Windows提供了一系列基于虚拟化安全性(VBS)功能，例如Device Guard和Credential Guard。这些功能可以借助虚拟机监控程序提高操作系统和用户数据的安全性。

尽管Trustlet也是常规的Windows可移植可执行(PE)文件，但它们包含了某些与IUM有关的属性：

1. 由于Trustlet可用的系统调用数量有限，因此只能从有限的Windows系统DLL(C/C++运行时、KernelBase、Advapi，RPC运行时、CNG Base Crypto以及NTDLL)中导入。注意，只作用于数据结构(例如NTML、ASN.1等)的数学性DLL也是可用的，因为它们不执行任何系统调用。
2. 可以从自己可用的、与IUM有关的系统DLL(也叫Iumbase)中调用，这些DLL提供了Base IUM系统API，并对邮槽(mailslot)、存储框(storage box)、加密(cryptography)等提供了支持。最终这个库会调用至Iumdll.dll(VTL 1版本的Ntdll.dll)

IUM是"Integrity Update"的缩写，是指Windows操作系统中的一项安全特性。它允许Windows对系统的核心组件进行更新并保持完整性，从而提高系统的安全性和稳定性。IUM的工作原理是在启动时通过检查数字签名来验证文件，以确保文件没有被篡改。如果文件的数字签名验证失败，则系统会拒绝启动该文件。IUM还允许Windows在系统运行时对核心组件进行更新，而不会破坏文件的完整性。这对于确保系统的安全性和稳定性至关重要。

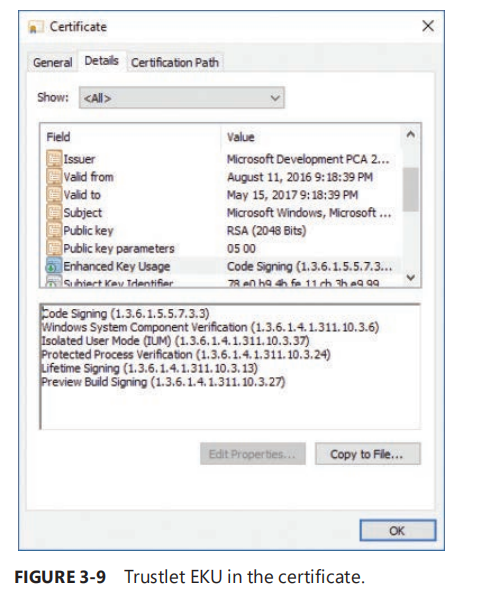
VTL 0是指Virtual Trust Level 0，是Windows 10中引入的一项新安全功能。它是一种基于硬件的虚拟化技术，可以提供比传统的用户模式应用程序更高的安全性。VTL 0运行在硬件层面，不受任何操作系统或软件的干扰，因此可以提供更加可靠的安全保障。

在VTL 0中，操作系统和应用程序运行在一个虚拟化容器中，该容器与物理硬件之间的通信受到严格控制。这使得恶意软件无法访问系统资源和数据，并防止对系统进行未经授权的修改。

VTL 0主要用于保护关键系统和数据，如金融、医疗和军事领域。它需要支持硬件的CPU，例如Intel的第7代Core处理器（Kaby Lake）及其后续版本。

VTL 1代表第二级虚拟化技术，也称为嵌套虚拟化。这是一种将虚拟化功能在虚拟机层面上进行叠加的技术，即在一个已经被虚拟化的虚拟机上再运行另一个虚拟机。在 VTL 1 中，每个虚拟机都有自己的虚拟机监视器（VMM），它们通过硬件的虚拟化功能来运行。在嵌套虚拟化中，第一级 VMM 控制着整个系统，并为第二级 VMM 提供虚拟化的环境。这样可以实现更灵活的虚拟化方案，如在云计算环境中更高效地使用资源。

1. 可以包含一个名为.tPolicy的PE节，以及一个名为s\_IumPolicyMetadata的导出的全局变量。借此为安全内核围绕对Truselet允许的VTL 0访问（例如允许调试、崩溃转储支持等）实现策略设置提供所需的元数据。
2. 对使用包含隔离用户模式EKU的证书进行签名。

证书中的Trustlet EKU

EKU 是 Extended Key Usage 的缩写，即扩展密钥用途，它是一个 X.509 数字证书中的扩展字段。该字段用于描述证书的使用情况，即该证书适用于哪些目的。在 SSL/TLS 中，服务器证书通常用 EKU 字段来指示服务器证书的用途，如证书是否可用于服务器身份验证、客户端身份验证等。 EKU 的值是一个 OID（Object Identifier，对象标识符），用于表示证书的使用目的。

Trustlet 策略元数据

策略元数据包含了多种选项，可用于配置Trustlet在VTL 0模式下的“可访问性”。这是通过s\_IumPolicyMetadata导出结构描述的，其中可包含一个版本号以及Trustlet ID。后者具备唯一性，可用于在各种已知Trustlet中区分指定的那一个（例如BioIso.exe(一款生物识别软件)的Trustlet ID为4）最后，元数据也提供了一组策略选项，目前可支持的选项见下表：

|  |  |  |
| --- | --- | --- |
| 策略 | 含义 | 详情 |
| ETW | 启用或禁用ETW | ETW（Event Tracing for Windows）是Windows操作系统提供的一种高效的事件跟踪机制，用于在运行时捕获和记录应用程序和系统事件。ETW是一个低开销的，高性能的事件基础结构，可以在Windows操作系统中跟踪各种事件，包括进程、线程、驱动程序、系统事件等等。通过ETW，开发人员可以对应用程序和系统进行高级的性能分析和故障排除。 |
| Debug | 配置调试 | 调试可随时调用，但前提是SecureBoot已禁用，或使用按需质询/回应机制 |
| Crash Dump | 启用或禁用崩溃转储 |  |
| Crash Dump Key | 指定崩溃转储加密用的公钥 | 转储文件可提交给微软产品团队，他们有解密的私钥 |
| Crash Dump GUID | 指定崩溃转储密钥的标识符 | 借此平产团队即可使用/识别多个密钥 |
| Parent Security Descriptor | SDDL格式 | 用于验证所有者/父进程是否符合预期 |
| Parent Security Descriptor Revision | SDDL格式修订ID | 用于验证所有者/父进程是否符合预期 |
| SVN | 安全版本 | 具备唯一性的数字，可被Trustlet用于(以自己的身份)加密AES256/GCM消息 |
| Device ID | 安全设备PCI标识符 | Trustlet只能与PCI ID匹配的安全设备通信 |
| Capability | 启动强大的VTL 1能力 | 可允许访问Create Secure Section API、DMA以及对安全设备和安全存储API进行用户模式MMIO访问 |
| Scenario ID | 为此二进制文件指定场景ID | 编码为GUID，为确保可用于已知场景，必须在创建安全映像节时由Trustlet指定 |

Trustlet的属性

启动Trustlet需要正确使用PS\_CP\_SECURE\_PROCESS属性，该属性用于验证调用方是否真正希望创建一个Trustlet,并可验证所执行的Trustlet是否确实是调用方的希望执行的。这是通过在属性中嵌入的Trustlet标识符实现的，该标识符必须与策略元数据所包含的Trustlet ID一致。随后可以指定一个或多个属性，如下表：

|  |  |  |
| --- | --- | --- |
| 属性 | 含义 | 详情 |
| Mailbox Key | 用于检索邮箱(mailbox)数据 | 邮箱可供Trustlet与VTL 0环境共享数据，前提是Trustlet密钥必须是已知的 |
| Collaboration ID | 设置在使用安全存储IUM API时使用的合作ID | 安全存储可供Trustlet在相互之间共享数据，前提是它们具备相同的合作ID。如果合作ID不存在，可用Trustlet实例ID代替 |
| TK Session ID | 标识会在Crypto中使用的会话ID |  |

PS\_CP\_SECURE\_PROCESS是Windows操作系统中的一个安全机制，用于确保系统上运行的进程不会被恶意软件篡改。它通过在进程创建期间向NT创建进程参数注入一个安全令牌来实现此目的。该安全令牌包含有关当前进程的信息，如其安全描述符和其代码签名等。

当使用PS\_CP\_SECURE\_PROCESS时，进程创建参数将传递到NT内核。NT内核将使用进程创建参数来生成一个新的安全令牌，该安全令牌将与要创建的进程关联。安全令牌包含有关当前进程的安全信息，以及其他用于确保其代码完整性的信息。然后，NT内核将使用这个新的安全令牌来创建进程。由于进程创建参数是由NT内核处理的，因此即使恶意软件尝试篡改这些参数，也无法绕过PS\_CP\_SECURE\_PROCESS机制。

系统内置的Trustlet

Win10内置

|  |  |  |
| --- | --- | --- |
| 二进制名称(Trustlet ID) | 描述 | 策略选项 |
| LasIso.exe(1) | Credential Guard和Key Guard Trustlet | 允许ETW、禁用调试、允许崩溃转储加密 |
| Vmsp.exe(2) | 安全虚拟机工作进程  (vTPM Trustlet) | 允许ETW、禁用调试、禁用崩溃转储、启用安全存储能力、验证父安全描述符是否为S-1-5-83-0(NT VIRTUAL MACHINE/Virtual Machines) |
| Unknown(3) | vTPM密钥登记Trustlet | 未知 |
| BioIso.exe(4) | 安全生物特征Trustlet | 允许ETW、禁用调试、允许崩溃转储加密 |
| FsIso.exe(5) | 安全框架服务器Trustlet | 禁用ETW、允许调试、启用创建安全区域能力，使用场景ID |

Trustlet的标识

1. **Trustlet标识符或Trustlet ID**
2. **Trustlet实例**
3. **合作ID**
4. **安全版本号**，可供Trustlet通过强密码证明的方式了解已签名或已加密数据的来源
5. **场景ID**，可供Trustlet创建命名安全内核对象

隔离用户模式服务

以Trustlet的方式运行，好处不仅在于可以防范来自常规(VTL 0)环境的攻击，而且可以访问安全内核专门为Trustlet提供的特权和受保护安全系统调用。此类服务包括(下面括号里的都是IUM驱动的API)

1. 安全设备(IumCreateSecureDevice, IumDmaMapMemory, IumGetDmaEnabler, IumMapSecureIo, IumProtectSecureIo, IumQuerySecureDeviceInformation,IopUnmapSecureIo, IumUpdateSecureDeviceState)

IumCreateSecureDevice：创建安全设备，用于创建安全 I/O 设备对象，并创建一个客户端上下文。

IumDmaMapMemory：在当前进程的上下文中将指定的虚拟内存页映射到 DMA 运行时的虚拟地址空间。

IumGetDmaEnabler：获取 DMA 使能程序，用于请求 DMA 硬件资源，返回 DMA 使能器对象。

IumMapSecureIo：映射安全 I/O 设备的 I/O 端口、内存和中断资源到调用者的地址空间。

IumProtectSecureIo：保护安全 I/O 设备，用于授权 DMA 访问并保护映射到调用者地址空间的 I/O 端口、内存和中断资源。

IumQuerySecureDeviceInformation：查询安全设备信息，返回安全 I/O 设备的安全性属性，例如硬件标识符和加密密钥。

IopUnmapSecureIo：取消映射安全 I/O 设备，释放调用者地址空间中映射的 I/O 端口、内存和中断资源。

IumUpdateSecureDeviceState：更新安全设备状态，用于更新安全 I/O 设备的状态信息，例如 DMA 通道状态和 DMA 完成状态。

1. 安全区域(IumCreateSecureSection, IumFlushSecureSectionBuffers, IumGetExposedSecureSection, IumOpenSecureSection)

**IumCreateSecureSection**：创建一个受保护的内存区域，可以被用来在用户模式和内核模式之间共享数据；

**IumFlushSecureSectionBuffers**：刷新指定的受保护内存区域的缓冲区；

**IumGetExposedSecureSection**：返回一个指定名称的受保护内存区域的句柄；

**IumOpenSecureSection**：打开一个已存在的受保护内存区域，返回一个句柄。

1. 邮箱(IumPostMailbox) IumPostMailbox：用于在安全环境和系统环境之间发送消息。
2. 标识密钥(IumGetIdk) IumGetIdk：获取系统标识符 (IDK)。
3. 加密服务(IumCrypto) IumCrypto：提供加密和解密的功能。
4. 安全存储(IumSecureStorageGet, IumSecureStoragePut)

IumSecureStorageGet：用于获取存储在安全存储区域中的数据。

IumSecureStoragePut：用于将数据存储在安全存储区域中。

Trustlet可访问的系统调用

为了通过安全内核将攻击面和暴露范围降至最小，常规应用程序只能使用一小部分系统调用，这些系统调用与Trustlet可以使用的系统DLL相兼容，并提供RPC运行时和ETW追踪所需服务的最低要求。

1. **Worker Factory和线程API**。用于支持RPC所用的线程池API和加载程序所用的TLS槽
2. **进程信息API**。用于支持TLS槽和线程栈分配
3. **事件、信号量**。用于支持线程池和同步
4. **高级本地过程调用（ALPC）API**。用于支持通过ncalrpc传输的本地RPC
5. **系统信息API**。用于支持读取安全引导信息、用于Kernel32.dll和线程池缩放的基本和NUMA系统信息、性能信息，以及时间信息的子集。
6. **令牌API**。为RPC模拟提供了最小支持
7. **虚拟内存分配API**。用于支持由用户模式堆管理器进行的分配。
8. **节API**。用于支持（DLL映像的）加载程序以及安全区域功能
9. **追踪控制API**。用于支持ETW
10. **异常和继续API**。用于支持结构化异常处理（SHE，Structured Exception Handling）