基于属性的访问控制

属性是配置中使用的自定义网络对象。您可以在ASA配置中定义和使用它们来过滤与VMware ESXi环境（由VMware vCenter管理）下的一个或多个虚拟机相关的流量。通过属性，您可以定义访问控制列表 (ACL) 为来自共享一个或多个属性的虚拟机组的流量分配策略。可向 ESXi 环境内的虚拟机分配属性并配置属性代理，属性代理使用 HTTPS 连接到 vCenter 或单个 ESXi 主机。然后，代理将请求和检索一个或多个绑定，从而将特定属性与虚拟机的主 IP 地址关联起来。

所有硬件平台以及 ESXi、KVM 或 HyperV 虚拟机监控程序上运行的所有 ASA Virtual 平台均支持基于属性的访问控制。只能从 ESXi 虚拟机监控程序上运行的虚拟机中检索属性。

- 基于属性的网络对象准则，第 1 页
- 配置基于属性的访问控制，第 2 页
- 监控基于属性的网络对象 ，第 9 页
- 基于属性的访问控制的历史记录，第 10 页

基于属性的网络对象准则

IPv6 准则

- vCenter 不支持使用 IPv6 地址作为主机凭证。
- IPv6 支持虚拟机绑定，其中虚拟机的主 IP 地址是 IPv6 地址。

其他准则和限制

- 不支持多情景模式。基于属性的网络对象仅支持单情景模式。
- 基于属性的网络对象仅支持绑定到虚拟机的主地址。不支持绑定到单一虚拟机的多个 vNIC。
- 只能为用于访问组的对象配置基于属性的网络对象。不支持为其他功能（NAT 等）配置网络对象。
- 虚拟机必须运行 VMware Tools，才能向 vCenter 报告主 IP 地址。不会通知 ASA 属性变更情况，除非 vCenter 知道虚拟机的 IP 地址。这是 vCenter 的限制。
- Amazon Web Services (AWS) 或 Microsoft Azure 公共云环境不支持基于属性的网络对象。

配置基于属性的访问控制

以下操作步骤介绍在 VMware ESXi 环境下，在托管的虚拟机上实施基于属性的访问控制的通用顺序。

过程

步骤 1 为托管的虚拟机分配自定义属性类型和值。请参阅配置 vCenter 虚拟机的属性，第 2 页。

步骤 2 配置一个属性代理，以连接到您的 vCenter 服务器或 ESXi 主机。请参阅配置虚拟机属性代理，第 4 页。

步骤 3 配置部署方案所需的基于属性的网络对象。请参阅配置基于属性的网络对象，第 5 页。

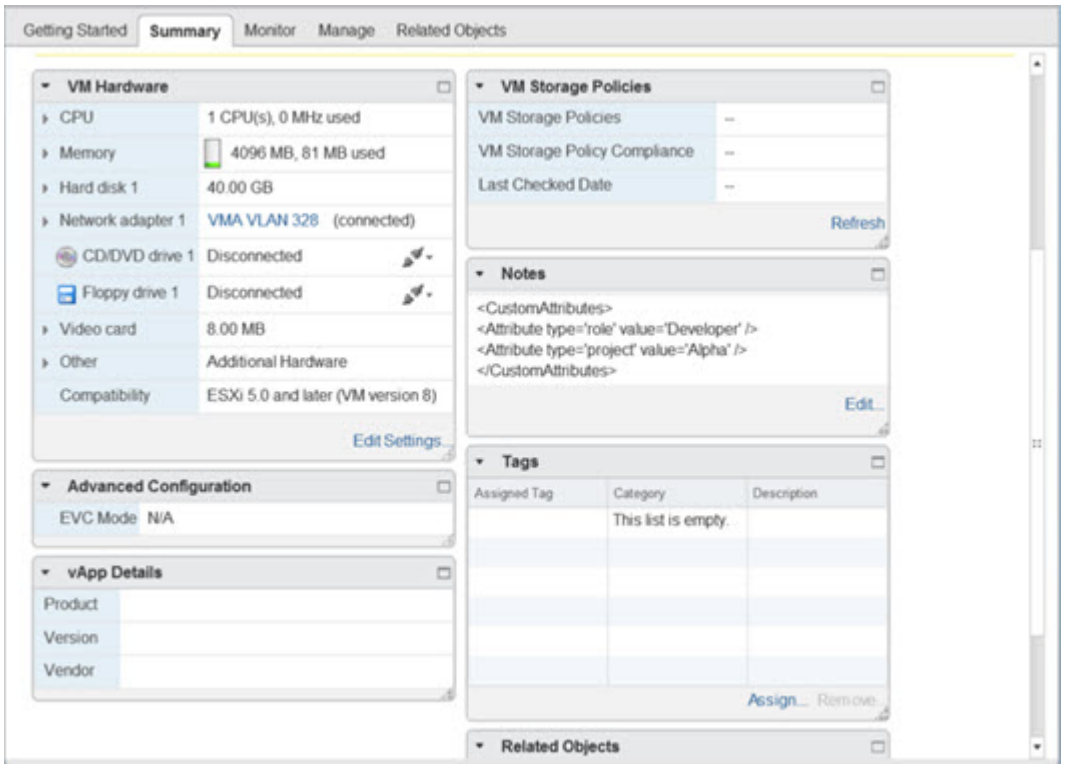
步骤 4 配置访问控制列表和规则。请参阅使用基于属性的网络对象配置访问控制，第 7 页。

配置 vCenter 虚拟机的属性

您可以为虚拟机分配自定义属性类型和值，并将这些属性与网络对象相关联。然后，即可使用这些基于属性的网络对象向一组具有通用用户定义特征的虚拟机应用 ACL。例如，可以隔离开发人员构建设备和测试设备，或按项目和/或位置组合虚拟机。要使 ASA 基于属性监控虚拟机，需要使 vCenter 可通过管理的虚拟机使用这些属性。为此，可向 vCenter 中虚拟机 Summary 页面的 Notes 字段插入一个格式化的文本文件。

在下图中可以看到 Notes 字段。

图 *1: vCenter* 中虚拟机的 ***Summary*** 选项卡



要指定自定义属性，需将格式正确的 XML 文件复制到虚拟机的 Notes 字段。该文件的格式为：

```
<CustomAttributes>
<Attribute type='attribute-type' value='attribute-value'/>
...
</CustomAttributes>
```

单个虚拟机可通过重复以上第二行定义多个属性。请注意，每行必须标识唯一的属性类型。如果为多个属性值定义的属性类型相同，则每次该属性类型的绑定更新都会覆盖上一次更新的值。

对于字符串属性值，与对象定义关联的值必须与虚拟机向 vCenter 报告的值完全匹配。例如，在虚拟机上，属性值 *Build Machine* 与注记值 *build machine* 不匹配。对于此属性，不会将绑定添加到主机映射。

在单个文件中可以定义多个唯一的属性类型。

过程

步骤 1 从 vCenter 清单中选择虚拟机。

步骤 2 点击虚拟机的 **Summary** 选项卡。

步骤 3 在 **Notes** 字段中，点击 **Edit** 链接。

步骤 4 将自定义属性文本文件粘贴到 **Edit Notes** 框中。该文本文件应遵循 XML 模板格式：

示例：

```
<CustomAttributes>
<Attribute type='attribute-type' value='attribute-value'/>
...
</CustomAttributes>
```

步骤 5 点击**确定 (OK)**。

示例

以下示例显示格式正确的 XML 文件文件，该文件用于为可应用于虚拟机的对象“role”和“project”定义自定义属性：

```
<CustomAttributes>
<Attribute type='role' value='Developer'/>
<Attribute type='project' value='Alpha'/>
</CustomAttributes>
```

配置虚拟机属性代理

配置虚拟机属性代理可实现与 vCenter 或单个 ESXi 主机的通信。当您向 VMware 环境下的虚拟机分配属性时，属性代理会向 vCenter 发送一条消息，指明已配置哪些属性；而 vCenter 的响应是绑定更新配置的属性类型匹配的每个虚拟机。

虚拟机属性代理和 vCenter 交换绑定更新的方式如下：

- 如果代理发出获取新属性类型的请求，vCenter 的响应是绑定更新已配置该属性类型的每个虚拟机。在此之后，vCenter 仅在添加或更改属性值时才会发出新绑定。
- 如果一个或多个虚拟机的受控属性发生变化，则会收到绑定更新消息。根据报告该属性值的虚拟机的 IP 地址标识每个绑定消息。
- 如果多个属性由单一代理监控，则单一绑定更新包含每个虚拟机所有受监控的属性的当前值。
- 如果虚拟机中并未配置代理监控的某个特定属性，则绑定中对于该虚拟机包含一个空属性值。
- 如果尚未对虚拟机配置任何受监控的属性，则 vCenter 不会发送绑定更新。

每个属性代理仅与一个 vCenter 或 ESXi 主机通信。可以为单个 ASA 定义多个属性代理，每个属性代理与不同的 vCenter 进行通信，或者一个或多个属性代理与同一个 vCenter 进行通信。

过程

步骤 1 创建与 vCenter 通信的虚拟机属性代理：**attribute source-group** *agent-name* **type** *agent-type*

示例：

```
hostname(config)# attribute source-group VMAgent type esxi
```

agent-name 参数指定虚拟机属性代理名称。*type* 参数是属性代理的类型。

注释
目前，ESXi 是唯一受支持的代理类型。

步骤 2 配置您的 vCenter 主机凭证：**host** *ip-address* **username** *ESXi-username* **password** *ESXi-password*

示例：

```
hostname(config-attr)# host 10.122.202.217 user admin password Cisco123
```

步骤 3 配置 vCenter 通信的保持连接设置：**keepalive retry-interval** *interval* **retry-count** *count*

示例：

```
hostname(config-attr)# keepalive retry-timer 10 retry-count 3
```

默认保持连接计时器值为间隔 30 秒进行 3 次重试。

步骤 4 检查虚拟机属性代理配置：**show attribute source-group** *agent-name*

示例：

```
hostname(config-attr)# sh attribute source-group VMAgent

   Attribute agent VMAgent
   Agent type: ESXi
   Agent state: Inactive
   Connection state: Connected
   Host Address: 10.122.202.217
   Retry interval: 30 seconds
   Retry count: 3
```

Agent State 将保持非活动状态，直到您配置网络对象并指定与该对象关联的属性。

步骤 5 退出属性配置模式： **exit**

示例：

```
hostname(config-attr)# exit
```

配置基于属性的网络对象

基于属性的网络对象可根据与 VMware ESXi 环境下一个或多个虚拟机相关的属性过滤流量。您可以通过定义访问控制列表 (ACL) 为来自共享一个或多个属性的虚拟机组的流量分配策略。

例如，可以配置允许具有 *engineering* 属性的计算机访问具有 *eng_lab* 属性的计算机的访问规则。网络管理员可添加或删除 engineering 计算机和 lab 服务器，安全管理员管理的安全策略会自动应用，而无需手动更新访问规则。

过程

步骤 1 启用对象组搜索： **object-group-search access-control**

示例：

```
hostname(config)# object-group-search access-control
```

必须启用对象组搜索，才能配置基于属性的网络对象。

步骤 2 使用对象名称创建或编辑基于属性的网络对象：**object network** *object-id*

示例：

```
hostname(config)# object network dev
```

步骤 3 指定与对象关联的代理、属性类型和属性值：**attribute** *agent-name attribute-type attribute-value*

示例：

```
hostname(config-network-object)# attribute VMAgent custom.role Developer
```

agent-name 指定 VM 属性代理，请参阅 <XREF>。如果将基于属性的网络对象配置为使用尚未配置的属性代理，则系统会自动创建一个无证书、无默认保持连接值的占位符代理。此代理将保持“无可用证书”状态，直到使用 **host** 子命令提供主机证书。

attribute-type 和 *attribute-value* 对一起可定义唯一的属性。*attribute-type* 是随机字符串，并且必须包含 **custom.** 前缀。如果使用多个属性值对同一属性类型定义了多次，则最后定义的值将覆盖以前的值。

示例

以下示例为开发组创建基于属性的网络对象 *dev*，其角色为“Developer”。VM 属性代理与 vCenter 通信并返回匹配属性 *custom.role* 的所有虚拟机绑定：

```
hostname(config)# object network dev
hostname(config-network-object)# attribute VMAgent custom.role Developer
```

以下示例为测试组创建基于属性的网络对象 *test*，其角色为“Automation”。VM 属性代理与 vCenter 通信并返回匹配属性 *custom.role* 的所有虚拟机绑定。请注意，本例使用的虚拟机列表与上一个示例相同：

```
hostname(config)# object network test
hostname(config-network-object)# attribute VMAgent custom.role Automation
```

以下示例为项目组创建基于属性的网络对象 *project*，其角色为“Alpha”。VM 属性代理与 vCenter 通信并返回匹配属性 *custom.project* 的所有虚拟机绑定。请注意，某些计算机的多个属性重叠：

```
hostname(config)# object network project
hostname(config-network-object)# attribute VMAgent custom.project Alpha
```

以下示例显示具有待定属性请求、处于活动状态的 VM 属性代理：

```
hostname(config-attr)# show attribute source-group VMAgent

   Attribute agent VMAgent
   Agent type: ESXi
   Agent state: Active
   Connection state: Connected
   Host Address: 10.122.202.217
   Retry interval: 30 seconds
   Retry count: 3
   Attribute requests pending:
       'custom.project'
       'custom.role'
```

使用基于属性的网络对象配置访问控制

在对来自共享一个或多个属性的虚拟机组的流量定义访问控制列表 (ACL) 时，可以使用基于属性的网络对象。访问列表包含一个或多个访问控制条目 (ACE)。一个 ACE 指访问列表中的一个条目，指定允许或拒绝规则（转发或丢弃数据包）。通常，允许或拒绝规则应用于协议、源与目标 IP 地址或网络，也可以选择性地应用于源与目标端口。

使用基于属性的网络对象时，可以使用这些对象替换源和/或目标 IP 地址。在部署、移动或废弃虚拟机时，可更新虚拟机上的属性，同时分配的访问控制策略可保持有效，不进行配置更改。

有关 ACL 所有可用选项的完整信息，请参阅配置 ACL。

过程

步骤 1 使用基于属性的网络对象创建和配置扩展 ACL 条目 (ACE)：**access-list** *access_list_name* **extended** {**deny** | **permit**} *protocol_argument* **object** *source_object_name* **object** *dest_object_name*

示例：

```
hostname(config)# access-list lab-access extended permit ip object dev object test
```

注释
根据需要，为您的策略重复上述操作。

选项有：

- *access_list_name* - 新的或现有 ACL 的名称。
- Permit 或 Deny - 如果条件匹配，**deny** 关键字可拒绝或排除数据包。如果条件匹配，**permit** 关键字可允许或添加数据包。
- Protocol - *protocol_argument* 指定 IP 协议：
  - *name* 或 *number* - 指定协议名称或编号。指定 **ip** 以应用于所有协议。
  - **object-group** *protocol_grp_id* - 指定使用 **object-group protocol** 命令创建的协议对象组。
- Source Object - **object** 指定使用 **object network** 命令创建的基于属性的网络对象。*source_object_name* 指定发送数据包的对象。
- Destination Object - **object** 指定使用 **object network** 命令创建的基于属性的网络对象。*dest_object_name* 指定将数据包发送到的对象。

步骤 2 将 ACL 绑定到接口或全局应用该 ACL：**access-group** *access_list_name* {**in interface** *interface_name* | **global**}

示例：

```
hostname(config)# access-group lab-access in interface inside
```

对接口特定的访问组：

- 指定扩展 ACL 名称。可以为每个接口上的每种 ACL 类型配置一个 **access-group** 命令。
- **in** 关键字会将 ACL 应用于入站流量。
- 指定 **interface** 名称。

对于全局访问组，指定 **global** 关键字，以将扩展 ACL 应用于所有接口的入站方向流量。

示例

以下示例显示如何全局应用基于属性的扩展 ACL：

```
hostname(config)# access-list lab-access extended permit ip object dev object test
hostname(config)# access-group lab-access global
hostname(config)# show access-list
access-list cached ACL log flows: total 0, denied 0 (deny-flow-max 4096)
            alert-interval 300
access-list lab-access; 1 elements; name hash: 0x62b4790b
access-list lab-access line 1 extended permit ip object dev object test (hitcnt=0) 0x64a1be76

  access-list lab-access line 1 extended permit ip object dev(2) object test(3) (hitcnt=0)
 0x64a1be76
```

监控基于属性的网络对象

要监控基于属性的网络对象，请输入以下命令：

- **show attribute host-map**

  显示特定属性的代理、类型和值的属性绑定。

- **show attribute object-map**

  显示对象到属性的绑定。

- **show attribute source-group**

  显示配置的 VM 属性代理。

示例

以下示例显示主机到属性绑定的映射：

```
hostname# show attribute host-map /all
IP Address-Attribute Bindings Information

          Source/Attribute                          Value
=========================================================================
VMAgent.custom.project                          'Alpha'
     10.15.28.34
     10.15.28.32
     10.15.28.31
     10.15.28.33
VMAgent.custom.role                             'Automation'
     10.15.27.133
     10.15.27.135
     10.15.27.134
VMAgent.custom.role                             'Developer'
     10.15.28.34
     10.15.28.12
     10.15.28.31
     10.15.28.13
```

以下示例显示对象到属性的绑定：

```
hostname# show attribute object-map /all
Network Object-Attribute Bindings Information

Object
          Source/Attribute                          Value
=========================================================================
dev
     VMAgent.custom.role                             'Developer'
test
     VMAgent.custom.role                             'Automation'
project
     VMAgent.custom.project                          'Alpha'
```

以下示例显示属性代理配置：

```
hostname# show attribute source-group
   Attribute agent VMAgent
   Agent type: ESXi
   Agent state: Active
   Connection state: Connected
   Host Address: 10.122.202.217
   Retry interval: 30 seconds
   Retry count: 3
   Attributes being monitored:
      'custom.role' (2)
```

基于属性的访问控制的历史记录

功能名称	平台版本	说明
基于属性的网络对象的支持	9.7.(1)	现在，除 IP 地址、协议和端口等传统网络特征之外，您还可以使用虚拟机属性控制网络访问。虚拟机必须位于 VMware ESXi 环境之下。 引入了以下命令： **object network** *attribute* **attribute** *agent-name attribute-type attribute-value* **attribute source-group** *agent-name* **type** *agent-type* **host** *ip-address* **username** *ESXi-username* **password** *ESXi-password* **keepalive retry-interval** *interval* **retry-count** *count*
从 ASA 5506-X（所有型号）、5508-X、5512-X 和 5516-X 中删除了对基于虚拟机属性的网络对象的支持。	9.10(1)	您无法再在以下平台上使用基于虚拟机属性的网络对象：ASA 5506-X（所有型号）、5508-X、5512-X、5516-X。

当地语言翻译版本说明

思科可能会在某些地方提供本内容的当地语言翻译版本。请注意，翻译版本仅供参考，如有任何不一致之处，以本内容的英文版本为准。