OWASP
Open Worldwide Application Security Project

2023
OWASP 机器学习安全风险 TOP 10

致谢

项目领导

- Abraham Kang
- Shain Singh
- Sagar Bhure
- Rob van der Veer

贡献者

- Vamsi Suman Kanukollu
- M S Nishanth
- Buchibabu Bandarupally
- Jamieson O’ Reilly
- Ashish Kaushik
- Jakub Kaluzny
- David Ottenheimer
- Haral Tsitsivas

感谢以下中文项目参与人员（按拼音顺序排列）：

戴楚南、吴楠、肖文棣、张坤、张淼

英文项目地址：

https://owasp.org/www-project-machine-learning-security-top-10/

十大风险列表

ML01:2023	对抗性攻击 Adversarial Attack	当攻击者故意更改输入数据以误导模型时，就会发生对抗性攻击。
ML02:2023	数据投毒攻击 Data Poisoning Attack	当攻击者操纵训练数据导致模型以不良方式运行时，就会发生数据投毒攻击。
ML03:2023	模型反转攻击 Model Inversion Attack	当攻击者对模型进行逆向工程以从中提取信息时，就会发生模型反转攻击。
ML04:2023	成员推理攻击 Membership Inference Attack	当攻击者操纵模型的训练数据以使其行为暴露敏感信息时，就会发生成员推理攻击。
ML05:2023	模型窃取 Model Stealing	当攻击者获得对模型参数的访问权时，就会发生模型窃取攻击。
ML06:2023	损坏的组件包 Corrupted Packages	当攻击者修改或替换系统使用的机器学习库或模型时，就会发生损坏的组件包攻击。
ML07:2023	迁移学习攻击 Transfer Learning Attack	当攻击者在一个任务上训练模型，然后在另一个任务中对其进行微调，导致结果未按照预期产生，就会发生迁移学习攻击。
ML08:2023	模型偏斜 Model Skewing	当攻击者操纵训练数据的分布，导致模型以不希望的方式运行时，就会发生模型偏斜攻击。
ML09:2023	输出结果完整性攻击 Output Integrity Attack	当攻击者的目的是为了改变其 ML 模型的行为或对使用该模型的系统造成损害，从而修改或操纵 ML 模型的输出结果，就会发生输出结果完整性攻击。
ML10:2023	神经网络重编程 Neural Net Reprogramming	当攻击者操纵模型的参数使其以不良的方式运行时，就会发生神经网络重编程攻击。

ML01:2023 对抗性攻击 Adversarial Attack

风险图表 Risk Chart

威胁代理	攻击载体	安全弱点	影响
应用描述	可利用性：5	可检测性 :3	技术 :5
威胁代理：具有深度学习和图像处理技术知识的攻击者。 攻击向量：故意制作与合法图像相似的对抗性图像。		深度学习模型准确分类图像的能力存在漏洞。	图像分类错误，导致安全绕过或对系统造成损害。

攻击场景示例 Example Attack Scenario

场景 1：图像分类

训练深度学习模型将图像分类为不同的类别，例如狗和猫。攻击者创建了一个与猫的合法图像非常相似的对抗图像。该对抗图像带有一些精心设计的小扰动，导致模型将其错误分类为狗。当模型部署在真实环境中时，攻击者可以使用对抗图像绕过安全措施进而对系统造成危害。

场景 2：网络入侵检测

训练一个深度学习模型来进行网络入侵检测。攻击者通过精心制作数据包来创建对抗性网络流量，从而使它们能够逃避模型的入侵检测系统。攻击者可以操纵网络流量的特征，例如源 IP 地址、目标 IP 地址或负载，从而使入侵检测系统无法检测到它们。例如，攻击者可能会将其源 IP 地址隐藏在代理服务器后面或加密其网络流量的有效负载。这种类型的攻击可能会导致数据被盗、系统受损等严重后果。

预防措施 Example Attack Scenario

对抗性训练	防御对抗性攻击的一种方法是在对抗性示例上训练模型。这可以帮助模型对攻击的识别变得更加敏锐，并降低其被误导的可能性。
稳健模型	另一种方法是使用旨在抵御对抗性攻击的模型，例如对抗性训练或包含防御机制的模型。
输入验证	输入验证是另一个重要的防御机制，可用于检测和防止对抗性攻击。这涉及检查输入数据是否存在异常，例如意外值或模式，并拒绝可能是恶意的输入。

ML02:2023 数据投毒攻击 Data Poisoning Attack

风险图表 Risk Chart

威胁代理	攻击载体	安全弱点	影响
应用描述	可利用性：3	可检测性：2	技术：4
威胁代理人：有权访问用于模型的训练数据的攻击者。		缺乏数据验证和对培训数据的监控不足。	该模型将基于中毒数据做出错误的预测，从而导致错误的决策和潜在的严重后果。
攻击向量：攻击者将恶意数据注入训练数据集中。			

攻击场景示例 Example Attack Scenario

场景 1：训练垃圾邮件分类器

攻击者给一款用于将电子邮件分类为垃圾邮件或非垃圾邮件的深度学习模型的训练数据投毒，该模型用于。攻击者通过侵入网络或利用数据存储软件中的漏洞或其他破坏数据存储系统的方式，将恶意标记的垃圾邮件注入训练数据集来执行此攻击。攻击者还可以操纵数据标记过程，例如伪造电子邮件标记或贿赂数据标记者以提供不正确的标签。

场景 2：训练网络流量分类系统

攻击者给一款用于将网络流量分类为不同类别（例如电子邮件、Web 浏览和视频流）的深度学习模型的训练数据投毒。他们引入了大量网络流量示例，这些示例被错误地标记为不同类型的流量，导致训练模型将此流量分类为错误类别。因此当部署模型时，模型可能会进行错误的流量分类，从而可能导致网络资源分配不当或网络性能下降。

预防措施 Example Attack Scenario

数据验证和验证	确保训练数据在用于训练模型之前得到彻底验证和确认。这可以通过实施数据验证检查并使用多个数据标记器来验证数据标记的准确性来完成。
安全数据存储	以安全方式存储训练数据，例如使用加密、安全数据传输协议和防火墙。
数据分离	将训练数据与生产数据分开，以降低泄露训练数据的风险。

访问控制	实施访问控制以限制何人、何时可以访问训练数据。
监控和审计	定期监控训练数据是否有异常，并进行审计以发现任何数据篡改。
模型验证	使用训练期间未使用过的单独验证集来验证模型。这有助于检测可能影响训练数据的任何数据投毒攻击。
模型集成	使用训练数据的不同子集训练多个模型，并使用这些模型的集成来进行预测。因为攻击者需要破坏多个模型才能实现影响预测，这给攻击者加大了难度，从而减少数据投毒攻击的风险。
异常检测	使用异常检测技术来检测训练数据中的任何异常行为，例如数据分布或数据标签的突然变化。这些技术可用于及早检测数据投毒攻击。

ML03:2023 模型反转攻击 Model Inversion Attack

风险图表 Risk Chart

<table>
<tr><th colspan="2">威胁代理  攻击载体</th><th>安全弱点</th><th>影响</th></tr>
<tr><td>应用描述</td><td>可利用性：4</td><td>可检测性：2</td><td>技术：4</td></tr>
<tr><td colspan="2">威胁代理：可以访问模型和输入数据的攻击者。

攻击向量：向模型提交图像并分析模型的响应。</td><td>模型的输出可用于推断有关输入数据的敏感信息。</td><td>有关输入数据的机密信息可能会被泄露。</td></tr>
</table>

攻击场景示例 Example Attack Scenario

场景 1：从人脸识别模型中窃取个人信息

攻击者训练一款深度学习模型来执行人脸识别。然后，他们使用该模型对公司使用的不同人脸识别模型进行模型反转攻击。攻击者将某个人的图像输入到模型中，并从模型的预测中恢复这个人的个人信息，例如姓名、地址或社会安全号码。

此攻击方式为：攻击者通过训练模型 A 来执行人脸识别，然后使用 A 模型来反转另一个人脸识别模型 B 的预测，从而攻击者可以从 B 模型的预测中恢复个人信息。此类攻击可以通过利用模型开发中的漏洞或通过 API 访问模型来实现。

场景 2：绕过在线广告中的机器人检测模型

广告商希望通过使用机器人执行点击广告和访问网站等操作来实现广告活动的自动化。然而，在线广告平台使用机器人检测模型来阻止机器人执行这些操作。为了绕过这类模型，广告商训练针对机器人检测的深度学习模型，并使用该模型来反转在线广告平台使用的机器人检测模型的预测。广告商将他们的机器人输入到模型中，使机器人以人类用户的身份出现，从而能够绕过机器人检测，成功地执行自动广告活动。

此攻击方式为：广告商首先训练他们自己的机器人检测模型 A，然后使用 A 来逆转在线广告平台使用的机器人检测模式 B 的预测，从而广告商可以使他们的机器人看起来像人类用户，成功地实现了广告活动的自动化。此类攻击能够通过开发中的漏洞或通过 API 访问模型来实现。

预防措施 Example Attack Scenario

访问控制	限制对模型或其预测的访问可以防止攻击者获得反转模型所需的信息。这可以通过在访问模型或其预测时要求身份验证、加密或其他形式的安全性来实现。
输入验证	验证模型的输入可以防止攻击者提供可用于反转模型的恶意数据。这可以通过在模型处理输入之前检查输入的格式、范围和一致性来实现。
模型透明度	使模型及其预测透明有助于检测和防止模型反转攻击。这可以通过记录所有输入和输出、为模型的预测提供解释或允许用户检查模型的内部表示来实现。
定期监测	监测模型对异常的预测有助于检测和防止模型反转攻击。这可以通过跟踪输入和输出的分布、将模型的预测与实际真实数据进行比较或随时间的变化监测模型的性能。
模型再训练	定期对模型进行再训练可以有助于防止对“过时”模型的反转攻击，从而避免信息泄露。这可以通过结合新数据并纠正模型预测中的任何不准确之处来实现。

ML04:2023 成员推理攻击 Membership Inference Attack

风险图表 Risk Chart

威胁代理	攻击载体	安全弱点	影响
应用描述	可利用性：4	可检测性：3	技术：4
有权访问数据和模型的黑客或恶意行为者。		缺乏适当的数据访问控制。	模型预测不可靠或不正确。
有恶意或被贿赂干扰数据的内部人员。		缺乏适当的数据验证和净化技术。	失去敏感数据的机密性和隐私。
允许未经授权访问数据的不安全的数据传输通道。		缺乏适当的数据加密。	违反法律法规。
		缺乏适当的数据备份和恢复技术。	声誉损害。

攻击场景示例 Example Attack Scenario

场景 1：从机器学习模型推断财务数据

恶意攻击者想要获取个人的敏感财务信息。他们通过在财务记录数据集上训练机器学习模型，并使用它来查询特定个人的记录是否包含在训练数据中来做到这一点。然后，攻击者可以使用这些信息来推断个人的财务历史和敏感信息。

攻击者通过在从金融组织获得的财务记录数据集上训练机器学习模型来执行此攻击。然后，他们使用这个模型来查询特定个人的记录是否包含在训练数据中，从而推断出敏感的财务信息。

预防措施 Example Attack Scenario

在随机或混洗数据上进行模型训练	在随机或混合数据上训练机器学习模型会使攻击者更难确定训练数据集中是否包含特定示例。
模型混淆	通过添加随机干扰或使用差分隐私技术来混淆模型的预测，可以使攻击者更难确定模型的训练数据，从而有助于防止成员身份推理攻击。
正则化	L1 或 L2 等正则化技术有助于防止模型与训练数据的过拟合，这会降低模型准确确定训练数据集中是否包含特定示例的能力。

减少训练数据	减少训练数据集的大小或删除冗余或高度相关的特征有助于减少攻击者从成员推理攻击中获得的信息。
测试和监控	定期测试和监控模型的异常行为，可以通过检测攻击者何时试图访问敏感信息来帮助检测和防止成员身份推理攻击。

ML05:2023 模型窃取 Model Stealing

风险图表 Risk Chart

应用描述	可利用性：4	可检测性：3	技术：4
威胁代理 / 攻击向量：这是指执行攻击的实体，在这种情况下，它是指想要窃取机器学习模型的攻击者。		模型的不安全部署：模型的不安全部署使攻击者更容易访问和窃取模型。	模型窃取可能既影响用于训练模型的数据的机密性，同时也会影响开发模型的组织的声誉。 机密性、信誉

攻击场景示例 Example Attack Scenario

场景 1：从竞争对手那里窃取机器学习模型

原公司开发了一款有竞争优势的机器学习模型，竞争对手雇佣恶意攻击者为其工作。 攻击者希望可以窃取此模型，以便竞争对手公司可以拥有此模型并获得竞争优势。

攻击者通过反汇编二进制代码或访问模型的训练数据和算法，对原公司的机器学习模型进行逆向工程来执行此攻击。 一旦攻击者对模型进行了逆向工程，攻击者就可以使用此信息重新创建模型并开始满足竞争对手公司的业务需求。 这可能会给原公司造成重大经济损失，并损害 A 公司的声誉。

预防措施 Example Attack Scenario

加密	对模型的代码、训练数据和其他敏感信息进行加密，可以防止攻击者访问和窃取模型。
访问控制	实施严格的访问控制措施，例如双因素身份验证，可以防止未经授权的个人访问模型和窃取模型。
定期备份	定期备份模型的代码、训练数据等敏感信息，确保万一失窃时可以恢复。

模型混淆	混淆模型的代码并使其难以逆向工程可以防止攻击者窃取模型。
水印	在模型的代码和训练数据中添加水印可以追踪盗窃的来源并追究攻击者的责任。
法律保护	为模型提供法律保护，例如专利或商业秘密，可以使攻击者更难窃取模型，并可以在发生盗窃时提供法律诉讼依据。
监控和审计	定期监控和审计模型的使用可以通过检测攻击者何时试图访问或窃取模型来帮助检测和防止窃取行为。

ML06:2023 损坏的组件包 Corrupted Packages

风险图表 Risk Chart

应用描述	可利用性：5	可检测性：2	技术：4
恶意攻击者修改机器学习项目使用的开源包代码		依赖不受信任的第三方代码	对机器学习项目的损害和对组织的潜在危害

攻击场景示例 Example Attack Scenario

场景 1：攻击组织中的机器学习项目

恶意攻击者想要破坏大型组织正在开发的机器学习项目。攻击者知道该项目依赖于几个开源包和库，并想通过这些依赖的开源包和库找来破坏该项目。

攻击者通过修改项目所依赖的某个开源包（例如 NumPy 或 Scikit-learn）的代码来执行攻击。然后，攻击者将这个修改后的开源包的版本上传到公共存储库，例如 PyPI，供其他人下载和使用。当受害组织下载并安装软件包时，攻击者的恶意代码也会被安装，并可用于破坏项目。

因为受害者可能没有意识到自己使用的软件包已经损坏，所以在很长一段时间内忽视该类型攻击，导致产生严重后果，例如窃取敏感信息、修改结果，甚至导致机器学习模型失效。

预防措施 Example Attack Scenario

验证包签名	在安装任何组件包之前，验证组件包的数字签名以确保该安装组件包没有遭到篡改。
使用安全的组件包存储库	使用安全的组件包存储库，例如 Anaconda，该存储库执行严格的安全措施并对组件包进行审查。
保持软件包最新	定期更新所有软件包以确保修补任何漏洞。
使用虚拟环境	使用虚拟环境将组件包和库与系统的其余部分隔离开来。这样可以更容易地检测并删除任何恶意软件包。

执行代码审查	定期对项目中使用的所有组件包和库执行代码审查，以检测任何恶意代码。
使用组件包验证工具	在安装前，使用 PEP 476 和安全组件包安装等工具验证组件包的真实性和完整性。
教育开发人员	教育开发人员了解与损坏的组件包攻击相关的风险以及在安装前验证组件包的重要性。

ML07:2023 迁移学习攻击 Transfer Learning Attack

风险图表 Risk Chart

威胁代理	攻击载体	安全弱点	影响
应用描述	可利用性：4	可检测性：2	要求：4
具有机器学习知识和拥有访问训练数据集或预训练模型权限的攻击者		缺乏对训练数据集和预训练模型的适当数据保护措施。	机器学习模型错误或产生错误结果。
		预训练模型的不安全存储和共享。	训练数据集中敏感信息泄露的漏洞。
		缺乏对预训练模型和训练数据集的适当数据保护措施。	对组织的声誉损害。
			法律或法规遵从性问题。

攻击场景示例 Example Attack Scenario

场景 1：迁移恶意模型知识，试图绕过人脸识别系统

攻击者在包含被操纵的人脸图像的恶意数据集上训练机器学习模型，并想用此攻击一家安全公司用于身份验证的人脸识别系统。

然后，攻击者将模型知识迁移到目标人脸识别系统。目标系统开始使用攻击者操纵的模型进行身份验证。

因此，人脸识别系统开始做出错误的预测，使攻击者能够绕过安全机制，获取敏感信息。例如，攻击者可以使用自己恶意修改的人脸图像，系统会将其误识别为合法用户。

预防措施 Example Attack Scenario

定期监控和更新训练数据集	定期监控和升级训练数据集有助于防止恶意知识从攻击者的模型转移到目标模型。
使用安全可信的训练数据集	使用安全可信的训练数据集有助于防止恶意知识从攻击者的模型转移到目标模型。

实施模型隔离	实施模型隔离有助于防止恶意知识从一个模型转移到另一个模型。例如，将训练环境和部署环境分开可以防止攻击者将知识从训练环境转移到部署环境。
使用差分隐私	使用差分隐私有助于保护训练数据集中的个人隐私数据，并防止恶意知识从攻击者的模型转移到目标模型。
执行定期安全审计	定期安全审计可以通过识别消除系统中的漏洞来帮助识别和防止迁移学习攻击。

ML08:2023 模型偏斜 Model Skewing

风险图表 Risk Chart

应用描述	可利用性：5	可检测性：2	技术：4
模型偏斜攻击中的攻击者可能是恶意个人，也可能是在操纵模型结果方面第三方既得利益者。		模型无法准确反映训练数据的分布。这可能是由于数据偏差、不正确地数据采样或攻击者操纵数据或训练过程等因素造成的。	可能导致基于模型输出做出错误的决策。如果该模型用于医学诊断或刑事司法等关键应用，这可能会导致经济损失、声誉受损，甚至对个人造成伤害。

攻击场景示例 Example Attack Scenario

场景 1：提供虚假反馈信息，获得贷款批准机会

一家金融机构正在使用机器学习模型来预测贷款申请人的信用度，该模型的预测被集成到贷款审批流程中。攻击者希望增加获得贷款批准的机会，因此他们操纵 MLOps（Machine Learning Operations）系统中的反馈回路。攻击者向系统提供虚假的反馈数据，表明高风险的申请人过去曾被批准发放过贷款，且该反馈用于更新训练模型数据。因此，该模型的预测偏向于攻击者为低风险申请人，攻击者获得贷款批准的机会显著增加。

这种类型的攻击可能会损害模型的准确性和公平性，导致预料之外的后果，并对金融机构及其客户造成潜在伤害。

预防措施 Example Attack Scenario

实施强访问控制	确保只有授权人员才能访问 MLOps 系统及其反馈回路，并记录和审计所有活动。
验证反馈数据的真实性	使用数字签名和校验和等技术来验证系统接收到的反馈数据是否真实，并拒绝任何与预期格式不匹配的数据。
使用数据验证和清理技术	在使用反馈数据更新训练数据之前，先对其进行清理和验证，以最大限度地降低使用错误或恶意数据的风险。

实施异常检测	使用统计和基于机器学习的方法等技术来检测和警告反馈数据中的异常，这表明可能发生了攻击。
定期监控模型性能	持续监控模型性能，并将其预测与实际结果进行比较，以检测任何偏差或曲解。
持续训练模型	使用更新和验证的训练数据定期对模型进行再训练，以确保其持续反映最新趋势和信息。

ML09:2023 输出结果完整性攻击 Output Integrity Attack

风险图表 Risk Chart

威胁代理 → 攻击载体		→ 安全弱点	→ 影响
应用描述	可利用性：5	可检测性：3	技术：3
可以访问模型输入和输出的恶意攻击者或内部人员		由于缺乏适当的鉴权及身份验证校验授权的措施，所以不能确保模型输入输出的完整性。	此风险可导致用户对模型的预测与结果失去信心。
		因对模型的输入输出缺乏充分的验证及鉴权，不能防止输入输出的结果被篡改。	如果模型的预测结果被用于做出重要的决定或决策，可能导致经济损失或名誉声誉受损。
有权获取输入和输出并可能篡改输入和输出以实现特定结果的第三方实体		缺少对模型输入输出的监控及相关日志的记录，导致无法检测出模型是否被第三方篡改。	如果模型在关键 / 重要的应用场景中使用，如金融诈骗的检测或网络安全场景中会存在安全风险。

攻击场景示例 Example Attack Scenario

场景 1：攻击 ML 模型篡改输出结果

攻击者能够访问到医院用于诊断疾病的 ML 模型及其输出结果，攻击者篡改模型的输出结果，使其给患者提供了一个错误的诊断结果。

因此，病人拿到了错误的治疗方法或处方，最终导致病人受到进一步的伤害甚至死亡。

预防措施 Example Attack Scenario

使用加密方式 / 算法	使用数字签名或安全的 hashes 对输出结果进行验证保护。
安全通信隧道 / 协议	在模型与输出结果的接口之间应该使用安全的协议进行通信的保护（如：SSL/TLS）。

输入验证	在提供结果反馈的位置应该进行输入参数的检测以检查未预期或被操纵的参数。
日志防篡改	将所有输入输出的交互及响应结果进行日志记录及使用防篡改技术保护日志的信息。
定期更新软件	定期更新软件和安全补丁以修复漏洞和降低输出完整性攻击的风险。
监视和审计	定期监视和审计结果以及模型和接口之间的交互行为可以帮助检测任何可疑活动并做出相应的响应。

ML10:2023 神经网络重编程 Neural Net Reprogramming

风险图表 Risk Chart

威胁代理	攻击载体	安全弱点	影响
应用描述	可利用性：4	可检测性：3	技术：3
拥有知识和资源且带有恶意的个人或组织操纵深度学习模型。		对模型的代码和参数缺少访问控制。	模型的预测可以被人为操纵以达到预期的结果。
			可以提取模型中的机密信息。
具有恶意行为的人员在组织内部开发深度学习模型。		缺乏适当的编码安全实践。	基于模型预测结果的决策可能会受到负面的影响。
		对模型活动的监控和记录不足。	组织的声誉和信誉会受到影响。

攻击场景示例 Example Attack Scenario

场景 1：改变模型参数，导致误识别支票手写名字

有这样一个场景：银行正使用机器学习模型来识别支票上的手写文字以实现自动结算。该模型在一个手写文字的大型数据集上进行了训练，它被设计成基于特定参数（如大小、形状、斜度和间距）以准确识别字符。

攻击者利用神经网络重编程攻击可以通过改变模型训练数据集里的图像或直接修改模型中的参数来操纵模型的参数。此类攻击可能导致模型被重新编程以识别不同的字符。例如，攻击者可以更改参数，使模型将字符“5”识别为字符“2”，从而导致处理错误的金额。

攻击者可以利用此漏洞将伪造的支票引入结算过程中，由于参数被恶意操纵，模型将伪造的支票处理为有效的。这会给银行带来重大的经济损失。

预防措施 Example Attack Scenario

正则化	在损失函数（loss function）中添加 L1（Lasso）或 L2（Ridge）的正则化技术有助于防止过拟合以减少神经网络重编程攻击的机会。
鲁棒（健壮）模型设计	设计具有稳健性的架构和激活函数（Activation Function）的模型可以有助于减少有效利用重编程攻击的机会。
加密技术	加密技术可用于保护模型的参数和权重，防止未经授权的访问或操纵这些参数。

说 明

1.OWASP Risk Rating Methodology（摘取部分）

https://owasp.org/www-community/OWASP_Risk_Rating_Methodology 风险 = 可能性 * 影响

1. 估计可能性因素

脆弱性因素

此组因素与所涉及的漏洞有关。这里的目标是估计特定漏洞涉及被发现和利用的可能性。

- “易于利用”：

这组威胁代理实际利用此漏洞的难易程度如何？（数值越高，利用漏洞的可能性越大）

0	1	2	3	4	5	6	7	8	9
	理论		困难		简单				自动化工具可用

- 入侵检测：

检测到漏洞的可能性有多大？（数值越高，检测到漏洞的可能性越小）

0	1	2	3	4	5	6	7	8	9
	应用程序中的主动检测		记录和审查					记录而不审查	未记录

2. 估计影响的因素

在考虑成功攻击的影响时，重要的是要意识到存在两种影响。首先是 “技术影响”， 对应用程序、应用程序使用的数据以及提供的功能的影响。另一个是“业务影响”，即是对业务和运营该业务的公司影响。

根本上来说，业务影响更为重要。但是，您可能无法收集到所有成功攻击后的业务后果。在这种情况下，尽可能多提供有关技术风险的详细信息，将有助于相关风险管理人员做出有关业务风险的决策。

技术影响

以“完整性损失”为例 - 有多少数据可能被损坏，损坏程度如何？

0	1	2	3	4	5	6	7	8	9
	最小轻微损坏的数据		最小严重损坏的数据		大量轻微损坏的数据		大量严重损坏的数据		所有数据完全损坏

2. 其他说明

1. 风险图表

注意，描述图表只是一个基于场景示例的情况，实际风险评估将取决于每个机器学习系统的具体情况。

2. 可利用性特定于 ML

ML01:2023	对抗性攻击 Adversarial Attack	ML 特定应用程序：4 ML 特定操作：3
ML02:2023	数据投毒攻击 Data Poisoning Attack	ML 特定应用程序：4 ML 特定操作：3
ML03:2023	模型反转攻击 Model Inversion Attack	ML 特定应用程序：5 ML 特定操作：3
ML04:2023	成员推理攻击 Membership Inference Attack	ML 特定应用程序：5 ML 特定操作：3
ML05:2023	模型窃取 Model Stealing	ML 特定应用程序：4 ML 特定操作：3
ML06:2023	损坏的组件包 Corrupted Packages	ML 特定应用程序：5 ML 特定操作：3
ML07:2023	迁移学习攻击 Transfer Learning Attack	ML 特定应用程序：4 ML 特定操作：3
ML08:2023	模型偏斜 Model Skewing	ML 特定应用程序：4 ML 特定操作：3
ML09:2023	输出结果完整性攻击 Output Integrity Attack	ML 特定应用程序：4 ML 特定操作：4
ML10:2023	神经网络重编程 Neural Net Reprogramming	ML 特定应用程序：4 ML 特定操作：4

- ML 特定应用程序：4

该攻击专门针对机器学习应用程序，可能对模型和组织造成重大伤害

- ML 特定操作：3

该攻击需要机器学习操作的知识，但可以相对轻松地执行

获取更多资讯