时间之后会显示故障（立即、约 10 秒、更长时间）？
3. 预期使用哪种身份验证（WIS、Trust、显式登录）？

以下摘要列出了症状以及采取哪些步骤来确定问题。

网络中的所有计算机都无法连接到服务器

1. 网络路径是否正确解析?
2. 本地 DNS 缓存是否停滞?
3. 是否存在已配置的防火墙?
4. 端口是否打开?
5. PI Server 是否正在检测传入连接?

网络中的某些计算机可以连接到服务器，但其他计算机无法连接

1. 网络路径是否正确解析?
2. 本地 DNS 缓存是否停滞?
3. 是否存在已配置的防火墙?
4. 有什么方法可以知道在哪里配置客户端与服务器之间的防火墙?
5. PI Server 是否正在检测传入连接?
6. PI Firewall Table 中是否存在一个条目?

应用程序无法连接到服务器

1. 网络路径是否正确解析?
2. 本地 DNS 缓存是否停滞?
3. PI Server 是否正在检测传入连接?

4. 应用程序是否受到监听?

应用程序连接速度很慢，或在连接过程中超时

1. PI Server 是否正在检测传入连接?
2. 应用程序是否配置为使用 Trust?

应用程序未使用预期的 PI Mapping

1. 应用程序是否配置为使用 WIS?
2. 应用程序使用 NTLM 还是 Kerberos 进行身份验证?
3. 映射是否使用 Windows 组?
4. 映射是否使用 BUILTIN\Administrators 组?

应用程序未使用预期的 PI Trust

1. 应用程序是否配置为使用 Trust?
2. 应用程序是否基于 PI API?
3. 应用程序基于 AF 还是 PI SDK?
4. 应用程序是否与多个 Trust 匹配?

网络路径是否正确解析?

客户端可能由于名称解析不正确而无法连接。在客户端打开服务器的 TCP/IP 连接之前，必须先将传递的主机名或 fqdn 解析为 IP 地址。此操作使用反向名称查找功能来完成。Windows 通过按顺序检查以下内容执行查找，直至找到匹配项：

1. 本地 DNS 缓存
2. 主机文件 (%WINDIR%\System32\Drivers\etc\hosts)
3. DNS
4. NETBIOS 缓存。

为名称解析测试选择正确的网络路径非常重要。

- 如果您正在排除接口连接故障，请使用接口批次文件（/host 参数）中指定的网络路径。

```
"C:\Program Files (x86)\PIPC\Interfaces\PIPerfMon\PIPerfMon.exe" 1 /PS=PERF /ID=4 /host=PIServer:5450 /maxstoptime=120 /perf=8 /f=00:00:30
```

- 对于 PI SDK 连接，请使用“已知服务器”表中列出的 PI Server 的网络路径。
- 对于 AF SDK 连接，请使用 AF Server 属性中列出的 AF Server 的主机属性。

用户常常使用 nslookup 测试名称解析。这不是彻底的测试，因为 nslookup 仅检查 DNS，而且其行为方式不同于客户端应用程序。一种更好的测试是使用遵循正常查找规则的客户端。例如，您可以使用 **ping** 执行名称查找：

.code C:\>ping PIServerNameOrFQDN Pinging PIServer.domain.int 10.0.0.1 with 32 bytes of data: Reply from 10.0.0.1: bytes=32 time<1ms TTL=128 Reply from 10.0.0.1: bytes=32 time<1ms TTL=128 Reply from 10.0.0.1: bytes=32 time<1ms TTL=128 Reply from 10.0.0.1: bytes=32 time<1ms TTL=128 Ping statistics for 10.0.0.1: Packets: Sent = 4, Received = 4, Lost = 0 (0% loss), Approximate round trip times in milli-seconds: Minimum = 0ms, Maximum = 0ms, Average = 0ms .code

这种情况下，由于您仅对命令输出开头列出的名称解析感兴趣，因此 ping 实际上是否工作都无关紧要（因为 IT 部门常常会阻止 ping）。另一种完成此操作的方法是使用 PowerShell。

```
PS C:\> [System.Net.Dns]::GetHostAddresses(“PIServerNameOrFQDN”)   Address           : 2048002058 AddressFamily     : InterNetwor
```

本地 DNS 缓存是否停滞?

如果名称解析看起来不正确，请检查客户端上的本地 DNS 缓存。

```
C:\>ipconfig /displaydns Windows IP Configuration PIServer.domain.int ---------------------------------------- Record Name . . . . . : PIServer.domain.int Record Type . . . . . : 1 Time To Live . . . . : 1135 Data Length . . . . . : 4 Section . . . . . : Answer A (Host) Record . . . : 10.0.0.1
```

条目应自动超龄（它们保留在缓存中的默认时间可以改变），但如果您想要强制清理本地 DNS 缓存，可以从命令行执行此操作。

```
C:\>ipconfig /flushdns Windows IP Configuration Successfully flushed the DNS Resolver Cache.
```

NETBIOS 缓存是否停滞?

如果未通过本地 DNS 缓存、主机文件或从 DNS 服务器解析名称，则 NETBIOS 解析可以用作备用选项。请您在完全确认来自前三个选项的条目无法解析时 **才** 选中此项。

```
C:\WINDOWS\system32>nbtstat -c vEthernet (machine.domain.int): Node IpAddress: [10.35.34.152] Scope Id: [] NetBIOS Remote Cache Nam
e Table Name Type Host Address Life [sec] ---------------------------------------------------------------- POLARIS <20> UNIQUE 10.35.3
4.108 181 Ethernet: Node IpAddress: [0.0.0.0] Scope Id: [] No names in cache
```

与本地 DNS 缓存类似，条目会超龄。为了清除缓存，您可以运行“nbtstat -r”。 **建议您修复 DNS/Hostsfile**，而不要触碰 NETBIOS 缓存，因为它会被用作最后的手段。

是否存在已配置的防火墙?

PortQuery 和 **telnet** 等工具可以帮助您确定客户端是否可以打开与服务器上特定端口的网络连接。例如，要使用 telnet 测试与 PI Server 的网络连接，您将运行：

```
C:\>telnet PIServerNameOrIP 5450
```

对于 AF Server，命令将是：

```
C:\>telnet AFServerNameOrIP 5457
```

注意：如果 telnet 成功，您应在运行命令后返回空命令提示。这是因为 PI 和 AF Server 均没有 telnet 客户端的 shell。

但是，在 Windows Server 2008 中，默认禁用 telnet 客户端。这使得使用 telnet 解决连接问题变得很困难，尤其是在用户无权启用 Windows 中的功能时。幸运的是，可以通过 PowerShell 获得一种替代方法。

```
PS C:\> try { $tcp = New-Object "System.Net.Sockets.TcpClient"; $tcp.connect("PIServerNameOrIP",5450); $tcp.close( ); "Success";
```

请注意，如果端口 5450 上没有监听程序（例如，如果您在安装 PI Data Archive 之前测试连接性），上述代码将显示错误。这时，您可以在目标计算机上使用以下命令打开监听程序。

```
PS C:\> $listener = [system.net.sockets.tcplistener]5450; $listener.start()
```

要关闭监听程序，可以关闭 PowerShell 窗口或使用以下命令。

```
PS C:\> $listener.stop();
```

如果使用此方法测试连接性，请确保在启动 PI Data Archive 之前关闭监听程序。每个端口上任何时候都只能有一个 TCP 监听程序，因此如果在未关闭 Powershell 监听程序的情况下启动 PI Data Archive，则将无法创建其自己的监听程序，这会导致连接问题。如果已在计算机上安装 PI Data Archive，最佳选项是使用以下部分介绍的方法： PI Server 是否正在检测传入连接?

有关其他所需端口的列表，请参阅 KB2820OSI8、 KB00751、 KB00768 和 KB00944。

有什么方法可以知道在哪里配置客户端与服务器之间的防火墙?

pathping 等工具可以提供网络中客户端与服务器之间的路径以及网络的可靠性。pathping 的输出与以下内容类似：

```
C:\>pathping computer2 Tracing route to computer2.domain.int [10.8.1.1] over a maximum of 30 hops: 0 computer1.domain.int [10.8.1.
1] 1 10.8.1.254 2 10.8.254.48 3 192.168.25.250 4 computer2.domain.int [192.168.1.1] Computing statistics for 100 seconds... Source
to Here This Node/Link Hop RTT Lost/Sent = Pct Lost/Sent = Pct Address 0 computer1.domain.int [10.8.1.1] 0/ 100 = 0% | 1 2ms 0/ 100
= 0% 0/ 100 = 0% 10.8.1.254 0/ 100 = 0% | 2 0ms 0/ 100 = 0% 0/ 100 = 0% 10.8.254.77 0/ 100 = 0% | 3 0ms 0/ 100 = 0% 0/ 100 = 0% 19
2.168.25.250 0/ 100 = 0% | 4 0ms 0/ 100 = 0% 0/ 100 = 0% computer2.domain.int [192.168.1.1]
```

第一部分提供跟踪路由的结果并将为您提供连接的通行路径。

第二部分描述通过往返时间 (RTT) 测量的延迟以及数据包丢失情况。理想情况下，RTT 会很低，尤其是在内部网络中时。对于路径的每个部分，数据包丢失率应为 0（如此处所示）。如果存在任何非零数值，则表示有问题。

如果您看到数据包丢失率为 100%，则很可能存在阻止所有 ping 通信的规则。这不意味着两个节点无法连接；在防火墙上设置某些规则以允许一种通信，同时阻止另一种通信，这种情况很常见。您还可以借此知道防火墙上是否设置了阻止通信的规则。如果数据包丢失率大于 0%，也会影响 PI System 的运行。例如，PI 接口可能无法从 PI Data Archive 加载 Tag 信息并将无法正确启动。

端口是否打开?

在服务器应用程序能够接受传入网络通信之前，必须先打开端口。通常，此操作在启动时进行。您可以从命令行使用 **netstat -a**。例如，要确定 PI Server 是否已成功打开端口 5450，可以运行以下内容：

```
C:\>netstat -a | find "LISTENING" | find "5450" TCP 0.0.0.0:5450 PIServer:0 LISTENING TCP [::]:5450 PIServer:0 LISTENING
```

对于 AF Server，命令将是：

```
C:\>netstat -a | find "LISTENING" | find "5457" TCP 0.0.0.0:5457 AFServer:0 LISTENING TCP [::]:5457 AFServer:0 LISTENING
```

您也可以使用 PowerShell 执行该检查：

```
1 PS C:\> netstat -a | where { $_ -like "*5450*listening*" }   TCP    0.0.0.0:5450           PIServer:0             LISTENING   TCP
```

注意：在上述示例中，netstat 输出中显示两个监听程序。这是因为服务器将为 IPv4 和 IPv6 各使用一个监听程序。

PI Server 是否正在检测传入连接?

PI Network Manager 在检测传入 tcp 连接时会向日志中写入一条消息：

```
D 23-Feb-12 16:40:11 pinetmgr (7004)
>> PInet accepted TCP/IP connection, cnxn ID 205 Hostname: , 10.0.0.2: 14108
```

如果消息日志中未出现此消息，可能表示网络存在问题（例如防火墙阻止通信）。

PI Firewall Table 中是否存在一个条目?

如果 PI Server 由于 PI Firewall Table 中的一个条目而阻止连接，日志中将包含一条指示该情况的消息，如以下示例所示：

```
I 14-Nov-13 10:25:40 pinetmgr (7003)
>> PInet: Firewall blocked TCP/IP connection, hostname: , 10.0.0.2:41326
```

应用程序是否获得许可?

接受连接时，PI Network Manager 验证该客户端是否为获许可的应用程序。如果许可证检查成功，您将在消息日志中看到以下内容：

```
I 23-Feb-12 16:40:11 pinetmgr (7039)
>> Connection accepted: Process name: piartool(1060):remote(1060) ID: 205
```

应用程序是否配置为使用 WIS?

为了进行测试，您可以在 AboutPI-SDK/PISDKUtility Connections（AboutPI-SDK/PISDKUtility 连接）-> Options（选项）中更改 SDK 协议顺序并移除除“Windows 安全性”外的所有协议。如果您使用的是 PI SDK 1.4 及更高版本，还可以在客户端中使用 %PIHOME% 目录中的 piartool 将身份验证方法限制为仅“Windows 安全性”。

```
C:\Program Files\PIPC\ADM>piartool -node PIServer -windows -authdebug -block pibasess -verbose Debug Authentication: PIarray: [$Wor
kfile: pitmplar.hxx $ $Revision: 60 $]:: _msize> 1 _mmaxsize> 8 _mincr> 8 _ChunkCount> 1 0 ProtocolType: Authprototype_msoft_sspi P
rotocol Result Status: ResultSuccess PIstatus returned: [0] Success ProtoCredentialresult: DOMAIN\PIGuy ProtoDetails: PIOperators |
PIWorld Subsystem pibasess query returned: [0] Success - Time (Sec):0.153
```

应用程序是否配置为使用 Trust?

为了进行测试，您可以在 AboutPI-SDK/PISDKUtility Connections（AboutPI-SDK/PISDKUtility 连接）-> Options（选项）中更改 SDK 协议顺序并移除除“PI Trust”外的所有协议。如果您使用的是 PI SDK 1.4 及更高版本，还可以在客户端中使用 %PIHOME% 目录中的 piartool 将身份验证方法限制为仅“PI Trust”。

```
C:\Program Files\PIPC\ADM>piartool -node PIServer -trust -authdebug -block pibasess -verbose Debug Authentication: PIarray: [$Workf
ile: pitmplar.hxx $ $Revision: 60 $]:: _msize> 1 _mmaxsize> 8 _mincr> 8 _ChunkCount> 1 0 ProtocolType: Authprototype_LegacyTrust Pr
otocol Result Status: ResultFail PIstatus returned: [-10407] No Access - Secure Object ProtoCredentialresult: DOMAIN\PIGuy ProtoDet
ails: Error occurred in command line processing: [-10407] No Access - Secure Object
```

当 PI Server 处理 Trust 请求时，它会对客户端的 IP 地址执行反向名称查找来获得其主机名。反向名称查找的延迟将导致 Trust 查找出现延迟。

如果客户端使用 PI 或 AF SDK 构建，则 PI Server 还会将客户端的 Windows 用户转换为 Windows SID。SID 查找延迟将导致 Trust 查找出现延迟。可以在 PI Server 日志中找到成功和失败的 Trust 请求的查找用时。

```
D 21-Jun-12 11:51:16 pibasess (4080)
>> Trust request from: DOMAIN\PIGuy|PIClient|10.0.0.2|piartool.exe failed: [-10413] No trust relation for
this request (72375)
```

应用程序使用 NTLM 还是 Kerberos 进行身份验证?

使用 WIS 进行的身份验证可能由于各种原因而失败（可能与 Windows 或 PI 有关）。排除 WIS 故障的第一步最好是确定问题是否与 Kerberos 有关。到达 PI Server 的身份验证尝试将在 PI Server 日志中列出。从这些消息中，您可以确定客户端使用 Kerberos 还是 NTLM 进行身份验证。

```
D 15-Nov-13 11:00:48 pinetmgr (7082)
>> Successful login ID: 91.Address: 10.0.0.2. Name: piartool(12488):remote.Identity List: PIOperators |
PIWorld.Environment Username : DOMAIN\PIGuy.Method: Windows Login (SSPI,Kerberos)
```

要在域环境中排除故障，大部分情况下，您可以在连接时使用 PI Server 的 IP 来强制采用 NTLM。

```
C:\Program Files\PIPC\ADM>piartool -node 10.8.18.37 -windows -authdebug -block pibasess -verbose Debug Authentication: PIarray: [$W
orkfile: pitmplar.hxx $ $Revision: 60 $]:: _msize> 1 _mmaxsize> 8 _mincr> 8 _ChunkCount> 1 0 ProtocolType: Authprototype_msoft_sspi
Protocol Result Status: ResultSuccess PIstatus returned: [0] Success ProtoCredentialresult: DOMAIN\PIGuy ProtoDetails: PIOperators
| PIWorld Subsystem pibasess query returned: [0] Success - Time (Sec):0.141998
```

注意：默认情况下，AF 和 PI Server 均不通过 IP 注册 SPN。因此，通过 IP 地址进行的连接将使用 NTLM 协议进行身份验证。

映射是否使用 Windows 组?

用户的 Windows 组成员关系记录在 Windows 用户标记中，它会在处理映射时传递到 PI Server。要确认您的 Windows 用户属于某个组，可以使用命令 whoami：

```
C:\>whoami /groups /fo list GROUP INFORMATION ----------------- Group Name: Everyone Type: Well-known group SID: S-1-1-0 Attribute
s: Mandatory group, Enabled by default, Enabled group Group Name: BUILTIN\Administrators Type: Alias SID: S-1-5-32-544 Attributes:
Group used for deny only
```

映射是否使用 BUILTIN\Administrators 组?

根据环境的不同，用户帐户控制 (UAC) 可能会从 Windows 用户标记中清除 BUILTIN\Administrators 组。因此，当 PI Server 使用它来扫描映射表时，将不使用它来授予映射。通过选项“Run as administrator”（以管理员身份运行）来启动应用程序应当会有所帮助。

应用程序是否基于 PI API?

通过 PI API 获得 PI Trust 与 PI SDK 和 AF SDK 有较大不同。连接时，PI API 仅向 PI Server 发送其应用程序名称。PI Server 负责为 PI API Trust 获得剩余信息：IP 地址和主机名。在大多数网络环境中，PI Server 为 PI API Trust 使用的 IP 地址将与客户端的私有 IP 地址不匹配。IP 地址已知后，PI Server 将执行反向名称查找以获得客户端的主机名。

注意：由于 PI API 在请求 Trust 时不包含其私有 IP 地址，因此 PI Server 必须使用网络连接中的信息来确定合适的 IP。由于 NAT（网络地址转换），因此用于 PI Trust 查找的 IP 地址通常与客户端的私有 IP 不同。这也会影响按主机名使用 Trust 的 PI API 客户端。

应用程序基于 AF 还是 PI SDK?

在基于 PI 或 AF SDK 构建的客户端负责发送用于执行 Trust 查找的所有信息（IP 地址、主机名、应用程序名称、Windows 域、Windows 用户）。此信息可以在客户端预先确定，方法是使用 **pidiag**：

.code C:\Program Files\PIPC\ADM>pidiag -host Domain <DOMAIN> Machine <PIClient> User <PIGuy> IP Addr <10.0.0.2> FQDN <PIClient.domain.int> Primary Domain Controller: \\dc.domain.int IPV6 Compatible host address resolution: Reverse name lookup setting: 1 Name: PIClient.domain.int Address: fe80::5491:88f4:1666:77c2%16 Family: PF_INET6 Name: PIClient.domain.int Address: 10.0.0.2 Family: PF_INET Reverse name lookup setting: 0 Name: fe80::5491:88f4:1666:77c2%16 Address: fe80::5491:88f4:1666:77c2%16 Family: PF_INET6 Name: 10.0.0.2 Address: 10.0.0.2 Family: PF_INET .code

---

应用程序是否与多个 Trust 匹配?

如果找到多个匹配项，PI Server 仅会选择一个 Trust。如果应用程序在不同条件下与多个 Trust 相匹配，它可能被授予一个非用户预期的 Trust。有关 Trust 匹配如何工作的更多信息，请参阅 KB00964 - Trust Precedence（Trust 优先级）。

Article ID:	KB01154	**Created:**	2018-01-25
Article Type:	Troubleshooting	**Last Updated:**	2018-05-10