使用 Precryption™ 消除最大的盲点

Gigamon Precryption 技术可为整个安全堆栈提供横向流量的明文可见性，包括虚拟环境、云环境和容器在内。无需任何解密操作。

#1

IT 安全主管最为担忧的问题：
存在暗藏的盲点被悄然利用[2]

91%

的威胁**使用加密通道**[3]

81%

的组织**在过去一年里遭遇过云安全事件**[1]

5 倍

面对未加密的流量，安全工具的**有效性可以提高五到七倍**[4]

31%

发生在去年的数据泄露事件未被安全工具和可观测性工具发现[2]

Gigamon Precryption™ 技术重新定义了虚拟环境、云环境和容器化应用程序的安全防护理念，为整个安全堆栈提供加密通信的明文可见性，避免了过往解密操作所带来的高昂成本和复杂性。

信息安全挑战

1. 云技术的采用率不断上升
2. 开发团队忙于赶进度
3. 隐蔽的威胁活动

在当今的混合云基础设施中，加密通信的使用几乎无处不在，目的是保护敏感数据免受传统拦截技术的监听。对此，威胁行为者针对性地采用了新的、更复杂的渗透方法来攻击关键系统，窃取敏感数据。现在，他们又利用相同的加密信道来掩盖自己的活动，尤其是横向移动、敏感数据访问和数据渗漏操作。而使用目前市面上的现有解决方案，几乎不可能获得对虚拟工作负载之间横向移动的加密流量的明文可见性，导致检测隐蔽的威胁活动异常艰难。因此，加密的横向通信仍然是安全防护中最大的盲点。

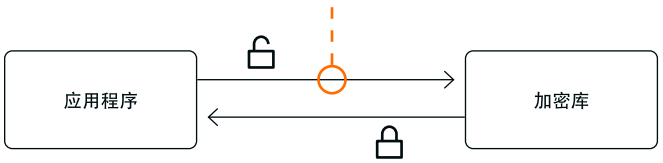
Precryption 技术可发现隐蔽的威胁活动

Precryption 技术是一种创新的解决方案，直击当今混合云基础设施中最大的盲点，即威胁行为者企图利用 TLS 1.3 等最新加密形式来掩蔽横向移动。Precryption 以高效、流畅的方式提供对加密虚拟通信的明文可见性，免去了代价高昂的解密操作与繁琐的密钥收集和管理。

Precryption 技术的运作原理

Precryption 技术利用 Linux 原生功能来捕获或复制应用程序与加密库（如 OpenSSL）之间的通信。

Precryption 捕获数据的位置



通过这种方式，无论在通信加密之前还是解密之后，Precryption 都能以明文形式捕获网络流量。Precryption 的运作不会干扰消息的实际加密，也不会干扰消息在网络上的传输。过程中无需代理、重传，也无需中断和检查。取而代之的是，明文副本被转发到 Gigamon Deep Observability Pipeline，进行进一步的优化、转换、复制，并传递至相应工具。

Precryption 技术建立在 GigaVUE® Universal Cloud Tap (UCT) 之上，适用于混合环境和多云环境，包括本地和虚拟平台。

另一个优势在于，结合 Precryption 技术的 UCT 独立于应用程序运行，不必纳入应用程序开发生命周期。

主要用例

阻止网络攻击：云环境中的横向移动是一个盲点，在网络攻击中尤为明显。加密数据包在通过边界安全防线后便不受监控，导致威胁行为者有机会使用各种技巧和手段来逃避检测。

TLS 1.3 合规性：如今，一些组织推迟了对所要求的 TLS 1.3 的采用，主要是因为缺乏对加密流量的可见性。另一些组织则不得不采用多个独立的解密解决方案。

零信任：对于任何有效的零信任架构，关键基础在于能够查看数据包、检查网络上资源之间的每一次交互并应用策略。

基于网络的情报分析：诸如 SIEM 这样的安全工具通常依赖 Metadata 元数据转换和增强来更好地检测威胁。

选择 Gigamon Precryption 的理由

结合 Precryption 技术的 GigaVUE Universal Cloud Tap 是一款运行流畅的轻量级解决方案，可消除当今混合云基础设施中存在的盲点，为虚拟环境、云环境和容器平台提供东西向可见性。它可提供对包括 TLS 1.3 在内的所有加密类型的清晰可见性，无需管理和维护解密密钥。如此一来，IT 组织不仅能够有效管理合规性，保持私有通信的私密性，为零信任奠定必要的基础，还能将安全工具的有效性提高 5 倍甚至更多。

主要特点

- **对采用现代加密协议**（TLS 1.3、mTLS 和具有完美前向保密的 TLS 1.2）**的通信，提供明文可见性**
- **对采用传统加密协议**（TLS 1.2 及更早版本）**的通信，提供明文可见性**
- **非侵入式流量访问**，无需在容器工作负载内运行代理
- **免去了**与传统流量解密相关的**昂贵资源消耗**
- **免去了**传统流量解密所需的**密钥管理**
- **对性能的负面影响为零**，不受密码类型、强度或版本的影响
- **支持混合环境和多云环境**，包括本地环境、虚拟环境和容器平台
- **保持网络中私有通信的私密性**，并将明文威胁活动信息传递至安全工具
- **与 Gigamon Deep Observability Pipeline 相集成**，提供全套优化、转换和代理功能

主要优势

- **消除东西向（横向）和南北向加密通信中存在的盲点**，包括可能不穿过防火墙的流量
- **使用独立的方法监控应用程序通信**，以提升开发团队的速度
- **将安全工具的可见性扩展**到所有通信，无论采用何种加密类型
- 在虚拟环境中**实现流量捕获效率最大化**
- 通过使用未加密的数据，将安全工具的**性能提升 5 到 7 倍**
- **支持**基于深度可观测性的**零信任架构**
- 在解密流量管理过程中**保持私密性**和合规性

挑战：深入剖析

IT 组织在保护受托系统和数据的安全时，面临三大挑战：虚拟技术和云技术的采用率不断上升，开发团队忙于赶进度，以及隐蔽的威胁活动。

1. 虚拟技术和云技术的采用率不断上升

81% 的组织在过去一年里遭遇过云安全事件[1]

虚拟化系统（无论是本地、私有云、公有云、虚拟机还是容器）的应用趋势继续增长，几乎没有放缓的迹象。这些最新架构旨在实现高效运营，并且发展速度远远超过了基于边界的安全架构。横向移动极难被检测。一些组织经过权衡后，选择承担一定的风险，允许加密通信在其混合云基础设施中流动；另一些组织则试图通过增设防火墙来加强虚拟架构，但牺牲了安全防护的效率。而且，由于大多数企业都拥有多个虚拟平台，挑战和风险会成倍增加。

2. 开发团队追求速度

83% 的组织在 **IT** 团队和安全团队之间实行了集体责任制[2]

软件开发团队主要致力于开发应用程序，为收入增长做出贡献或为组织节省时间和金钱。由于时常面临紧迫的截止期限，开发运维团队只能将重点放在核心职能上。他们或许对安全问题有所关注，但通常不是入侵防御方面的专家，对于可能引入的漏洞也知之甚少。此外，他们可能不愿在软件和系统中部署安全代理，因为代理会阻碍测试并增加软件开发生命周期的工作量和时间。

对此，安全团队的处理方式不尽相同。有些安全团队采用严格的合规性措施，强制要求在所有代码中部署代理；有些安全团队则将安全人员安插到开发团队；另一些安全团队则别无选择，只能默许开发人员在没有严格安全监督的情况下赶进度。但绝大多数的安全团队都会对开发团队实施一定程度的安全责任制。

3. 隐蔽的威胁活动

91% 的威胁使用加密通道[3]

加密通信对于预防某些威胁非常有效，但同时也为其他一些威胁提供了可乘之机。威胁行为者在获得系统的访问权限后，通常会立即删除、禁用和/或修改日志。紧接着，他们通过加密通信完成一系列操作，包括向命令与控制服务器发出指令，提升权限，进行横向移动，秘密复制数据，并最终完成数据的窃取。

安全工具在面对加密流量时，有效性可能会降低 **5** 到 **7** 倍[4]

常见的加密方法分为两类：

- 现代加密，使用完美前向保密 (PFS) 来防止对截获的通信进行任何中断检查解密，因为任何截获的加密密钥都是临时的，对于带外解密毫无用处。现代加密包括 TLS 1.3、mTLS 以及一些启用了 PFS 的 TLS 1.2 部署。据 Gigamon 估计，当今约 30-40% 的网络流量使用现代加密，而且这一比例将继续增长。

- 传统加密，不使用 PFS，可被截获的密钥解密。传统加密包括部分 TLS 1.2 部署及较早版本的 TLS 和 SSL（安全套接层）。

市面上有一些安全工具可以对使用加密通信的网络进行监控。对于传统加密，这些安全工具通常会尝试自行解密网络流量。但这是一种计算成本高昂的做法，会对性能产生严重影响，需要更多的“机箱”来满足处理需求。此外，底层密钥库必须持续更新，密钥的管理也非常耗时且复杂。但即便如此，这种做法仍然只能应对传统加密，而对现代加密无能为力。

对于现代加密，安全工具必须采取不同的做法，因为现代加密的通信无法“在中间过程”解密。因此，所采用的做法是将数据包标头、数据包大小、数据包频率和其他签名输入机器学习算法，以此来评估任何特定通信的风险。尽管这种做法总比没有好，但实际效果并不理想，迫使一些组织要么只监控传统加密，要么依赖于边界安全防护，要么禁止对应用程序进行现代加密，但这些都无法构成理想的安全态势。

近期，一项对 1,000 多位 IT 主管和安全主管的调查显示，31% 的数据泄露事件未被安全工具和可观测性工具发现。

亟需更好的解决方案。

Precryption 解决方案：深入剖析

GigaVUE Universal Cloud Tap (UCT) 现在结合了 Precryption 技术，消除了加密虚拟通信和容器通信中的盲点，使 IT 主管和安全主管能够重新掌握控制权。

GigaVUE UCT 是一款现代虚拟 TAP 工具，利用原生 Linux eBPF 技术，旨在为虚拟环境中的镜像通信提供最高效的方法。UCT 会将捕获的未加密数据高效地传递至 Gigamon Deep Observability Pipeline，进行进一步的优化、转换、过滤和代理，最终将正确的数据传递至正确的物理工具或虚拟工具。

Gigamon Precryption 技术建立在 GigaVUE UCT 之上，可与 Linux 和 OpenSSL 等加密库无缝集成，能够在网络通信加密前或解密后（适用于某些应用程序），捕获虚拟通信和容器通信。

- ✓ 网络通信在整个网络中未经任何改动、保持原样并维持加密状态。
- ✓ 无需成本高昂的解密计算。因此，Precryption 技术适用于现代加密和传统加密，不受密码类型、强度或版本的影响。
- ✓ 不会暴露应用程序密钥，没有管理应用程序密钥的麻烦，也不需要强行添加虚拟路由。
- ✓ Precryption 技术独立于受监控的应用程序运行，因而不会对应用程序的资源和生命周期管理有任何影响，也不会在应用程序内部造成故障。

Gigamon Precryption 技术的工作原理：单节点（图 1）

1. 当任何应用程序需要对消息进行加密时，会使用加密库（例如 OpenSSL）来执行实际加密。

2. 启用 Precryption 技术的 GigaVUE Universal Cloud Tap (UCT) 会在该消息加密前在网络上获得该消息的副本。

3. 加密后的消息会发送到接收端应用程序，保持原有加密状态不变。无需代理，无需重新加密，也无需重传。

4. GigaVUE UCT 根据需要创建数据包标头，封装在隧道中，然后转发到深度可观测性管道中的 GigaVUE V 系列。Gigamon 进一步优化、转换数据，并将数据传递至相应的工具，无需进一步解密。

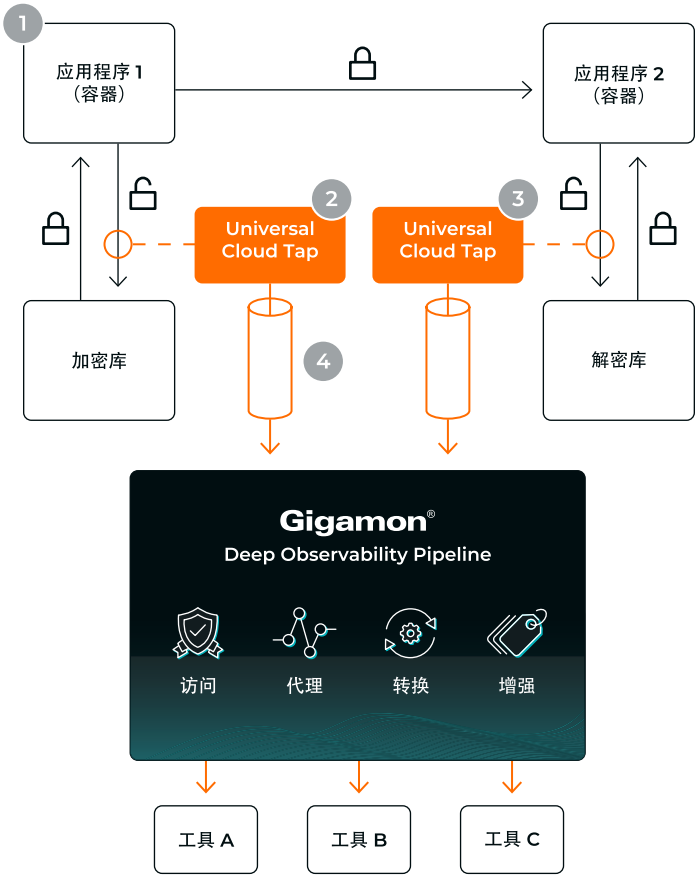


图 2

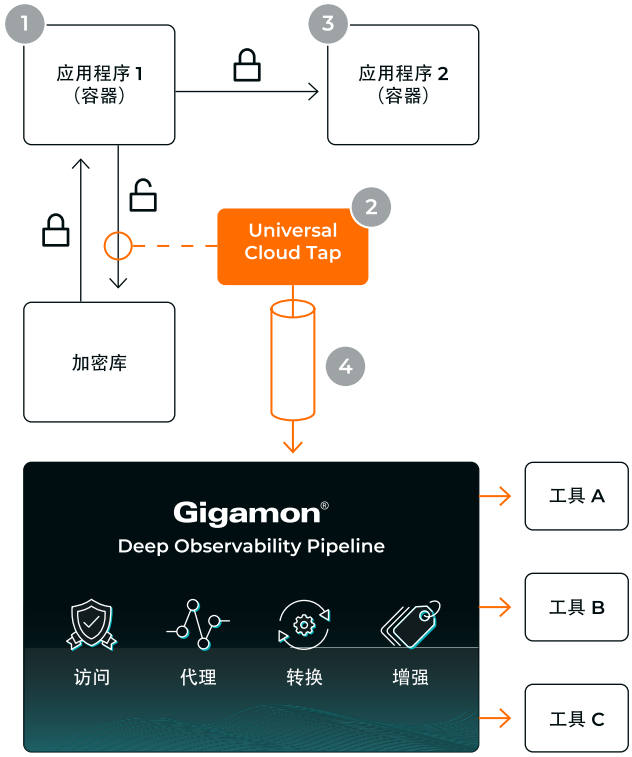


图 1

Gigamon Precryption 技术的工作原理：多节点（图 2）

1. 当任何应用程序需要对消息进行加密时，会使用加密库（例如 OpenSSL）来执行实际加密。

2. 启用 Precryption 的 GigaVUE Universal Cloud Tap (UCT) 会在该消息在网络上被加密前，获得该消息的副本。

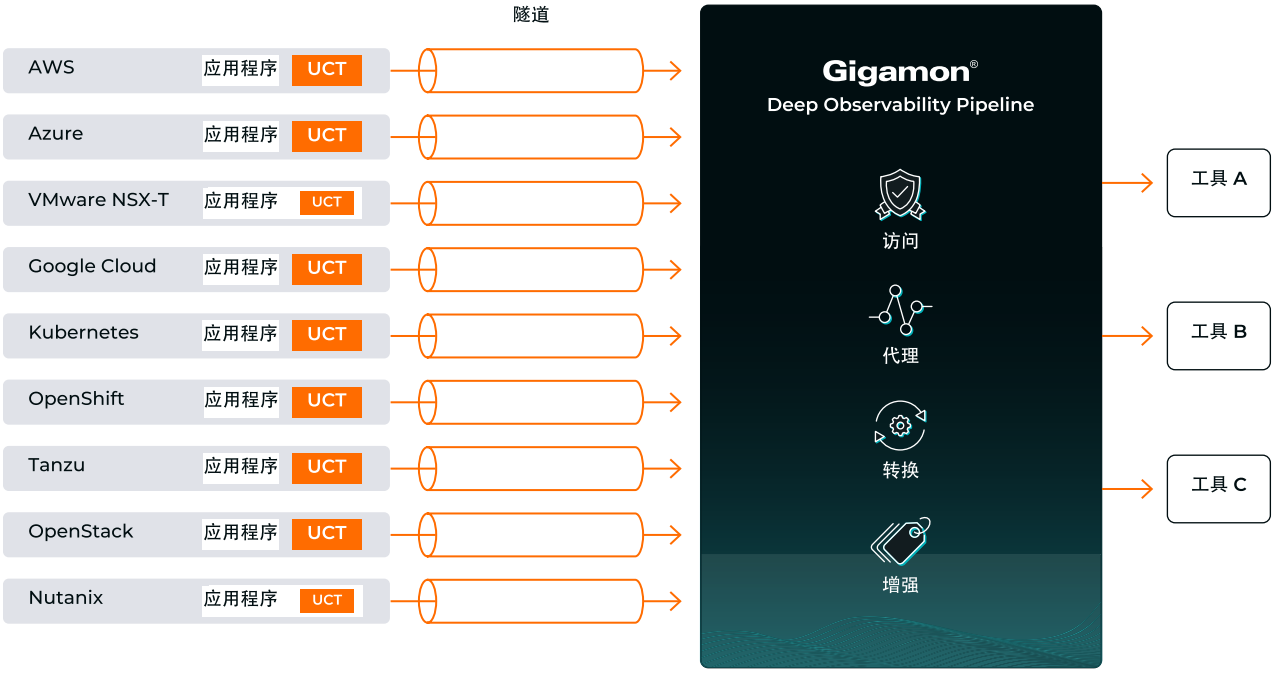
3. 或者，启用了 Precryption 技术的 GigaVUE UCT 也可以在解密后从服务器端捕获该消息的副本。

4. GigaVUE UCT 根据需要创建数据包标头，封装在隧道中，然后转发到深度可观测性管道中的 V 系列，进一步增强、转换数据并将数据传递至相应的工具，无需进一步解密。

用于多云和大规模环境

结合 Precryption 技术的 GigaVUE UCT 适用于多个虚拟平台和云平台，包括 VMware、AWS、Microsoft Azure、OpenStack、Google Cloud、Nutanix 等，可将数据输入到一个公共的数据管道中，并通过一站式全局管理界面进行管理。

- ✓ 支持在 Kubernetes 中自动部署，便于扩展
- ✓ 所有云环境共用一个许可证池；实例数量不受限制



GigaVUE UCT 独立于应用程序运行

“代理”一词的含义可能因上下文环境而异。可参考下表来了解 UCT 相较于典型代理所具备的优势。

典型代理	GigaVUE UCT
X 在应用程序空间/容器集内运行	✓ 位于独立容器集内的独立用户空间
X 影响应用程序资源的使用	✓ 独立的节点资源
X 需要协调版本升级	✓ 独立升级
X 需要与应用程序一起测试	✓ 独立的生命周期管理
X 可能会造成应用程序延迟	✓ 独立捕获
X 可能会造成不稳定或失效关闭	✓ 独立的故障域

添加基于网络的情报分析，以便在开发人员忙于赶进度的同时，改善安全态势

提取未加密的数据后，可进一步利用 Gigamon Deep Observability Pipeline，将原始通信数据转换为流量级 Metadata 元数据记录以减少误报，帮助识别端口欺骗等恶意活动，并通过主动实时监控而不是被动取证来加快威胁检测。这种基于网络的情报分析不会受到日志修改的影响，适用于物联网和其他无代理设备，并且可以输入到安全运维团队和开发运维团队使用的可观测性工具中。

对于高度敏感的环境，UCT 还可以选择对发往深度可观测性管道的镜像通信进行重新加密，并在转发到相应工具之前屏蔽信用卡信息或个人身份信息 (PII) 等敏感数据。

用例

使用 Precryption 技术检测网络攻击

网络攻击，有时也称为勒索软件攻击，通常始于威胁行为者通过网络钓鱼或其他凭据收集技术获得离网访问员工笔记本电脑的权限。人们希望端点安全能够检测到或防止这种情况，但不幸的是，这种情况并不总是发生。

威胁行为者侵入网络后，可使用许多可用资源，包括运用复杂技术来删除日志、提升权限等，并寻找其他更重要的网络资源（例如主机、应用程序、工作负载等），窃取更多更敏感数据。只要有足够的时间和攻击途径，他们将能够渗透到其他这些网络资源中。这种技术被称为横向移动。

最终，威胁行为者将侵入更重要的应用程序，窃取数据并窃听通信。威胁行为者会慢慢地将数据抽取到网络中由他们控制的某个存储位置。他们会小心谨慎地进行抽取操作，以免影响性能或触发警报。当威胁行为者拥有足够的数据并准备就绪时，他们会实施最后一次快速且大规模的数据渗漏事件，将窃取的数据导出到外部，然后勒索该组织要钱。

在这种情形中，威胁行为者执行的活动主要有四种类型：

1. 使用网络钓鱼或凭据收集等初始手段来绕过端点安全防护措施
2. 在网络内进行横向移动
3. 将敏感数据缓慢地抽取到特定的存储位置
4. 发动快速的数据渗漏事件

以下概述了 Gigamon Precryption 技术所提供的明文可见性如何帮助工具检测此类威胁活动：

	安全工具的发现能力 - 无 Precryption	安全工具的发现能力 - 有 Precryption
网络钓鱼等初始手段	员工的常规活动	员工的常规活动
横向移动	无害噪声	已知攻击已部署并成功渗透到服务器中
数据抽取	无害噪声	正在通过未经授权的通道访问和传输 VIP 数据
数据渗漏	大规模数据传输	详细记录对被窃取的数据

要更详细地了解网络攻击场景，请下载信息图。

结论

获得对加密流量和 Metadata 元数据的可见性后，能够显著提升混合云的安全防护、监控能力和故障排查效率。Gigamon Deep Observability Pipeline 直击当今重大的安全挑战，能够监控本地和公有云中的虚拟流量和容器流量。GigaVUE UCT 提供强大的平台支持和一站式管理界面，应对云技术日益普及所带来的种种挑战。Gigamon 基于网络的的情报分析可为开发运维、云运维和安全运维等团队的安全工具提供高质量的 Metadata 元数据。Gigamon Precryption 技术则解决了当前一个特别棘手的难题，即如何在使用现代加密的云环境中监控隐蔽的活动，并且提供了巧妙的轻量级解决方案，旨在改善安全态势并有效抵御恶意攻击。

关于 Gigamon

Gigamon 提供深度可观测性管道，利用切实可行的基于网络的情报分析来增强可观测性工具的能力。这种强大的组合使 IT 组织能够满足安全性和合规性治理要求，加快对于性能瓶颈的根本原因分析，并降低在管理混合云和多云 IT 基础设施方面的运营开销，由此帮助许多现代企业充分实现了云技术所蕴含的转型潜力。Gigamon 为全球 4,000 多家客户提供服务，包括 80% 以上的“财富 100 强”企业、10 家最大移动网络提供商中的 9 家，以及全球数百个政府机构和教育机构。要了解更多信息，请访问 gigamon.com。

1. Shelley Boose.81% of Companies Have Had a Cloud Security Incident in the Last Year.Venafi, September 28, 2022. https://venafi.com/blog/81-companies-have-had-had-cloud-security-incident-last-year-venafi-research.
2. 2023 Hybrid Cloud Security Survey:Perception vs.Reality.Gigamon, 2023. https://www.gigamon.com/content/dam/gated/wp-gigamon-survey-hybrid-cloud-security-2023.pdf.
3. Internet Security Report – Q2 2021.Watchguard, 2021. https://www.watchguard.com/wgrd-resource-center/security-report-q2-2021.
4. Deepen Desai.Encrypted Attacks Rise 314%:New ThreatLabz State of Encrypted Attacks Report.Zscaler, October 28, 2021. https://www.zscaler.com/blogs/security-research/encrypted-attacks-rise-314.

Gigamon®

全球总部
3300 Olcott Street, Santa Clara, CA 95054 USA
+1 (408) 831-4000 | gigamon.com

09.23_01