人工智能赋能网络威胁研究进展

王志[1,2], 尹捷[2], 崔翔[3], 刘奇旭[1,2], 刘潮歌[1,2], 汪旭童[1,2]

[1]中国科学院大学网络空间安全学院 北京 100049

[2]中国科学院信息工程研究所 北京 100093

[3]广州大学网络空间先进技术研究院 广州 510006

摘要 近年来，人工智能赋能的网络威胁日趋增多。为了理解这类威胁的原理、研究其防御方法，本文对现有的人工智能赋能的网络威胁案例进行了梳理和综述。本文首先分析了人工智能的能力和神经网络模型的性质，总结了人工智能对网络威胁的五种赋能作用，包括伪造与欺骗、隐蔽与隐匿、感知与决策、定向与定制、规模化与自动化，并在此基础上形成了人工智能对网络威胁的赋能矩阵。随后，本文将现有的人工智能赋能的网络威胁案例归纳为 18 个类别，并结合网络杀伤链模型构建智能化网络威胁框架，从网络攻击的准备、入侵和执行三个阶段对相关案例进行介绍和分析。随后，本文从攻防能力角度分析现有防御方法的有效性与局限性，指出智能化网络威胁与其他威胁的不同，并从场景、技术和系统三个维度提出针对性防御建议。最后，本文结合人工智能领域与网络威胁领域的技术方向，对智能化网络威胁的发展趋势进行展望，对人工智能在网络威胁中的作用进行深入探讨，以期对未来相关防御研究提供有益参考。

关键词 人工智能；网络威胁；人工智能安全；网络空间安全

中图法分类号 TP309

AI-Powered Cyber Threats: A Survey

WANG Zhi[1,2], YIN Jie[2], CUI Xiang[3], LIU Qixu[1,2], LIU Chaoge[1,2], WANG Xutong[1,2]

[1] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, 100149, China

[2] Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China

[3] Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou, 510006, China

Abstract In recent years, the threat of artificial intelligence (AI) in cyber attacks has been continuously increasing. In order to help researchers quickly understand the relevant principles and conduct research on defense methods, it is necessary to analyze the characteristics of AI and the principles of AI-powered cyber threats and sort out relevant cases. To this end, we analyzed the capabilities of artificial intelligence and the nature of the neural network model and divided the roles of artificial intelligence in cyber threats into five categories: forgery and deception, stealth and anonymity, perception and decision-making, targeting hand customization, and scale and automation. On this basis, the influence of the characteristics of artificial intelligence on the five roles is analyzed, and the enabling matrix of artificial intelligence to cyber threats is formed. Then, we collected the existing AI-powered cyber threat works and classified the cases into 18 categories. Combined with the cyber kill chain, an AI-powered cyber threat framework is formed, and the cases of AI-powered cyber threats are introduced based on the three stages of an attack: preparation, intrusion, and execution. The principles, effects, and progress of representative works in each category are concluded, as well as their strength and limitation. Subsequently, we analyzed the effectiveness and limitations of existing defense methods from the perspective of the capabilities of attackers and defenders, pointed out the difference between AI-powered cyber threats and other threats, and put forward the possible defense measures targeted at AI from three dimensions of scenario, technology, and system. Combined with the evolution of AI technology and cyber threats and the deficiencies and constraints of AI-powered cyber threats, we discussed the effectiveness of AI in cyber threats and prospected the future trends of AI-powered cyber threats. We hope this paper will help defend against AI-powered cyber threats in the future.

Key words artificial intelligence; cyber threat; AI security; cyber security

通讯作者：尹捷，博士，工程师, Email: yinjie@iie.ac.cn 。

本课题得到国家自然科学基金项目(No.61902396)；中国科学院青年创新促进会(No.2019163)；中国科学院战略性先导科技专项项目(No.XDC02040100)；中国科学院网络测评技术重点实验室和网络安全防护技术北京市重点实验室资助。

收稿日期：2022-04-XX；修改日期：201X-X-X；定稿日期：201X-X-X

1 引言

随着计算设备性能的大幅提升以及大数据和云计算的普及，以深度神经网络为代表的人工智能（Artificial Intelligence, AI）技术在自动驾驶、智慧城市、医学图像等多个领域取得巨大突破，推动各领域向智能化演进。与此同时，网络空间安全领域也积极探索与人工智能技术的结合，研究人员不断利用人工智能技术来提升网络安全防御能力，在漏洞挖掘、身份识别、威胁发现、入侵检测、追踪溯源等方向卓有成效。然而，人工智能技术在助力网络防御的同时，也为网络空间来了新的威胁。2018年，26 位来自不同研究机构的科学家联名发布了针对人工智能恶意利用的报告[1]。他们结合 AI 技术进展对恶意利用 AI 可能导致的数字安全、物理安全和政治安全问题做了预测。安全厂商 BeyondTrust[2]在网络安全发展趋势预测中指出，机器学习训练数据污染、AI 武器化的泛滥将给网络安全带来巨大挑战。Gartner[3]则表示，到 2022 年，30%的网络安全问题将与人工智能安全有关，人工智能趋于工程化。Fortinet[4]也认为，借助于多形态恶意代码的进化、集群攻击和人工智能的武器化将成为趋势。

近年来，研究人员针对人工智能技术与网络安全威胁进行了大量研究工作。2018 年，来自 IBM 的研究人员在世界著名的黑客大会 Black Hat 上展示了 AI 赋能的高度定向的网络攻击场景 DeepLocker[5]，该场景下的恶意代码具备自动精准识别攻击目标的能力，同时可以抵抗住来自安全人员的分析，隐藏攻击意图。Takaesu[6]提出了一种基于强化学习的自动化渗透测试工具 DeepExploit，DeepExploit 可以自动化地开展信息探测和漏洞发现，并完成渗透测试。Fortinet 的研究人员在 RSA 大会上提出了智能蜂群网络的概念[7]。智能蜂群网络借助去中心化的结构，利用集群智能实现信息共享和自主决策，通过自动化攻击工具和分布式节点，完成智能化对抗。

当前，人工智能已能够覆盖到网络攻击的全生命周期。为了深入剖析基于 AI 的网络威胁原理，帮助研究人员预测未来网络威胁的形态、建立有针对性的防御措施，本文对基于 AI 的网络威胁案例进行了梳理，对 AI 在网络攻击中起到的作用进行了分析，主要工作包括：

- 总结分析 AI 的能力与 AI 模型的性质，把 AI 对网络威胁的赋能作用分为伪造与欺骗、隐蔽与匿名、感知与决策、定向与定制和规模化与自动化等五种类型，形成 AI 网络威胁赋能矩阵；
- 以网络杀伤链模型为主线，梳理基于 AI 的网络安全威胁的案例，总结 18 种基于 AI 的网络威胁场景，形成基于 AI 的网络威胁框架；
- 提出从场景、技术和系统等三个角度对基于 AI 的网络威胁的防御建议，并对未来的智能化网络威胁发展趋势、AI 对网络威胁的影响进行分析和讨论。

需要说明的是，本文聚焦于人工智能赋能的网络威胁案例，主体是网络威胁，因此不包含基于 AI 的防御类工作、针对 AI 自身安全问题的工作及将 AI 技术应用在信息系统之外的工作（如机器人、实体武器威胁）等。具体地，本文的讨论是在针对信息系统的网络威胁活动中，对攻击起促进作用的 AI 任务，威胁对象包括攻击各阶段要面对的信息系统组件、网络威胁检测产品及信息系统管理人员等。

2 AI 的内生特性

AI 对网络威胁的赋能作用取决于 AI 的内生特性。AI 的内生特性可以增强恶意代码的能力，从而引发新的网络威胁。本章从 AI 的能力和 AI 模型的性质两个方面分析 AI 的内生特性。

2.1 AI 的能力

针对 AI 能力的研究，Ching[8]根据 AI 的应用场景将其分为 6 类，分别是个性化和分析（Personalization and profiling）、预测（Predictions）、模式识别和异常检测（Pattern recognition and anomaly detection）、自然语言（Natural language）、对象识别（Object identification）和目标达成（Goal achievement）。Sahota[9]将 AI 的能力分为三类，分别是机器学习能力、自然语言处理能力和交互能力。

聚焦于计算机领域，Schmid[10]等将 AI 的能力分为四类，分别是感知（Sense）、处理与理解（Process and Understand）、行动（Act）和交流（Communicate），并对每一类进行了划分。邱锡鹏[11]根据 AI 对人类能力的模拟，将 AI 的能力分为感知、学习和认识三类。感知是对外部刺激信息进行感知和加工，主要包括对文字、语音、图像等视频等连续或非连续信号进行处理。学习表示在与环境或样例的交互中进行学习，主要包括无监督学习、有监督学习和强化学习等。认知代表从已有知识进行推理、规划和决策等，主要包括自然语言处理、知识表示等。

本文借鉴文献[11]的 AI 能力分类方法，当 AI 的感知、学习和认识等能力应用到网络安全领域时，可以赋予网络威胁以学习能力和决策能力：

学习能力指攻击者从结构化、非结构化数据中

通过正负反馈学习数据的统计规律、特征分布或最优路径等，并将规律用于攻击任务中。例如，AI 可以从良性应用中学习其流量特征，并把恶意软件的流量特征转换为良性应用的流量特征，以规避基于流量的机器学习检测器。

决策能力指攻击者可以根据已学习到的知识，对未知数据或事件进行处理、分析与决策，使恶意代码能够根据环境变化指导恶意活动的状态变化。例如，AI 可以帮助恶意代码找到攻击目标，执行定向攻击；集群智能恶意代码可以根据环境的变化来确定节点状态和下一步的攻击任务等。

2.2 AI 模型的性质

在近些年 AI 领域的成果中，以深度神经网络（Deep Neural Network，DNN）为结构的工作占据主要部分，且 DNN 模型的特性已在攻击案例中起到主导作用，因此本节以 DNN 模型为代表，对 AI 模型的性质进行分析。

首先定义一组概念。定义 $f_M(\cdot): X \to Y$ 为将神经网络模型 M 由输入数据 X 转换为输出数据 Y 的处理；$S(\cdot): A,B \to L$ 为计算两个同维度变量 A 和 B 的相似度 L 的函数。已知高维空间内一点 P，若在高维空间内的一点 P' 满足 $L = S(P,P') = 0$，则称 P 与 P' 相同，即 $P = P'$；若 $L = S(P,P') < \delta, \delta > 0$，其中 δ 是阈值，则称 P 与 P' 相似。

2.2.1 单向性

给定神经网络模型 M 和输入 X，可以以较低的计算量得到输出 $Y = F_M(X)$，而给定模型 M 和输出 Y，逆向得到一个输入 $X' = F_M^{-1}(Y)$，使得 $L = S(X',X) < \delta, \delta > 0$ 较为困难，其中 δ 是阈值。

2.2.2 抗碰撞性

给定神经网络模型的一个输入 X_1 和对应的输出 Y_1，找到另一个输入 X_2 满足 $S(X_1,X_2) > \delta_1, \delta_1 > 0$，使得对应的输出 Y_2 满足 $S(Y_1,Y_2) < \delta_2, \delta_2 > 0$ 较为困难，δ_1 和 δ_2 是阈值。

2.2.3 容错性

给定神经网络模型的两个相似的输入 X 和 X'，即 $L_1 = S(X,X') < \delta_1, \delta_1 > 0$，可以得到两个相似的输出 Y 和 Y'，即 $L_2 = S(Y,Y') < \delta_2, \delta_2 > 0$。特别地，在一定条件下，若 X 和 X' 满足 $L_1 = S(X,X') < \varepsilon < \delta_1, \varepsilon > 0$，则可以得到两个相同的输出 $Y = Y'$，即 $L_2 = S(Y,Y') = 0$。

2.2.4 定向性

已知一种神经网络结构，给定具备不同参数 W_1 和 W_2 的神经网络模型 M_{W1} 和 M_{W2}，给定同一输入 X，可以得到两个不同的输出 $Y_1 = f_{M_{W1}}(X)$ 和 $Y_2 = f_{M_{W2}}(X)$，即 $S(Y_1,Y_2) > \delta, \delta > 0$。

2.2.5 泛化性

给定以数据集 D_{train} 训练的神经网络模型和属于类别 C 的样本 X_1，满足 $X_1 \in D_{train}$ 且 $Y_1 = f(X_1) \to C$，则对于同属于类别 C 的样本 $X_2 \notin D_{train}$ 且 $S(X_1,X_2) > \delta, \delta > 0$，模型能以一定的概率 P 使得 $Y_2 = f(X_2) \to C$，其中 P 为模型的准确率。

2.2.6 冗余性

冗余性主要指神经网络模型的结构复杂性、神经元复杂性和参数复杂性。在神经网络设计时，在能完成某一任务所需的最低网络层和神经元以外而引入的其余网络层及神经元构成了神经网络模型的冗余性，即以较多的网络层和神经元完成较简单的任务所带来的神经网络模型结构的冗余性。

2.2.7 弱解释性

弱解释性主要指由神经网络模型的复杂性带来的对神经网络运算与决策的不透明性，或难以通过数学证明等方式解释神经网络的推理与决策过程。

3 AI 对网络威胁的赋能矩阵

3.1 AI 在网络威胁中的作用

3.1.1 伪造与欺骗

攻击方借助 AI 增强的学习能力，通过模仿其他系统的成员的静态特征和行为特征，来创造或改造恶意对象的相关属性，将恶意对象伪造成来自其他系统的成员，进而欺骗第三方检测系统或相关人员。代表性案例有通过改造恶意代码或流量的特征来绕过黑盒检测器、模仿他人邮件内容进行钓鱼以及伪造人像图像和音视频的 DeepFake[12]等。

3.1.2 隐蔽与匿名

攻击方借助 AI 增强的能力及 AI 模型的性质，通过形态变换等方式，改造恶意对象的相关属性，将恶意对象混淆于良性应用中，以达到隐藏自身行为、增强溯源难度和规避检测的目的。隐蔽与匿名和伪造与欺骗有重合的部分。例如，通过学习良性应用的通信特征，改造自身通信特征的恶意软件，

在实现伪造与欺骗的同时，也实现了自身的隐匿。其他案例有借助神经网络进行隐蔽寻址、借助神经网络模型隐蔽投递恶意载荷以及借助神经网络隐藏攻击意图等。

3.1.3 感知与决策

攻击方借助 AI 增强的学习和决策能力，对环境中出现的文字、图像、视频、音频及其他数字信号进行处理，以理解环境状态，并根据已学习到的知识和环境反馈，对下一步的行动给予指导。代表性案例有基于机器学习方法的验证码破解、防火墙规则探测、攻击目标选择以及集群智能僵尸网络等。

3.1.4 定向与定制

攻击方借助 AI 增强的能力及 AI 模型的性质，根据预设规则，有针对性地选择和识别攻击目标、定制开发攻击载荷，以达到攻击效果和收益的最大化。代表性案例有基于机器学习的鱼叉式钓鱼、定向网络攻击以及用户追踪等。

3.1.5 规模化与自动化

攻击方结合 AI 增强的能力，将需要人工干预指导的攻击任务大规模、自动化地完成，进而扩大攻击影响范围，提升攻击收益。例如，自动化的鱼叉式钓鱼、防火墙规则探测以及凭证窃取等，都将需要人工干预的部分交给神经网络来完成，大大提升攻击效率。

当攻击者能力得到增强后，便能将这些能力应用于网络威胁中。需要注意的是，攻击者可以同时将多项能力应用于同一网络威胁任务中，比如在 DeepLocker 中，隐蔽与匿名、感知与决策和定向与定制三种能力都对攻击的成功实施起主导作用，而在 AI 赋能的命令控制场景 DeepC2[13]中则主要是隐蔽与匿名和感知与决策。

3.2 赋能矩阵

AI 的内生特性能够支撑其在网络威胁中的作用点。比如，单向性保证了模型的推理过程不可逆，可以对抗分析；抗碰撞性使得枚举输入实体变得困难，能避免对攻击目标的爆破；容错性能保证对攻击目标的准确识别；定向性有助于攻击资源的重复性利用，降低攻击成本；泛化性保证模型从训练环境转移到真实环境后的可用性；冗余性增强模型的健壮性；弱解释性则可以增强恶意代码对抗分析的能力，隐藏攻击意图和触发条件。

AI 对网络威胁的赋能效应往往与其内在特性紧密相关。将 AI 的作用点和其内生特性相关联，可以更好地分析 AI 对网络威胁的赋能作用，帮助了解 AI 特性在网络威胁中起到的相应作用。本文根据相关案例的应用情况，分析其中 AI 的作用点和内生特性的关系，形成 AI 对网络威胁的赋能矩阵，形如图 1。关于赋能矩阵的赋值，本文以相关案例为依托，根据案例在真实场景中的应用情况，把 AI 内生特性对网络威胁的赋能效果划分为三个等级，如表 1。

表1 AI 内生特性对网络威胁的赋能效果等级

Table 1 Level of Influence of AI Endogenous Characteristics on Cyber Threats

等级	影响
3	相关案例已形成成熟工具或已在真实场景中出现
2	相关案例尚未在真实场景中应用，但已形成方法或原型系统
1	尚未有实际案例，但存在应用于网络威胁中的可能

伪造与欺骗	隐蔽与匿名	感知与决策	定向与定制	规模化与自动化	
3	2	3	3	3	学习能力
2	2	3	2	3	决策能力
1	2	1	1	1	单向性
1	2	1	1	1	抗碰撞性
1	1	3	2	2	容错性
3	1	1	2	1	定向性
1	2	3	3	3	泛化性
1	2	1	1	1	冗余性
1	2	1	1	1	弱解释性

图1 AI 对网络威胁的赋能矩阵

Figure 1 Enabling Matrix of AI for Cyber Threats

本文综合现有的案例，逐一分析案例中的 AI 作用点与内生特性的关系，并依据影响力等级进行赋值。在赋值过程中，同一结合点具备不同数值的，取其最大值。最终形成的 AI 对网络威胁的赋能矩阵如图 1 所示。下面以 DeepLocker[3]为例，简要阐述赋能矩阵的形成过程。

在 DeepLocker 中，AI 的主要作用点在隐蔽与匿名、感知与决策和定向与定制三个方面。其中，隐蔽与匿名负责隐藏攻击触发条件，感知与决策负责处理与目标相关的特征，定向与定制负责对目标特定属性的维护，三者共同作用来完成攻击任务。支撑隐蔽与匿名的特性有 AI 模型的单向性、抗碰撞性和弱解释性，它们防止了对攻击目标的爆破与逆向；支撑感知与决策的特性有 AI 的学习能力和决策能力，它们负责对外部输入的图像、声音、位置等信息进行处理；支撑定向与定制的特性有 AI 模型的容错性和泛化性，它们支撑了对特定目标的寻找任务。由于 DeepLocker 已形成一套定向攻击的方法论，但

尚未出现在真实攻击中出现，因此对应属性的结合点赋值为 2。继续分析其他案例，并不断更新各结合点的取值，最终得到图 1 的赋能矩阵。

AI 对网络威胁的赋能矩阵对理解 AI 网络威胁和指导防御工作有着重要作用。赋能矩阵能更直观地展现出 AI 在不同场景中的作用。通过对具体案例的分析，防御者能够知道 AI 的哪个特性赋予了网络威胁什么样的能力，因此可以更具针对性地寻找防御策略。由于不同用户对网络资产的重要性有不同理解，因此在对同一网络威胁进行分析时，会得出不同的矩阵数值。在当前研究阶段，尚缺乏足够的数据来量化描述赋能矩阵，因此本文仅给出定性描述来大致体现出赋能效果的差异性，在应用时还需结合实际情况进行研判和调整。

4 基于 AI 的网络威胁案例

本章以网络杀伤链模型为主线对基于 AI 的网络威胁案例进行介绍。

4.1 攻击模型

网络攻击模型是从攻击者视角出发的网络威胁行为建模和攻击场景表示[14]。常见的攻击模型有杀伤链模型[15]、钻石模型[16]、MITRE ATT&CK[17]、NSA 10 Step[18]等。本文聚焦于网络威胁生命周期的各个阶段，故采用杀伤链模型为主线。

杀伤链模型是对一个完整的网络攻击过程的总结，包含侦察、武器化、投递、利用、植入、命令与控制、行动等七个环节。根据攻击者在不同环节的行为与目的，可以将网络攻击划分为三个阶段，分别为准备阶段、入侵阶段和执行阶段，如图 2。

图2 基于杀伤链模型的攻击阶段划分

Figure 2 Attack Stages Based on the Cyber Kill Chain

- 准备阶段包括侦察和武器化环节，主要目的是搜集目标信息，定制攻击工具。
- 入侵阶段包括投递、利用和植入环节，主要目的是突破网络边界，进入目标内部，进行漏洞利用并植入恶意代码，在目标系统建立堡垒。
- 执行阶段包括命令与控制和行动环节，主要目的是在目标系统内部完成攻击任务。

攻击者在每个阶段都存在能力限制。比如，在准备阶段，攻击者要进行目标探测和工具定制，需要完成目标发现、信息收集、钓鱼、水坑、漏洞扫描等任务，但此阶段过度依赖人为选择，缺乏对目标的自主发现和对环境的自适应能力；在入侵阶段，攻击者接入目标系统，要保证恶意代码的生存，需要完成权限提升、漏洞利用、恶意代码植入等任务，此阶段存在对环境的感知能力不足及对抗分析能力不足的特点；在执行阶段，攻击者的主要任务是控制恶意代码、执行攻击任务，此阶段需要完成命令控制、检测对抗、数据回传等任务，主要面临隐蔽性差和抗检测能力不足的问题。

AI 能在一定程度上解决这些问题。比如，在准备阶段，AI 能够自动化地选择高价值的攻击目标，弥补自主发现目标的短板；在入侵阶段，AI 能助力恶意代码发现漏洞，并实现免杀；在执行阶段，AI 能增强恶意代码的隐蔽性，增强溯源难度。本文将现有的 AI 与网络威胁结合的案例进行汇总，根据威胁目的、手段和影响将案例划分为 18 个类别，并对应到杀伤链模型的各阶段。随着技术的发展，未来可能会出现新的基于 AI 的网络安全威胁类型，本文所划分的类别不一定能将其完全覆盖，届时本文的分类会有更新和补充。由于各个类别均有较多案例，本章仅选取部分有代表性的工作进行介绍。值得注意的是，攻击者对目标系统外部信息的掌控程度较高，而防御者对外部信息的控制能力较弱，因此攻击者在准备阶段的可利用资源更加充分，行动方法更加丰富，其相关研究工作也多于入侵和执行阶段。相反，在入侵阶段和执行阶段，恶意载荷脱离攻击者环境或位于目标系统内部，攻击者往往得不到充分的数据和内部资源来对恶意载荷进行智能化训练，故在这两个阶段的相关成果相对较少。

将现有案例与 AI 内生特性、赋能矩阵、杀伤链模型相结合，可形成智能化网络威胁框架，如图 3。

4.2 准备阶段

4.2.1 侦察

侦察是攻击者在实施攻击前针对目标身份、目标系统或目标网络域开展的一系列探测和分析活动。当前，人工智能技术被广泛的应用在侦察过程中，本文将这些工作总结为 5 类技术场景，分别是目标定位、密码分析、敏感信息恢复、攻击路径选择和侧信道信息探测。

（1）目标定位

网络空间中充满了大量个人信息，尤其是社交网络平台，通过人工智能技术将这些有价值的信息进行整合和关联，能够帮助攻击者快速选择高价值的攻击目标，并对目标进行定位和追踪，进而采取具备针对性的攻击方法。

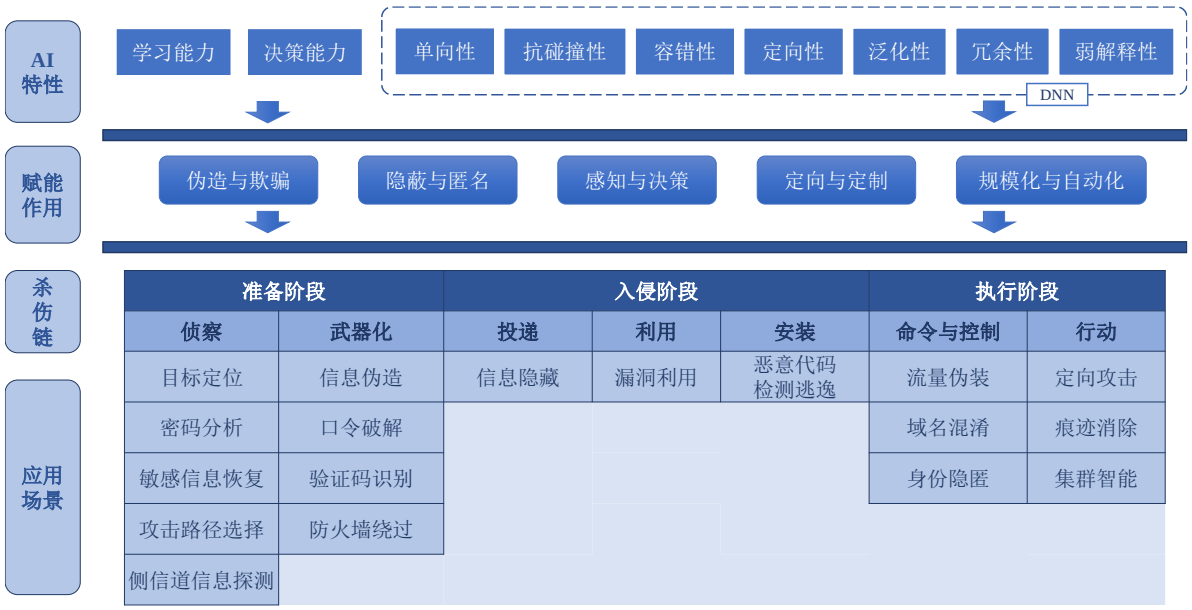


图3 智能化网络威胁框架

Figure 3 AI-powered Cyber Threat Framework

研究表明[19]，同一用户通常在不同网络平台使用相同或相似的用户名，可以通过用户名相似度的方法进行跨平台用户关联。同样的，同一用户在不同平台上预留的个人信息（如性别、出生日期、所在地等）也有很高的相似度[20]，可用来进行聚类。EagleEye[21]通过用户照片和用户名来进行跨平台的目标追踪。使用者仅需输入目标的图片和用户名线索，EagleEye 即可根据人脸识别等方法从 Instagram、Facebook、YouTube 和 Twitter 等平台的用户资料页爬取相关数据进行用户追踪和关联。

2016 年，Seymour[22]在 Black Hat 大会上展示了一种定向化钓鱼框架 SNAP_R（社交网络自动侦察钓鱼，Social Network Automated Phishing with Reconnaissance）。SNAP_R 通过聚类方法从社交网络平台上自动筛选潜在的目标用户，并学习目标用户的行为习惯，包括上线时间、内容喜好等，通过生成符合对方兴趣的定制化钓鱼文章，在对方的活动时间发布推文并@目标，把钓鱼文章推送给目标用户。实验表明，这种钓鱼手段与传统的大规模钓鱼方法相比，成功率由 5-14%提高到了 30-66%。

2021 年，Liu 等[23]提出了基于兴趣地点和查询似然模型的微博用户位置推断方法 PaQL。PaQL 借鉴信息检索领域的逆文档频率（Inverse Document Frequency，IDF）的方法构建基于兴趣地点的逆区域频率（Inverse Region Frequency，IRF），将用户位置推断问题转化为文档检索问题，根据查询似然模型计算用户与各个区域的相关性，选择相关性最高的区域作为用户的推断位置。实验表明，城市级的位置推断准确率达到 55.9%。

（2）密码分析

把 AI 与密码学相结合进行密码分析，能帮助攻击者获取与目标有关的密码信息，助力攻击者进一步获取敏感信息。葛钊成等[24]认为，神经网络的特性和密码学的需求相契合（如减少人工干预、鲁棒性、混淆原则、机密性等）。研究人员也针对神经网络在密码学中的运用展开了探索。2010 年，Alallayah 等[25]使用多层感知机对古典密码和流密码的破解，能实现接近 100%的准确率。2012 年，Alani[26]使用神经网络对 DES 和 3-DES 进行已知明文攻击。对 DES 和 3-DES，在拥有平均 2^{11} 及 2^{12} 个明密文对的条件下，能分别在 51 和 72 分钟完成密码分析。2017 年，Greydanus[27]提出使用 RNN 学习多表代换密码 Vigenère、Autokey 和 Enigma 的解密算法。作者证明了神经网络能对古典密码进行已知明文攻击。

（3）敏感信息恢复

敏感信息恢复能够帮助攻击者在准备阶段获取更多有利信息。腾讯的研究人员提出了几种应用场景，包括碎纸信息恢复[28]、马赛克信息恢复[29-30]和联邦学习的梯度信息恢复[31]等。碎纸信息恢复可以用 AI 从碎纸机的碎片中对原始文件进行拼接，大幅提高拼接的效率，助力攻击者获取敏感信息；马赛克信息恢复中，可使用 CNN 和 RNN 提取图像特征，利用 Seq2Seq 文本生成方案恢复被马赛克遮挡的原始文字信息等；联邦学习通过传递梯度信息而不是原始数据来保护用户隐私，然而研究人员却可以使用 iDLG 算法从梯度信息中还原原始图像，能 100%

准确地恢复样本标签，提高数据恢复效率。

（4）攻击路径选择

攻击图是基于模型的网络安全评估技术。攻击图从攻击者的角度出发，将综合分析内部网络中多种网络配置和脆弱性，找出所有可能的攻击路径，帮助管理人员直观地理解目标内部网络安全状态。攻击者可以对目标系统内部的主机、服务、漏洞等进行建模，构建攻击图，通过人工智能方法选择最佳的攻击路径。2018 年，Yousefi 等[32]提出基于强化学习的攻击图近似分析方法。作者先根据多主机多阶段的威胁分析建立给定网络拓扑的攻击图，然后在攻击图的基础上提炼出状态转换图，使用 Q-Learning 找出可能的攻击路。2020 年，Wu 等[33]在工控系统上使用 Q-Learning 方法进行攻击路径选择与评估，对工控系统的跨层威胁传播进行了建模，找到对攻击者有最大收益的攻击链路，具有较高的准确率。

（5）侧信道信息探测

侧信道攻击（Side Channel Attack）基于软硬件运行时产生的时间信息、功率消耗、电磁泄露或声音信息等额外信息源来完成信息探测和窃取。作为一种数据分析方法，AI 能处理和分析侧信道攻击中的数据。本文将 AI 与侧信道攻击结合的相关成果可以分为口令探测、密钥探测、网站指纹和其他信息探测等。

口令探测旨在通过侧信道信息探测用户在信息系统接入时使用的口令信息。2020 年，Chen 等[34]使用随机森林算法进行工业互联网设备的口令破解。作者利用用户对 IIoT 触摸屏智能设备进行击键时，设备的加速度计、陀螺仪和磁力计等传感器的反馈数据来识别受害者的口令。

密钥探测指通过侧信道信息探测密码算法的密钥。2011 年有研究人员使用 SVM 进行侧信道分析来窃取密钥[35]。2018 年，Yu 等[36]提出了基于深度学习针对 AES 密码电路的侧信道攻击，利用深度神经网络分析 AES 密码电路的功率耗散和电磁发射曲线来模拟功率噪声和电磁噪声之间的关系，进而对 AES 密码电路进行电源攻击，获取 AES 密钥。作者进一步研究了使用 CNN[37]、自动编码器[38]、硬件木马[39]等方法对密码电路进行侧信道攻击的场景。2019 年，Carré 等[40]利用缓存时序攻击（Cache timing attack）来窃取密钥。作者使用一个 RNN 模型，能够从缓存计时中自动检索一系列函数调用，用自然语言处理领域的技术处理部分标记数据。实验表明，该方法能够在 secp256k1 曲线上对 OpenSSL ECDSA 实施端到端自动攻击。

网站指纹旨在通过对加密流量、电磁信号等侧信道信息进行分析，获取用户访问的网站信息。从 2009 年起，Herrmann 等[41]开展针对加密网络进行网站指纹分析。近些年，Wang[42-43]、Panchenko[44]、Rimmer[45]、Jansen[46-47]等使用 SVM、*k*-NN、自编码器、CNN、LSTM 等方法在此领域取得了很多成果，逐渐实现网站指纹方法从实验环境向生产环境的转移应用。2021 年，La Cour 等[48]提出利用无线充电接口的电源侧信道攻击进行移动设备的网站指纹攻击。作者使用了 CNN 和 LSTM 对 iPhone 11 或 Google Pixel 4 上捕捉的电流轨迹数据进行分析，结果表明，无线充电电流数据能以超过 90%的准确率识别出两种设备上加载的网站。

在其他信息的探测上，2021 年，Gong 等[49]提出一种隐形红外阴影攻击（Invisible Infrared Shadow Attack，IRSA）场景，使用具备红外功能的摄像头在有遮挡物（如窗帘）的情况下捕捉目标的状态。作者引入 CNN、LSTM 等神经网络结构，提出使用 DeShaNet 通过多维特征融合进行阴影关键点检测，能够在严重的阴影变形下，成功恢复目标的 3D 骨架，进而提取目标信息，如活动和身份等。此外，还有研究人员提出用 AI 识别脚步声来判别目标的行动轨迹[50]，采集单个脚步声并转换为频谱图后，平均识别准确率可达 90%以上。不过，当前这个领域的攻击成本相对较高。

4.2.2 武器化

在武器化阶段，攻击者为突破目标系统边界前后的行动进行工具定制工作。AI 能更高效地完成此类工作。本文主要将突破网络边界类的案例归为此类，侧重于工具的定制、使用和维护，并据此把相关场景分为信息伪造、口令破解、验证码识别和防火墙绕过等类别。

（1）信息伪造

攻击者通过生成虚假的信息，能诱使目标进入预先设定的陷阱中，使目标完成攻击者设定的操作。借助人工智能在学习文字、图像、语音和视频等领域的能力，攻击者能够更快生成更具有欺骗性的信息。本节按信息载体的不同，分文字类、图像类、语音类、视频类及其综合应用进行介绍。

在文字方面的应用场景包括写作风格模仿、虚假信息生成和笔迹模仿等类别。2017 年，Baki 等[51]使用基于递归转换网络（Recursive transition network）的数据引擎（Data Engine）学习美国政治人物 HC 和 SP 泄露的电子邮件特征，并根据学习结果

生成新的邮件，让普通人辨别真伪。结果表明，模仿的 HC 的邮件中，有 71.69%的被认为是真实邮件。2021 年，Ranade 等[52]提出一种污染开源情报 OSINT 的方法。作者提出使用 GPT-2 进行迁移学习，用安全文章、APT 报告和漏洞库信息对 GPT-2 模型进行微调，使模型能生成虚假的威胁情报信息。通过将假信息在安全论坛、社交网络等平台进行引流，能吸引第三方威胁情报搜集引擎进行情报采集，进而实现用虚假的威胁情报信息对第三方威胁情报数据库进行投毒。人工智能还可以对手写文字进行伪造。2021 年，腾讯安全平台部使用生成对抗网络来模拟人类笔迹[53]，可以做到让签名以假乱真的地步。作者引入生成器和鉴别器，使用非成对样本进行训练，利用两个网络进行内容提取和风格提取，完成字迹模拟。

在语音模仿方面，AI 能完成语音合成、语音转换、对抗攻击等任务，对说话人验证系统（Automatic Speakers Verification，ASV）或相关个人进行攻击。语音合成能将指定的文字信息转换为目标人物的说话声音，完成从文字到语音的映射。语音转换能将 A 用户的语音转换为 B 用户的语音，完成语音到语音的映射。Rebryk 等[54]提出了用于实时语音风格转换的神经网络 ConVoice，能借助很少量的目标用户的语音数据实时地将其他用户的语音转换为目标用户的语音风格，可以在 1 秒内合成 12 秒的语音数据。对抗攻击主要针对智能化语音识别系统进行攻击，通过在普通音频中添加扰动的方式，让语音识别系统错误识别语音内容。2018 年，Yuan 等[55]利用对抗攻击提出了 CommanderSong，能把命令嵌入到音乐中，可以在人类无感的情况下，让语音识别系统识别并执行嵌入的命令。Wenger 等[56]对真实场景种的语音攻击进行了调查，发现基于 DNN 的语音合成工具能以 50%到 100%的成功率欺骗智能语音识别系统，合成语音能够模仿真实语音识别系统（如微软 Azure、WeChat 和 Alexa）中 60%的用户声音，而现有防护手段却无法对此类攻击进行有效防护。语音领域的攻击已造成经济损失。2019 年，攻击者使用 AI 模仿英国一家能源公司的母公司 CEO 的声音与该能源公司的高管通话[57]，让其将资金汇给一家国外供应商，骗取了该公司 22 万欧元。随后，2020 年初，阿联酋的一位银行经理也遭遇了使用语音伪造进行的诈骗[58]，被骗取 40 万美元。

在图像领域的场景包括图像生成、图像风格改造、对抗攻击等任务，可以用来进行欺骗。例如，通过 AI 手段进行换脸，能绕过一些静态人脸验证系统；通过对抗攻击，能让人脸验证系统失效，让攻击者进入目标系统。2019 年，腾讯玄武实验室[59]发现，苹果手机的 FaceID 在用户带上眼镜时不能区分真实的人眼，进而通过在普通眼镜的镜片上贴上黑色胶带和白点，来绕过 FaceID 人脸识别系统，能在受害者不在场的情况下，登录进手机系统并执行转账等操作。2019 年，Thys 等[60]提出使用一个小面积的对抗扰动图像让深度学习模型检测不到监控摄像头中出现的人。作者在 YOLOv2 模型上的实验表明，把扰动图像覆盖到人物身上后，模型对人的检测召回率由 100%下降到了 26.46%。在真实场景中，把扰动图像打印出来放在人的胸前，也能让检测模型失效。2019 年的 GeekPwn 大赛上[61]，清华的 TSAIL 战队使用仅 14x14 大小的纸片，就成功攻破了 YOLOv3 模型，让人在镜头下“隐身”。

在视频领域包括结合 AI 在文字、语音和图像领域的能力，创造出以假乱真的作品。深度伪造[12]（DeepFake）是一类最经典的案例。2017 年 12 月，一名 Reddit 论坛的用户以 DeepFake 为用户名发布了使用深度学习技术将一位女明星与成人视频主角换脸的视频，并由此引发了轰动。近些年，生成对抗网络和自编码器等技术趋于成熟，攻击者利用深度伪造技术生成重要人物的虚假视频，实现人脸和语音的高效生成和替换，完成钓鱼和诈骗等社会工程学攻击。2019 年一款换脸软件 ZAO[62]的上市引发了关于隐私保护和 AI 伦理的大量争议。深度伪造技术还被用在了涉及政治的话题上，被人用来制作和政治人物相关的虚假视频。2022 年 3 月，在俄乌两国发生军事冲突时，一段伪造的乌克兰元首呼吁士兵投降的视频出现在网上[63]，引发轩然大波。

综合以上领域的滥用行为，AI 会带来更大的威胁。除了 4.2.1 节中提到的 SNAP_R 外，2018 年，研究人员在 Black Hat 上提出使用 AI 技术进行网站钓鱼的演示系统 DeepPhish[64]。作者收集了大量的正常网站和钓鱼网站使用的域名、URL、证书及网站内容，使用神经网络学习恶意内容的样式，生成包括图文在内的恶意内容。实验表明，DeepPhish 的钓鱼成功率比传统方法提升了 20%-32%。除了执行攻击性的任务，还有研究人员提出使用 AI 技术进行社交网络上的用户身份伪造。Salminen 等[65]提出了使用神经网络生成用户信息的方法。作者收集了用户信息，包括个人资料、发布的推文和评论、转发等，使用 CNN、BiLSTM 生成对应的用户信息。2020 年，Basu[66]在 Black Hat 上提出了通过社会工程方法进行社交网络用户克隆的技术。作者通过社交软件上的

文字对话、语音、视频等信息作为训练集，通过神经网络模型成功克隆了另一个“自己”。作者指出，该方法能借助 VoIP 等技术对目标进行欺骗。

（2）口令破解

攻击者通过各种手段破解用户口令，能帮助攻击者更高效地接入目标系统。攻击者通常会使用常见的弱口令、互联网泄露的用户口令和基于用户身份信息生成的口令来实施爆破行为，而 AI 的武器化则会大大提升口令破解效率。

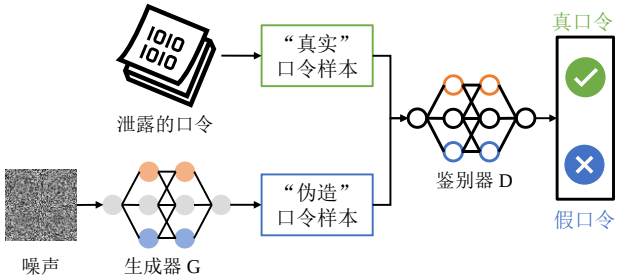


图4 PassGAN 结构示意[67]

Figure 4 Architecture of PassGAN[67]

2016 年，Melicher 等[68]首先提出了使用循环神经网络进行口令爆破的方法。实验表明，与传统方法相比，基于 AI 的方法在口令爆破方面有更高的成功率。2019 年，Hitaj 等[67]提出使用 GAN 生成用户口令的方法 PassGAN，如图 4。作者认为，神经网络的复杂度足以描述大量的口令属性和结构。在测试集上的对比结果显示，PassGAN 生成的口令能匹配到更多的真实口令。2020 年，Xia 等[69]提出将基于统计的概率上下文无关文法（Probabilistic Context-Free Grammar，PCFG）和 LSTM 结合起来进行口令猜测的模型 GENPass。GENPass 将猜测由字符级提升到词级，能学习多种来源的泄露口令库来生成“通用”词库，在达到 50%匹配率的情况下，能减少大量的猜测次数。2021 年，Pasquini 等[70]提出使用表示学习进行口令猜解。作者使用 GAN 的生成器和 Wasserstein 自编码器（Wasserstein Auto-Encoder，WAE）对口令表示进行建模，使语义相近的口令在高维空间的距离更近。作者提出了两个口令猜测框架 CPG 和 DPG，能以一定条件生成任意偏置的口令，并能根据新的口令知识模拟目标口令分布，实现较高的口令匹配度。Xu 等[71]提出块（chunk）的概念，将口令按照常用搭配分成若干个块（如 p@ssw0rd4ever 可划分为 p@ssw0rd 和 4ever 两个常见口令块），构建基于块的马尔可夫模型、PCFG 模型和神经网络模型来完成口令猜解。与已有方法比较，这三种模型的猜解成功率分别提高了 5.7%、51.2%和 41.9%。除了口令外，人工智能还可以攻击指纹识别系统，助力攻击者接入目标系统。Bontrager 等[72]提出使用神经网络进行指纹合成的方法 DeepMasterPrints，能够创建指纹图像来欺骗指纹识别系统，可用于对指纹验证系统发起字典攻击。

（3）验证码识别

一些 Web 系统会使用验证码来对访问者进行人机验证，防止自动化脚本对系统进行无限访问或爆破。攻击者能使用 AI 技术高效准确地识别各类型验证码，使自动化爆破目标系统成为可能。Guerar 等[73]对已有的验证码类型进行了总结，通常包括基于文字、语音、图像、问答的验证码和交互式验证码。基于文字的验证码通常由大小写字母和数字的形态变换组成，是最常见的验证码形式。在中文社区也有使用汉字的验证码。为了增加破解难度，此类验证码会添加噪声干扰。然而，AI 仍能完成验证码的破解。2017 年，Kopp 等[74]提出使用 CNN 识别基于文字的验证码。作者用两个神经网络来完成识别任务。首先使用一个带有神经网络的滑动窗口创建热力图来判断窗口中心是否有字符存在，然后使用 *k*-means 算法判断热力图中字符最可能存在的位置，最后再使用另外一个神经网络来识别字符。作者采集了 11 种不同样式的验证码，测试结果表明，当两个神经网络均使用 CNN 时，文中的方法能达到 80%以上的识别准确率。不同于只包含 62 种字符的英文验证码（26 个字母的大小写、10 个数字），中文验证码通常包含大约 2000 个常见汉字，自动化识别难度较大。在中文验证码方面，2021 年，Wang 等[75]提出使用 Faster R-CNN、Inception V3 和 LSTM 的识别方案，使用 Faster R-CNN 确定图片上文字的位置，用 Inception V3 模型提取包含文字的子图特征并生成特征图，LSTM 将特征映射解码为单个文本。作者爬取了来自百度、QQ、网易、人民网等 11 个知名中文社区的 2000 张验证码图像，在此基础上生成 12 万张新图像用于模型的训练。实验表明，文中的方法对中文验证码识别的准确率能达到 86.9%。除此之外，研究人员还提出了基于 GRU、GAN 等方法的验证码破解方法，能对中英文验证码及其变换形式进行识别，能达到 90%以上的破解准确率。基于语音的验证码破解上，Tam 等[76]于 2008 年率先提出使用统计学习的方法进行验证码的识别。作者使用了 AdaBoost、SVM 和 *k*-NN 等方法破解包括 reCAPTCHA 在内的语音验证码，整体准确率达到 71%。基于图像的验证码以谷歌 reCAPTCHA 为代表。此类验证码由一个关键词和一组或多组图像组成，需要用户识别给出的图像与关键字是否匹配。

此类验证码的破解难度大，但仍被研究人员找到了破解方法。2016 年，Sivakorn 等[77]提出用图像切割和图像检索的方式破解谷歌的 reCAPTCHA。作者会首先切割验证码图片，并在图像检索网站（如谷歌图片）检索被切割后的图片，从返回结果里的抽取关键文字，并和验证码页面的标签进行对比，根据对比结果来判断是否“选中”当前图片。2020 年，Alqahtani 等[78]基于统计学习方法进行 reCAPTCHA 的自动化识别。作者使用了 Word2Vec、朴素贝叶斯、CART、Bagging 和 RF 等方法分别进行了测试，RF 方法对单个图像的识别准确率达 85.32%，对 reCAPTCHA 破解成功率达 56.29%。本领域的代表性成果见表 2[74-87]。

表2　验证码识别的相关工作

Table 2 Related Work on CAPTCHA Recognition

年份	工作	类型	变换	训练集	测试集	数据来源	方法	模型	准确率
2015	[79]	文本	形变、旋转	1 万	10 万	Cool PHP CAPTCHA	CNN	自定义：3 个卷积层，3 个池化层，2 个全连接层	约 80%
2017	[80]	文本	背景、噪声、形变、旋转	1.5 万		爬取的 3 个网站的 CAPTCHA	Fast R-CNN	VGG_CNN_M_1024, ZF[81], VGG16	96.50%
2017	[74]	文本	背景、噪声、形变、旋转、多余字符、重合	10 万	1000	使用 BotDetect CAPTCHA 服务生成的不同变形的验证码	CNN+聚类	LeNet-5	65.50%
2018	[82]	文本	中文，背景、噪声、形变、旋转	5 万	1.5 万	在 kaptcha 上基于 200 个汉字生成	CNN	基于 LeNet-5 的更深的结构	84%
2018	[83]	文本	中英文，背景、噪声、旋转、变形、重合、多层结构	2000	1400	从目标网站上爬取	CNN	基于 LeNet-5 的更深的结构	最高 90%
2018	[84]	文本	中英文，背景、噪声、旋转、变长	6 万	1 万	自生成	LR+CNN	基于 LENET-5，增加 BN 和 dropout	94.60%
2019	[85]	文本	背景、噪声、形变、旋转、重合、变长、多层	20 万	3 万	Google CAPTCHA	CNN+RNN	Inception-v3、LSTM	最高 98.3%
2020	[86]	文本	背景、噪声、形变、旋转、多层	2500	80 万	训练：爬取 Alexa top 50，测试：EMNIST 生成	无监督	ResNet-101、GRU	最高 91.5%
2021	[75]	文本	中文，背景、噪声、旋转、变形、多字体	10 万	2 万	爬取 2 千验证码，生成 12 万验证码	CNN + LSTM	Faster R-CNN，Inception V3，LSTM	最高 86.9%
2020	[78]	图像	reCaptcha	700 组 3x3 数据		reCAPTCHA dataset challenges	统计学习	NB, CART, bagging with CART, and RF	56.29%
2021	[87]	图像	reCaptcha	9582		deathlyface reCaptcha	CNN	自定义：5 个卷积层，5 个最大池化，dropout 等	92.98%
2008	[76]	语音		900	100	google.com、digg.com 和 recaptcha.net	统计学习	AdaBoost, SVM, and *k*-NN	71%

（4）防火墙绕过

Web 应用防火墙（Web application firewall，WAF）通常部署在系统边界，能拦截恶意带有攻击载荷的非法访问数据，保护应用系统的安全。近些年，有研究人员使用 AI 技术自动化地探测 WAF 规则，生成可以绕过防火墙的攻击载荷。

2017 年，Takaesu[88]提出使用遗传算法自动生成注入代码的方法 DeepGenerator。作者以生成能绕过 WAF 且过去未使用过的注入代码为目的，以 HTML 和 JavaScript 的组成元素为基因，生成以 XSS 注入代码为个体的多个基因组合。作者以 HTML 语法的正确性、脚本的可执行性和 WAF 的绕过能力对每代的个体进行评估，选择每代得分高的个体进行基因交叉，并继续生成新个体。实验结果表明，使用遗传算法能生成在 WAVSEP 中未出现过的新注入代码。在可执行性和绕过 WAF 方面，第 349 代开始，生成的个体可以被执行；第 6307 代起，出现了可以绕过 WAF 的个体。

2020 年，腾讯朱雀实验室展示了一种机器学习驱动的 WAF 规则探测器 Deep X-Ray[89]。该探测器用带有注意力机制的 LSTM 自动化地探测 WAF 规则，能够复制 WAF 的防护能力，构造出绕过 WAF 的载荷。实验结果表明，该方法对基于规则的主流 WAF 产品的拟合率达 97%以上，能够实现无专家干预下的一键规则窃取。Demetrio 等[90]提出基于对抗机器学习的 WAF 逃逸工具 WAF-A-MoLE，能够生成攻击载荷。其思路是，在能被 WAF 拦截的载荷的基础上，迭代增加突变来生成新的载荷，同时测试新

载荷在 WAF 上进行分类的置信度，通过逐渐降低置信度到阈值以下来完成 WAF 的绕过。作者以 SQL 注入（SQLi）为例进行了实验，在迭代一定轮数后，大多数分类器对恶意增量载荷的置信度降为 0。

2022 年，Qu 等在 Black Hat 上提出了能绕过 AWS WAF 的方法 AutoSpear[91]。以 SQL 注入为例，AutoSpear 首先将 SQLi 载荷表示为层次树（hierarchical tree）的形式，并进行语义分析。载荷的叶子节点是 SQLi 语句中不可再分的元素（如连接词、数字、符号、空格等），父节点是由其子节点组成的 SQLi 语句。随后，AutoSpear 利用基于上下文无关文法（context-free grammar）的加权变异算法生成大量语义等效负载，即候选节点/子树。最后，AutoSpear 采用增强的蒙特卡洛树搜索（Monte Carlo Tree Search）来指导每个节点的变异替换，输出最终的有效载荷。实验表明，AutoSpear 生成的载荷能以高达 99.73%的逃逸率绕过 AWS WAF，在现实场景中实现了防火墙绕过。

4.3 入侵阶段

4.3.1 投递

在投递环节，攻击者需要将恶意载荷传递到目标系统内部。攻击者常用的手法有利用电子邮件附件、水坑攻击、USB 接入及信息隐藏等。AI 能助力构建具有欺骗性的电子邮件和网站，这属于武器化的过程。涉及到载荷的传递时，通常用到的是免杀及信息隐藏技术。本节主要关注信息隐藏技术。

信息隐藏技术可以将信息隐蔽地传递至目标系统。传统的信息隐藏技术能将信息嵌入到图片、音频或视频载体中，在不影响载体的视觉或听觉效果的前提下，将嵌入的信息传递到接收方。AI 方法也可以用于信息隐藏领域。研究人员提出了很多基于深度学习的隐写方法，让深度学习助力载体图像的获取、隐写失真的设计和含密图像的生成[92]。例如，使用 GAN 可以根据隐写的需要生成适合隐写的载体图像[93]，使含密图像具有欺骗隐写分析器的能力。同时，深度学习还能助力无载体的图像隐写[94]，完成载体图像选择和合成等任务。2018 年，Zhu 等[95]提出了一套使用深度网络进行信息隐藏的方法 HiDDeN，可以在图像经历了 Dropout、Cropout、切割、高斯模糊和 JPEG 转换后，仍能以 85%的准确率恢复原始信息。除了传统隐写领域的应用外，研究人员提出使用神经网络基于大数据进行信息隐藏。2019 年，Zhu[96]提出用神经网络对选定图像的识别结果作为混淆种子和明文消息进行计算得到差值，将差值、图像和模型传递给信息的接收方，接收方能逆向恢复出原始消息。对于分析者来说，在不知道恢复协议的情况下，难以通过已有信息得到消息。不过，受限于信道容量和载体大小，图像类信息隐藏的方式只能传递少量信息。同时，这类方法需要攻击者在目标内部已经有一个信息提取器，以便提取嵌入的信息。

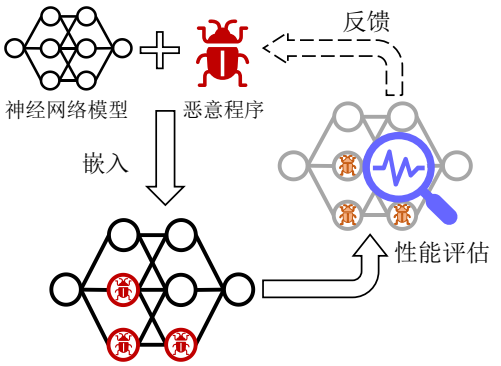


图5 把恶意程序嵌入到 DNN 模型中

Figure 5 Embedding Malware into DNN Model

鉴于常见的神经网络模型具备较大的体积，有人提出借助神经网络模型的冗余性来传递大体积的数据，如图 5。2020 年，Liu 等[97]提出了将恶意代码当作神经网络模型参数嵌入到模型进行传递的方法 StegoNet。基于神经网络模型的冗余性，当部分参数被替换时，在接受一定程度的神经网络性能损失的情况下，能将恶意代码嵌入到模型中。不过，StegoNet 中的方法嵌入率较低，对模型性能的影响较大。2021 年，Wang 等[98-99]提出了能解决上述问题的恶意代码嵌入方法 EvilModel。在常见的神经网络框架（如 PyTorch、TensorFlow、Keras 等）中，模型的参数均为 32 位浮点数。当按照 IEEE 754 标准表示成 4 字节的形式时，参数的后几个字节属于浮点数的尾数部分，对参数值和模型运算的影响较低。EvilModel 中的半替代法保持参数的前两个字节不变，将后两个字节替换为恶意代码，对参数值的影响在小数点后的第 5 位。从实验效果看，该方法能在模型中嵌入大小为模型体积一半的恶意代码，具备较高的嵌入率，同时不影响模型性能，能助力攻击者将恶意代码投递到目标系统内。

4.3.2 利用

利用环节主要是寻找和利用系统或程序漏洞，使用包括 0 day 在内的各种方式触发目标程序漏洞，获取目标系统权限。当前，AI 能助力漏洞发现与利用的过程。本文将相关工作分为针对软件漏洞和 Web 应用漏洞的发现与利用。

（1）软件漏洞

漏洞发现技术可以帮助防御者找到应用程序中

的漏洞，也可以助力攻击者发现目标系统的安全缺陷。目前，有研究人员提出使用 AI 进行漏洞发现。2018 年，Li 等[100]提出基于深度学习的漏洞检测系统 VulDeePecker，首次将深度学习引入漏洞检测领域。作者先将程序解析成代码片段，进行标注后转换为特征向量，随后使用基于注意力机制的 BiLSTM 对代码片段进行漏洞检测。实验表明，此方法可以识别多种漏洞，准确率可达 90%以上。随后，研究人员提出更多的基于深度学习的漏洞检测方法。She 等[101]提出使用神经网络的 Fuzz 方法 Neuzz。作者使用一个平滑的替代函数来模拟目标对不同输入分支的行为，将神经网络和梯度下降引入 Fuzz 中，提高模糊测试的效率。针对定向灰盒 Fuzz 中存在的测试输入对可疑代码不可达的问题，Zong 等[102]使用基于深度学习的方法，在执行目标程序之前预测输入对可疑代码的可达性（即是否错过目标），过滤掉不可达的输入，以此提高 Fuzz 的效率。Zhou 等[103]提出基于图神经网络进行漏洞识别的方法 Devign。作者从 4 个大型开源项目（Linux Kernel、QEMU、Wireshark 和 FFmpeg）的提交记录中提取相应的函数，使用 Joern 将提取的数据转换为抽象语法树和控制流图，随后用图神经网络对邻接节点的信息进行聚合来学习节点特征，最后通过卷积模块提取有意义的节点特征用于图级别的分类预测。尽管这些方法能帮助攻击者发现软件漏洞，但要用于目标系统内部的漏洞发现，还存在一些技术门槛。

（2）Web 应用漏洞

近些年，有研究人员提出使用 AI 构建自动化的渗透测试工具。从已有的案例看，此类工具已经能涵盖攻击的信息搜集、漏洞发现、载荷生成和渗透测试等多个阶段，能组成较为完整的攻击链。2016 年，Takaesu[104]在 Black Hat 上提出基于机器学习的自动化漏洞发现工具 SAIVS。该工具以发现 Web 漏洞为目的，会爬取并分析 Web 页面信息。以用户的注册或登录页面为例，该工具使用朴素贝叶斯方法识别页面类型，判断页面包含哪些字段，随后根据字段内容构造访问请求。当收到请求后，该工具还会使用朴素贝叶斯方法判断页面是否有报错，根据报错信息判断该 Web 系统是否有漏洞，以及所提交的字段需要进行哪些调整，并进行下一步的测试。在构造请求和根据页面结果进行调整时，该工具使用多层感知机和强化学习（Q-Learning）确定参数数据和字段内容。2017 年，Petro 等[105]在 DefCon 上提出基于强化学习的自动化渗透测试工具 DeepHack。作者认为，通过神经网络拟合状态函数，让神经网络给出下一步行为的激励，可以促使程序自动化地选择最优解，以指导攻击载荷的生成，完成自动化攻击。2018 年，Masuya 等[106]提出一个智能化 Web 信息收集和渗透测试工具 GyoiThon，能自动地识别目标的产品类型，如 CMS 版本、Web 服务器软件、架构、编程语言等，使用 Metasploit 对识别出的产品执行攻击。在识别服务器信息阶段，GyoiThon 使用特征字符串匹配和机器学习两种方法进行产品版本及漏洞识别。2018 年，Takaesu[6]在 Black Hat 上提出基于强化学习的自动化渗透测试工具 DeepExploit。DeepExploit 也借助 Metasploit 执行攻击任务。不同的是，DeepExploit 使用 RPC API 供机器学习模块和 Metasploit 进行交互，让机器学习模块下发指令，Metasploit 提供执行的反馈，同时记录指令和反馈结果，用来训练攻击服务器。DeepExploit 还可以自动化地生成测试报告。

4.3.3 植入

攻击者在植入环节将恶意代码植入到目标系统中。生存是恶意代码的首要任务之一。目标系统往往部署有反病毒引擎，恶意代码必须能隐蔽免杀地驻留在目标系统。检测逃逸指恶意程序能通过目标系统安全设备的多维度检查，成功规避检测器的检测，进入并驻留系统内部。目前，一些厂商开始使用机器学习方法完成威胁检测和发现，如 McAfee[107]、Fortinet[108]、奇安信[109]、深信服[110]等，如表 3。然而，机器学习方法往往面临可解释性差和对抗攻击的问题，给恶意代码绕过检测提供了机会。

表3　工业界人工智能安全产品案例

Table 3 Examples of Security Products Using AI

厂商	产品	检测对象	目的
迈克菲 McAfee	[107]	恶意软件	威胁追踪和分析
Fortinet	FortiAI[108]	日志、恶意软件	安全事件发现与分析
诺顿 Norton	[111]	恶意软件	防病毒
卡巴斯基 Kaspersky	KFP[112]	环境状态、行为分析、恶意软件	反欺诈
瑞星	[113]	网络流量、恶意软件	网络威胁检测
奇安信	网神[109]	网络流量	DNS 安全防御
深信服	EDR[110]	恶意软件	终端安全防护
华为	HiSec Insight[114]	网络流量	安全态势感知
H3C	[115]	网络流量	防火墙
观成	瞰云[116]	网络流量	网络威胁检测

近些年，研究人员陆续利用对抗机器学习方法构建了一些可以绕过检测的恶意代码。如图 6 所示，恶意代码寄生载体包括 PDF 文件、Windows 设备、Android 设备、IoT 设备和硬件电路等，使用的主要方法有 FGSM、C&W、GAN 和强化学习等，攻击方

式可根据攻击者对检测器的了解程度分为黑盒、灰盒和白盒攻击。此类攻击的难度在于，既要使生成的恶意代码能逃避检测，又要使修改后的程序能正常运行，同时原有的恶意功能不受影响。

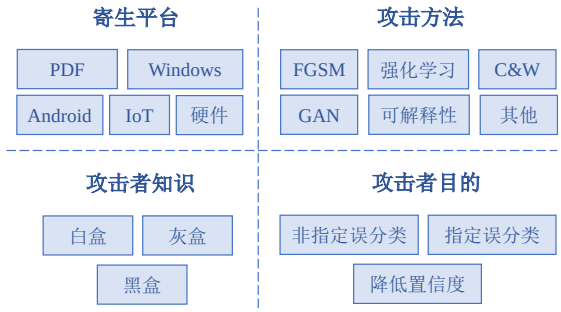


图6 恶意代码检测逃逸类别划分

Figure 6 Categories of Malware Detection Evasion

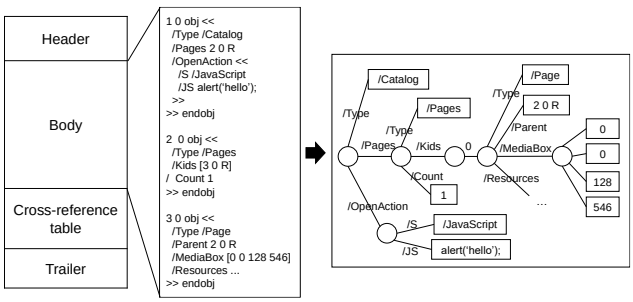


图7 PDF 文件的物理和逻辑结构[117]

Figure 7 Physical and Logical Structure of a PDF File[117]

（1）PDF 恶意代码

PDF 文件可以作为邮件附件、网络资源等投递到目标网络内部。如图 7，由于 PDF 结构的复杂性，恶意代码可以通过隐藏至 PDF 文件里。在恶意 PDF 的检测逃逸上，2016 年，Xu 等[117]提出使用基因编程的方法修改 PDF 文件的统计特征。作者攻击的目标是 PDFrate 和 Hidost，二者分别以随机森林（RF）和支持向量机（SVM）作为恶意 PDF 检测算法。实验表明，构造的恶意 PDF 在两个分类器上的逃逸率均能达到 100%。2017 年，Dang 等[118]在上述研究的基础上提出了一种逃逸方法 EvadeHC，能在黑盒的条件下达到 100%的逃逸率。2019 年，Dey 等[119]使用基因编程方法对[117]进行改进，能以 100%的逃逸率绕过另一个检测器 AnalyzePDF。2020 年，Li 等[120]提出使用基于特征向量的 GAN 方法进行检测逃逸，从每个 PDF 文件中提取 135 维的向量输入至 GAN 模块中，形成对抗特征后重组为 PDF 文件，能实现 100%的逃逸率。2021 年，Bae 等[121]提出了基于 GAN 的恶意 PDF 逃逸方法，能以最小的改动绕过主流检测器。值得一提的是，作者构建的恶意 PDF 文件还能绕过 VirusTotal 上的反病毒引擎的检测。

（2）Windows 恶意代码

Windows 上的恶意程序最为主流，因此针对 Windows 上的检测和对抗工作也最多。2017 年，Hu 等[122]提出了基于 GAN 的 Windows 恶意代码逃逸方法 MalGAN，如图 8。针对使用样本 API 序列的黑盒检测器，作者使用生成模型添加扰动构造对抗样本，引入代替检测器判断样本是否为恶意。代替检测器由良性样本和对抗样本训练而来，用来模拟黑盒检测器。通过对抗训练，能够最小化对抗样本被检出的概率。实验表明，生成的恶意样本能使恶意代码检测率由 90%以上降到 0.2%以下甚至 0。2018 年，Anderson 等[123]提出利用强化学习的激励和反馈机制修改静态 PE 恶意代码的结构特征来规避检测的方法，使用黑盒检测器作为样本修改结果的反馈源，通过增加无用函数调用、修改段名、移除签名信息等方式得到能逃逸检测的样本。2018 年，Hu 等[124]提出绕过 RNN 检测器的方法。RNN 检测器通常使用样本的 API 序列进行检测。作者通过一个生成式 RNN 模型，在一段 API 序列后附加新的 API 序列来使目标检测器失效。2019 年，Luca 等[125]提出使用可解释性的方法分析恶意样本中对检测器判别结果影响最大的部分，并对其进行修改，来绕过 MalConv 检测器。2020 年，Yuan 等[126]提出了端对端字节级黑盒对抗攻击方法，用生成器生成一段 Payload 添加在样本后，通过黑盒检测器的反馈结果训练鉴别器。在使用时，只需要生成器生成 Payload 即可对抗检测器。此方法能在添加的 Payload 长度为 2.5%时达到 100%的逃逸成功率。2021 年，Amich 等[127]提出使用模型可解释性方法分析扰动区域来得到扰动效果，根据扰动效果构建新的对抗样本，构造的新样本能逃避大多数恶意代码检测器。与 Windows 上的恶意代码检测器逃逸相关的主要工作见表 4[122-131]。

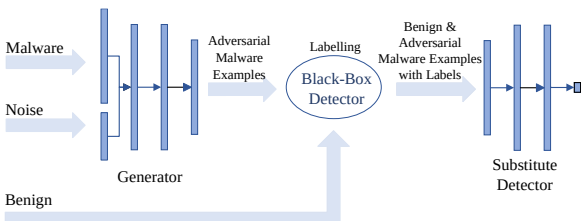


图8 MalGAN 结构[122]

Figure 8 The Architecture of MalGAN[122]

（3）Android 恶意代码

移动设备的普及使 Android 恶意代码开始流行，相关的检测和对抗工作也逐步升级。2016 年，Grosse 等[132]率先提出利用梯度构建对抗样本来规避 Android 恶意软件分类器。作者使用了 Android 应用的 8 个类别（硬件组件、权限、组件、Intent、受限 API 调用、用户权限、可疑 API 调用、网络地址）的

545333 个特征构成二值向量，使用前馈神经网络进行检测。在对抗样本的构造上，作者寻找使梯度变化的方向，构造最大化判别结果的扰动。实验表明，该方法能使 63%的恶意样本逃逸检测。2020 年，Li 等[134]使用 26 种攻击方法在两种数据集上对 10 种检测器的对抗效果进行实验，系统研究了 Android 恶意软件对抗领域的方法及其效果。2021 年，Li 等[135]提出针对 Android 恶意软件检测器的后门攻击，通过污染数据集的方式，将触发器体积较小的后门植入到检测模型中，使恶意软件的逃逸率达到 99%。这种攻击虽然不是对抗机器学习方法，但也是针对神经网络模型的一种常见的攻击。由此可以看到，神经网络仍是十分脆弱的，攻击者可以利用多种方式完成任务。Android 方面的主要工作见表 5[133-138]。

（4）IoT 恶意代码

IoT 设备也逐渐成为攻击者的目标。2019 年，Abusnaina 等[139]提出使用对抗学习的方法攻击以控制流图进行 IoT 恶意软件检测的系统。作者首先对已有的 8 种攻击方法进行验证，包括 C&W、DeepFool、FGSM 等，并提出了图嵌入和增强（Graph Embedding and Augmentation，GEA）对抗方法，通过修改代码的方式改变控制流图，进而使恶意软件规避检测。实验表明，已有的攻击方法和 GEA 方法都能在一定条件下达到 90%以上的逃逸率。

表4　Windows 恶意代码检测逃逸方向的代表性工作

Table 4 Representative Work on Windows Malware Detection Evasion

年份	工作	场景	主要方法	对抗目标	数据	描述	效果
2017	[122]	黑盒	GAN	RF、LR、DT、SVM、MLP 和投票 VOTE	爬取自 malwr[1]网站，18 万样本，30%为恶意	用 GAN 给恶意样本添加扰动，最小化被检出的概率。	生成的恶意样本能使恶意代码检测率由 90%以上降到 0.2%以下甚至 0。
2017	[129]	黑盒	RL	GBDT	10 万样本	修改导入表、修改段名、创建新段、段尾增加字节、添加跳转、调整调试信息等操作。	有 16%的逃逸率。
2018	[123]	黑盒	RL	GBDT	10 万样本	使用和上文一样的操作来修改 PE 头、段表、导入导出表、可读字符串数量、字节统计、二维字节熵统计信息。	能大幅增加逃避概率，生成的样本不仅能逃避 GBDT 的检测，也能逃避 VirusTotal 上一些检测器的检测。
2018	[124]	黑盒	RNN	RNN(LSTM)	爬取自 malwr 网站，18 万样本，70%为恶意	训练一个代替 RNN 和一个生成 RNN，通过添加多余的 API 调用来使检测器失效。	生成序列在多个 LSTM 模型上的检出率由 90%以上降至 1%-3%。
2019	[128]	白盒、黑盒	FGSM	MalConv[2]	恶意 5200 个（DAS[3]、MB[4]和 VirusShare[5]），良性 5150 个（Windows 系统和其他软件）	附加良性特征或扰动并用 FGSM 等方法优化。	添加 20000 字节扰动时，逃逸率达 73%。
2019	[125]	白盒	可解释性	MalConv	60 个恶意样本来自 the Zoo[6]和 DAS	分析每个字节对判别结果的贡献，找出影响最大的结构，进行修改。	修改 DOS 头可使 52 个样本绕过 MalConv 检测。
2020	[126]	黑盒	GAN	MalConv	恶意样本来自 VirusTotal[7]和 Kaggle2015[8]，良性来自 Chocolatey[9]软件	用生成器在样本后附加一段载荷，使添加载荷后的样本能绕过检测。	附加 1%载荷时，逃逸率可达 64.66%；2.5%时，小样本逃逸率可达 100%；20%时，对大样本可达 80%以上。
2021	[127]	灰盒	LIME	MLP、LR、RF、DT、ET、LGBM、DNN	恶意 2 万（VirusShare）、良性 2 万（CNET[10]）、EMBER[11]数据 1 百万样本、MNIST	通过分析扰动区域来得到扰动效果，根据扰动效果构建的新对抗样本。	能逃避大多数恶意代码检测器。
2021	[130]	黑盒	RL	未说明	来自 VirusTotal 的样本，后门、木马、蠕虫各 100 个	将优化激励函数问题转化为求最大熵问题，实现激励函数的自动生成。	对比了 DQEAF 和基于 DQN 的方法，本方法的逃逸成功率最高，为 76%。
2021	[131]	黑盒	RL	EMBER、MalConv	来自 VirusTotal，19650 个样本	添加或修改导入表、段、校验和、时间戳、附加字节、删除调试段、数据签名等。	和其他 RL 方法对比，本方法的逃逸率有提升。

[1] malwr. https://malwr.com/
[2] MalConv. https://aaai.org/ocs/index.php/WS/AAAIW18/paper/view/16422
[3] DAS. http://dasmalwerk.eu/
[4] MB (MalwareBenchmark). https://doi.org/10.1155/2018/4947695
[5] VirusShare. https://virusshare.com/
[6] the Zoo. https://github.com/ytisf/theZoo
[7] VirusTotal. http://www.virustotal.com/
[8] Kaggle2015. arXiv:1802.10135
[9] Chocolatey. https://chocolatey.org/
[10] CNET. https://download.cnet.com/s/software/windows/?licenseType=Free
[11] EMBER. arXiv:1804.04637

表5　Android 恶意代码检测逃逸方向的代表性工作

Table 5 Representative Work on Android Malware Detection Evasion

年份	工作	场景	主要方法	对抗目标	数据	描述	效果
2017	[136]	白盒	FGSM	MLP	DREBIN[12]（129013 个安卓应用，123453 个良性，5560 个恶意）	寻找使梯度变化的方向，构造最大化判别结果变化的扰动。	使用增强的对抗算法，使 63%的恶意样本被误分类。
2019	[137]	白盒、灰盒、黑盒	C&W 、JSMA	RF 、SVM 、*k*-NN、DNN	良性：5879 个(PlayDrone[13])，恶意：5560 个(DERBIN)	在安卓 APK 的资源文件、语义特征等上进行自动化地添加扰动。	不同攻击方法和模式下的逃逸率不同，最低约 20%，最高达 100%。
2020	[133]	白盒、灰盒	n-strongest node ；GSM	Adagio[14]	良性：49947 个 APK (AndroZoo[15])，恶意：5560 个(DREBIN)	把 *n* 个对分类起促进作用的节点插入领域散列，执行特征嵌入得到 *P* 维向量，降低检出概率。或根据梯度调整向量值逃避检测。	使用白盒 *n*-strongest node 方法能在改动 22.7%节点的条件下，使 72.2%的样本错误分类。
2020	[134]	白盒、灰盒、黑盒	26 种攻击方法	6 种 DNN 检测器	DERBIN 和 AndroZoo	使用多种方法，使 DREBIN 特征使向量值发生变化，进而影响检测器。	系统研究了各种攻击方法对不同检测器的影响。
2021	[135]	黑盒	数据投毒	DREBIN 、MaMaDroid[16]、DroidCat [17] 、APIMiner[18]	4725 个良性样本来自 PlayDrone，5558 个恶意样本来自 DERBIN	修改特征植入触发器，随后打标签和重打包。	使用 1%的毒化数据能使逃逸率升至 90%以上。
2021	[138]	白盒	基于强化学习的结构攻击	基于函数调用图（FCG）的检测器	良性 11613 个，恶意 11583 个，均来自 AndroZoo	对 FCG 的边和节点进行增删改操作来改变图特征。	50 次修改后逃逸率超过 90%，500 次达 99.94%，不限次可达 100%。

（5）硬件恶意代码

实际上，对抗攻击针对的是机器学习模型，与系统无关，故此类攻击具备通用性。比如，有研究人员提出硬件恶意软件检测器（Hardware Malware Detectors，HMD），通过架构特征、内存模式、传感器状态等底层特征进行恶意软件的检测。同样，此类检测器也可以被攻击者绕过。Khasawneh 等[140]验证了针对 HMD 的对抗攻击的可行性。作者构建恶意样本的动态控制流图，在不影响程序执行状态的条件下将额外的指令添加到控制流图中，从而改变样本特征，绕过检测器。随着集成电路的需求增加，恶意第三方植入恶意电路（硬件木马）的威胁日益增加。在硬件木马检测器的对抗上，Nozawa 等[141]提出了基于门电路的对抗攻击，用逻辑上等效的硬件木马电路代替了硬件木马电路，使其难以被检出。

现阶段，对恶意代码的检测通常不仅仅依靠机器学习方法，也依靠传统的静态分析、动态分析和特征码等方法，因此能绕过机器学习检测器的恶意代码不一定能绕过传统的检测器。然而技术是不断发展的，近些年的工作中已经逐渐出现能绕过 VirusTotal 上的反病毒引擎的案例。随着技术的进步，未来几年可能会出现绕过大多数普通杀软的对抗恶意代码。在一些工作中，作者提出了一些防御方法，比如将生成的恶意代码用于检测器的训练，能提升检测器的泛化能力，进而检测出对抗恶意代码。攻防此消彼长、相互博弈，要不断跟进前沿才能达到更好的防御效果。

4.4　执行阶段

4.4.1　命令与控制

攻击者和恶意代码之间需要维持一条命令信道以便攻击者能够远程控制恶意代码的行为。该环节主要面临隐蔽性问题，即恶意代码的行为易被安全设备检测出来、控制者的信息易通过网络连接溯源。借助 AI 技术，攻击者能增强命令与控制环节的恶意代码和控制端身份的隐蔽性。

（1）流量伪装

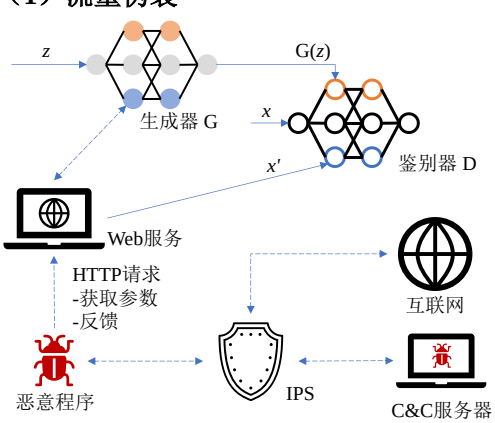


图9 使用 GAN 进行流量模仿[142]

Figure 9　Mimicking Traffic using GAN[142]

[12] Arp D., Spreitzenbarth M., Hubner M., et al. DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket. *NDSS*, 2014: 23-26.
[13] PlayDrone. https://doi.org/10.1145/2591971.2592003
[14] Adagio. https://doi.org/10.1145/2517312.2517315
[15] AndroZoo. https://doi.org/10.1145/2901739.2903508
[16] Mariconti E., Onwuzurike L., Andriotis P., et al. MaMaDriod: Detecting Android Malware by Building Markov Chains of Behavioral Models. *NDSS*, 2017.
[17] DroidCat. https://doi.org/10.1109/TIFS.2018.2879302
[18] DroidAPIMiner. https://doi.org/10.1007/978-3-319-04283-1_6

在一些情况下，恶意软件需要与攻击者进行通信，以完成报活、获取命令、回传数据等任务。这个过程的流量往往具备一定的特征，如访问间隔、数据流的长度、数据流持续时间、数据报文大小等。根据这些行为特征，防御者可以构建相应的恶意行为样式库，使用统计学习等方法对恶意程序的通信流量进行检测。

在对抗检测方面，2018 年，Rigaki 等[142]提出使用 GAN 学习正常软件的通信流量，让恶意程序模仿正常软件的通信行为，进而使检测器检测不出恶意通信流量，如图 9。作者的实验表明，当进行了 200 轮训练后，IPS 对恶意样本通信流量的拦截率降到了 3.16%，当训练 400 轮时，拦截率降为 0。2018 年，Apruzzese 等[143]用对抗样本的方法绕过基于随机森林的检测器，通过增量改变流的持续时间、数据包数量和数据包大小来调整流量特征。实验表明，对于一些家族，将通信持续时间增加 1 秒即可将检测率由 99%以上降至 20%以下，再修改数据包大小和数量，能使检测率降至 1%以下。2019 年，Wu 等[144]使用强化学习技术，通过修改时间戳、添加固定载荷、添加随机载荷等方法改变流特征，能实现对不同的恶意代码家族的不同程度的逃逸，逃逸率可达 80%以上。2020 年，胡永进等[145]使用三种对抗样本方法对基于 LeNet-5 的 CNN 检测器进行绕过。作者在数据预处理后，调用不同的扰动生成方法生成扰动数据，并与原始流量进行叠加。实验表明，FSGM 方法的整体欺骗率达 99.05%。2021 年，Wang 等[146]通过强化学习方法修改流持续时间、时间间隔、数据包的大小、位置和内容来绕过基于 XGBoost 和 BotCatcher（CNN+LSTM）检测器，逃逸率最高达 87%。本方向近些年的代表性工作见表 6[142-149]。

表6　流量检测器逃逸方向的代表性工作

Table 6 Representative Work on Traffic Detector Evasion

年份	文章	场景	主要方法	对抗目标	数据	描述	效果
2018	[143]	灰盒	对抗样本	随机森林检测器	CTU	增量改变流的持续时间、数据包数量和数据包大小。	对一些家族，可将检测率由 99%以上降至 1%以下。
2018	[147]	黑盒	WGAN	机器学习 IDS	NSL-KDD	在恶意流量后添加扰动，使生成器输出对抗流量，黑盒 IDS 输出标签，使用对抗流量和标签训练鉴别器，并将梯度传播至生成器。	检测率从 80%以上降至 1%以下。
2018	[142]	白盒	GAN	机器学习 IPS	抓取的 Facebook 流量	使用 GAN 生成修改流量的关键参数（流间隔时间、流持续时间和数据包大小）模拟 Facebook 聊天流量，欺骗 IPS	当模型训练超过 400 轮时，IPS 对恶意流量的检测率降为 0。
2019	[144]	黑盒	强化学习	决策树和 CNN 检测器	CTU	修改时间戳、添加固定载荷、添加随机载荷等，改变流特征。	对不同的家族有不同的逃逸率，最高可达 80%以上。
2020	[148]	白盒	FGSM	全连接 DNN	MTA	在恶意样本端和 C2 服务端添加代理，收集恶意样本产生的流量，通过代理来改变数据流的持续时间和包长。	整体攻击成功率达 95%。
2020	[145]	白盒	FSGM C&W DeepFool	基于 LeNet-5 的 CNN	Moore	数据预处理后，调用不同的扰动生成方法，生成扰动，并与原始流量进行叠加。	FSGM 方法的整体欺骗率达 99.05%。
2021	[149]	黑盒	WGAN	决策树、随机森林、AdaBoost、GTB	自建数据集	针对流持续时间、流间隔、TLS 版本和流长度进行修改，模拟 Github 访问流量	逃逸率从 1%左右增长至 96%左右。
2021	[146]	黑盒	强化学习	XGBoost、BotCatcher（CNN+LSTM）	CTU、ISOT	修改流持续时间、时间间隔、数据包的大小、位置和内容。	逃逸率最高达 87%。

（2）域名混淆

除了流量特征外，防御者还可以根据 C&C 通信的域名来进行僵尸网络检测。通常来说，正常的网络活动使用的域名大多是由可读的有语义的单词或拼写组成，这类域名往往可以拼读出来，如 office.com、github.com、kugou.com 等，也有一些域名使用单词或拼写的首字母组合而成，如 cctv.com、hgjjgl.com、wsj.com 等，其长度往往有限。然而，在僵尸网络活动中使用的域名通常为使用 DGA 或者 Domain Flux 等技术生成的域名，具备与普通域名不同的长度、元音字符比、文本分布和熵值等特征[150]，因此有研究人员提出使用域名的文本特征和统计特征进行恶意域名识别[151-153]。为了逃逸此类检测器的检测，攻击者可以生成与普通域名的特征相同的域名。2016 年，Anderson 等[154]提出基于 GAN 的 DGA 算法 DeepDGA，能生成高度混淆的域名，使恶意域名分类器对此类域名具备较低的检出率。在域名生成方面，经过 3 轮对抗训练后的模型生成的域名，在字符分布方面和 Alexa Top 1M 高度相似。在对抗检测方面，作者使用基于随机森林的恶意域名检测器对普通 Botnet 家族生成的域名和 DeepDGA 生成的域名进行检测，结果表明，普通僵尸网络家族生成

的域名有 98%的检出率，而 DeepDGA 生成的域名只有 48%的检出率。在 DeepDGA 之后，研究人员提出了更多的检测逃逸方法，如 DomainDGA[155]、MaskDGA[156]、CharBot[157]、ShadowDGA[158]等。

除此之外，还有其他方法生成一些混淆度较高的域名。例如，uriDeep[159]使用机器学习方法生成包含 Punycode 编码的 Unicode 域名，而一些域名能欺骗 Chrome 和 Firefox 的国际化域名 IDN（Internationalized Domain Name）策略。比如，给定目标域名 www.example.org，uriDeep 能生成大量的诸如 www.exampl**e**.org（www.xn--exampl-n21c.org）、www.example.org（www.xn--exampe-mo0b.org）等与目标域名很相似的 IDN 域名。2017 年曾出现过使用 Punycode 进行网络攻击的案例[160]，攻击者注册一个名为 xn--80ak6aa92e.com 的域名，输入到浏览器之后，浏览器会自动将其还原成 apple.com，进而诱导用户上当。

（3）身份隐匿

近些年使用社交网络平台进行命令与控制的情况越来越多，如 Hammertoss[161]、MiniDuke[162]和 Turla[163]等。使用社交网络平台进行命令控制有一些优势，比如不会掉线、不会被防火墙拦截、不必搭建自己的服务器等。其缺点在于，其一，被控端寻址方法要硬编码到被控端程序中，若采用可逆编码（如静态的用户名、ID、URL 等或动态生成算法 DGA），一旦被控端被逆向，控制端的账号会被计算出来，从而使社交平台提前封锁此类账号；其二，控制端发布的命令不具备可读性，会被判定为异常内容，引起社交平台对控制端账号的风控，影响命令的传播。AI 技术能解决这些问题。2020 年，Wang 等[13]提出人工智能赋能的命令控制技术 DeepC2，使用人工智能技术来解决可逆硬编码和异常内容的问题。在寻址上，DeepC2 借助一个神经网络模型，通过识别用户头像的方式找到控制端账号。神经网络通过计算相似度来判断输入的头像是否来自控制端。当被控端被逆向后，安全人员能够获取到模型等信息，但由于模型的单向性和抗碰撞性，安全人员不能提前找到控制端。在异常内容上，DeepC2 使用文本数据增强生成大量语义相似且具备上下文语境的博文，并用哈希碰撞来选取包含命令的博文。对于被控端来说，仅需在找到控制端账号后，计算博文的哈希即可得到命令。

4.4.2 行动

在行动环节，入侵者完成各种攻击任务，对目标系统展开全面攻击。根据 AI 在行动环节的应用，本文将相关工作分为定向攻击、痕迹消除和集群智能三个部分。

（1）定向攻击

攻击者借助 AI 能使攻击定向化和定制化，能高效隐蔽地执行定向攻击。借助 AI 的学习能力，攻击者可以使用目标的属性特征（如人脸、声音、地理位置、传感器信息、设备信息等）训练模型，使用模型的输出作为目标标识。在执行攻击时，只有当模型的输入满足训练时所设定的目标属性时，模型才能准确输出目标标识，使恶意代码精确地执行定向攻击。同时，由于神经网络模型的单向性和抗碰撞性，分析人员无法通过逆向模型或爆破枚举等方式找到攻击目标。

2018 年，Kirat 等[5]在 Black Hat 上提出隐式定向攻击场景 DeepLocker。DeepLocker 使用与目标相关的信息（如目标人脸）进行神经网络模型的训练，使模型的输出为一个固定的密钥，并使用该密钥加密一段恶意载荷。DeepLocker 假设通过供应链攻击等方式，将捆绑有恶意程序的视频通讯软件下发至受害者系统。当使用该软件进行视频通讯时，恶意程序会自动捕捉人脸图像，并输入至神经网络模型，得到模型的输出。若该软件的使用者是攻击目标，模型的输出将会是预先设定的密钥，否则模型的输出是无意义的其他数值。对于分析人员来说，在不知道目标时，分析人员无法通过模型得到正确的密钥，也无法解密被加密的恶意载荷，进而无从知晓恶意程序的攻击意图。DeepLocker 将传统的识别目标方式由“if-then”模式变成了黑盒模式。

在 DeepLocker 的实现上，作者给出了一种方案[164]。首先，攻击者训练一个人脸识别模型 M1。完成训练后，当输入同一个人的不同人脸图像时，M1 会输出相似但不相同的特征向量。然后，借助微调（fine-tune）的思想，在 M1 的基础上构建模型 M2。M2 的输入是目标人脸在 M1 的输出，输出是一个设定的随机密钥（key）。作者指出，仅需准备 10 张左右的目标人脸图像即可完成 M2 的训练。此时，攻击的触发条件和目标都隐藏在神经网络模型中。2020 年，Ji 等[165]提出了一种基于二分类模型的实现方式。模型训练时使用大量目标人脸作为正样本，使用其他人脸作为负样本。训练完成后，当输入为目标人脸时，模型的倒数第二层将得到一个稳定的输出，可用来分桶化（bucketize）得到密钥 key。相比于 DeepLocker，Ji 提供了一种新的触发条件及目标隐藏方式。此方法的局限性在于，密钥 key 不可预知，难以证明其随机性，进而限制了 key 的安全性。此外，

此方法在训练过程中需要大量目标人脸图像，这在实际应用中存在一定的困难。

除了定向化攻击目标人物外，Yu 等[166]提出了一种 GUI 攻击的场景，能定向化攻击应用程序。Yu 等假设攻击者要窃取用户在浏览器中保存的凭证信息，如银行卡账号。恶意程序需要打开浏览器和相应的网页，触发浏览器的自动填写机制，然后窃取敏感信息。作者指出，由于恶意程序可以获取到用户的桌面截图，因此可以使用神经网络识别桌面上的浏览器图标，驱使恶意程序模拟点击行为，打开浏览器。实验表明，桌面浏览器图标的识别准确率可达 98%以上，快速启动栏的图标识别准确率也可达 97%以上。虽然打开浏览器的方式远不止文中提到的这一种，但作者以一个简单的实验展示了人工智能助力定向攻击的新场景，对未来防御此类攻击有参考意义。

（2）痕迹消除

安全人员会根据系统的日志信息进行攻击发现和追踪，而 AI 技术能够更改和生成日志信息，隐藏攻击痕迹。近些年，自动化日志生成工具不断地被研发出来[167-168]。2021 年，Tommel[169]提出使用 GAN 进行 Windows 事件日志的生成。作者认为，自动化生成系统日志需要保证日志内容的正确性和可读性，也要保证日志上下文之间的连贯性，比如进程对文件的读写操作总是发生在进程的启动之后。作者使用 williamSYSU 的框架对三种 GAN 模型 SeqGAN、MaliGAN 和 CoT 进行实现和训练。实验结果表明，GAN 能够以正确的格式生成日志信息，但上下文的连贯性还有待提升。

（3）集群智能

集群智能（Swarm Intelligence）[170]由一组代理或生物群组成，它们在本地相互交流并与环境交互，具有分布式、无中心、自组织的特点。集群智能来源于群居性生物通过协作表现出的宏观智能行为，通过对昆虫间智能集群行为的探索，逐渐涌现了诸多智能集群算法，如蚁群算法（Ant Colony System，ACS）和粒子群优化算法(Particle Swarm Optimization，PSO)。近些年，有研究人员提出，恶意程序可以组建集群智能，在网络攻击中发挥更大作用。类比于蚁群，假设这类恶意程序组建了一个集群僵尸网络，它们没有控制端（Botmaster），可以在不依赖攻击者干预的条件下学习外部知识，动态感知环境变化，分析当前状态，并指导下一步活动。除了应用在网络攻击中，集群智能也能应用在无人机作战[171]中，DARPA[172]通过 CODE、小精灵（Gremlins）和 OFFSET 等项目逐步将 AI 引入无人机集群作战。

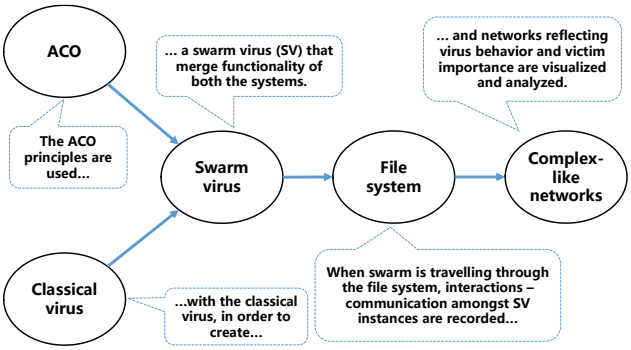


图10集群恶意代码状态转换图[173]

Figure 10 Swarm Virus Behavior Patten[173]

集群智能是恶意代码演进方向之一[174]，相关团队也对其实现进行了探索。2013 年，Castiglione 等[175]提出了基于集群智能的僵尸网络命令控制方法，根据蚁群觅食行为优化控制消息传播过程，提升了僵尸网络的容错和对网络变化的动态适应能力。2014 年，Cani 等[176]提出使用进化算法（Evolutionary Algorithm，EA）让恶意代码免杀，并优化在宿主机上的代码注入过程。2017 年，Danziger 等[177]提出智能自治僵尸网络的构想，将 Bot 节点划分为超级节点（Super Agent）、侦察节点（Recon Agent）、防御节点（Defense Agent）和攻击节点（Attack Agent）四类，把任务目标、攻击方法、逃逸方法、机器学习方法等结合在一起来创建恶意代码。在使用时，自治僵尸网络可以从互联网及被感染设备处使用强化学习等技巧学习当前状态信息，并由超级节点指挥其余节点协作完成攻击任务。2018 年，Zelinka 等[173]提出了一种集群病毒（Swarm Virus）的原型，包括传播、通信、状态转换、数据存储和行为等功能，如图 10。2020 年，Truong 等[178]描述了一种神经集群病毒（Neural Swarm Virus）。该恶意代码包含感染、通信、传播、载荷和触发五个主要模块，能够自我复制，可以用来消除僵尸网络的 C&C 控制中心。不过，囿于算力资源、存储资源、通信资源、数据资源、免杀等条件限制，现阶段攻击者尚不能实现这类“完美”的拟生的恶意代码。在网络产品、设备与服务高度定制化的今天，此类攻击依然有存在和发展的土壤。在遵循科研道德和伦理的前提下，研究人员应该针对此类攻击展开防御领域的工作，以便能及时应对此类攻击，保护网络空间安全。

5 防御与应对

攻击和防御是对立统一的，然而攻击和防御都具备一定的滞后性。防御领域的滞后性主要指从攻

击发起到有效防御措施部署之间的窗口期。防御方需要在攻击发起后，以最快的速度发现攻击并进行有效响应，尽可能地缩小窗口期。为此，提前针对各类攻击进行预演和防御方案设计很有必要。攻击领域的滞后性，在这里主要指 AI 赋能的网络威胁应用在现实网络中的滞后性。尽管 AI 赋能的网络威胁已经覆盖了网络攻击的各个阶段，但因现阶段存在的技术条件限制，一些威胁场景仍然停留在概念验证阶段，不会在短时间内大规模应用。这就为缩小防御的窗口期提供了机会。本章从已有的通用防御方法和面向 AI 威胁的针对性防御方法两个方面来对防御措施进行探讨。

5.1 已有防御方法分析

针对当前网络空间面临的一些威胁，安全人员采取了一系列防御措施。目前的防御措施主要分为两大类，分别是被动防御和主动防御技术。被动防御包括防火墙、入侵检测、恶意代码扫描和网络监控等技术，可以及时发现威胁和脆弱点，降低系统安全风险。主动防御包括零信任、数据加密、蜜罐、审计和态势感知等技术，可以对整个信息系统进行实时监控，捕捉网络及系统异常，对可疑行为进行告警，提升信息系统面对安全威胁时的响应速度。由于攻击者要从目标系统窃取情报或者对目标系统进行破坏、利用目标系统进行恶意代码传播等，要经过信息搜集、漏洞发现、漏洞利用等一系列步骤，如果信息系统能做好日常安全防护，降低安全风险，也能抵御住一些已知的攻击。

表7　常用防御方法的有效性分析

Table 7 Effectiveness Analysis of Defense Methods

攻击阶段	环节	应用场景	常用防御方法	有效性	解释
准备阶段	侦察	目标定位	不在互联网上发布涉及到个人信息的真实数据。	否	若发布有数据，即可使用 AI 方法进行分析。
		密码分析	保证密钥的随机性，不使用特殊密钥[179]。	否	AI 提供了一类通用分析方法。
		敏感信息恢复	增加物理或信号上的干扰，对原有信息进行破坏。	部分	取决于信息处理后的形态。
		攻击路径选择	杜绝内部网络信息泄露[180]。	是	当攻击者获取不到目标网络内部信息时，也就无法寻找最佳攻击路径。
		侧信道信息探测	增加混淆或消除特征[181]。	部分	取决于信息处理后的形态。
	武器化	信息伪造	增强员工安全意识[180]；使用 AI 方法进行伪造信息检测[12]等。	否	人是薄弱环节；AI 方法也可被攻击。
		口令破解	设置规则，对短时间内多次登录失败的行为进行拦截；引入双因子身份认证（2FA）[182]。	是	AI 方法也需要进行多次试探才能得到正确口令。
		验证码识别	增加混淆与变形，使用基于行为或传感器的验证码[73]。	否	要保证普通人能通过验证码，则验证码必有一定的特征。
		防火墙绕过	设置规则，对短时间内多次探测的行为进行拦截。	是	AI 方法也需要进行多次试探才能得到实现绕过。
入侵阶段	投递	信息隐藏	隐写分析与隐写检测[92]。	是	深度学习方法生成的含密图像隐蔽性差。
	利用	漏洞利用	遵循安全开发标准进行系统或程序的开发[183]。	否	若应用程序存在漏洞，则在一定条件下可以使用 AI 技术进行漏洞发现。
	植入	恶意代码检测逃逸	基于静态分析、动态分析或启发式检测方法。	是	具备逃逸行为的恶意代码也要保证原有恶意功能不被破坏，进而能用传统方法进行检测。
执行阶段	命令与控制	流量伪装	异常流量特征发现。	是	AI 在修改的特征的同时会引入新的特征。
		域名混淆	异常 DNS 流量特征发现[184]。	是	在被控端探测可访问的域名时，AI 方法生成的域名也会产生有规律的 NXDomain 流量。
		身份隐匿	社交网络平台进行用户风险行为和可疑行为的动态监测[13]。	是	控制端账号离不开社交网络或相关平台。
	行动	定向攻击	在可控环境下遍历组织内的相关目标属性，寻找可能的攻击目标。	是	若能及时做到威胁发现，则在一定程度上可以应对此类攻击。
		痕迹消除	对敏感文件的敏感操作进行拦截[185]。	是	若能拦截住对敏感文件的修改，则能防御此类攻击。
		集群智能	从网络层面进行特征分析和拦截。	是	在牺牲精确率的条件下提升召回率，可以防范此类攻击。

对于现阶段已出现的 AI 赋能的网络威胁，本文总结常用的防御方法，并对其是否能防御此类威胁进行讨论；针对尚未应用或还没有防御方法的一些新型攻击，本文从传统思路出发，给出可能的防御方案。结合智能化网络威胁框架，本节对 AI 赋能的网络威胁的防御方法有效性进行探讨，相关结果如表 7。表中“常用防御方法”是现有对相应网络威胁应用场景的一些防御方法；“有效性”表示传统防御方法能否应对相应的 AI 赋能的网络威胁，其中“是”表示该防御方法能应对此类威胁，“否”代表该方法不能应对此类威胁，“部分”代表该方法在一定条件下可以应对相应的威胁；“解释”是对能否防御相关威胁的解释。

从表中可以看出，已有的防御方法能够对抗一部分 AI 赋能的网络攻击。但同时也能看出，尽管现阶段一些防御手段可以对一类攻击进行防御，但这类攻击却还是网络信息安全的一个重要威胁，如口令探测和防火墙绕过。对于这类攻击，AI 做的是把已有的工作自动化和高效化，其主要作用是提升了攻击的效率，因此对攻击的危害和防御本身并没有本质影响，所以已有防御手段仍能对 AI 赋能的网络威胁进行防御。“安全是三分技术七分管理”，已有安全措施能防御住的网络威胁仍成为当前面对的隐患之一，说明信息系统的网络安全建设仍有很长的路要走。

一些防御场景虽然能对相关攻击进行防御，但防御方需要为此付出较大的代价。比如身份隐匿场景，这里主要针对 DeepC2 进行防御。尽管社交网络平台能够在第一时间内判断用户上传的头像是否为目标头像，但对一个平台的所有用户上传头像行为都进行检测需要较大的计算力。同时，如果攻击者发布的恶意代码包含多个版本的神经网络模型，那么社交网络平台的计算量将成倍增加。同样，对于集群智能的防御，尽管可以从网络流量层面对集群恶意代码进行识别，但当这些恶意代码使用一些不具备独特特征的通用协议时，这类防御措施可能会对普通主机的正常网络行为进行误报。虽然在牺牲精确率的条件下能够实现对此类网络行为进行告警，但高误报率对普通用户也有一定的影响。

在部分场景中，要兼顾用户体验和安全性。用户体验度越高，则安全性的损失将越大；反之，用户体验度较低时，能实现较高的安全水平。典型的代表是验证码识别。在不使用验证码的场景中，用户体验最好，而安全性则有损失。在使用简单的验证码的情况下，用户体验有所折衷，但安全性有所增强。当对验证码进行各种形变和混淆，甚至使用问答交互类验证码后，用户体验下降明显，但其安全性得到进一步增强。然而，网站采用验证码还是要让用户识别出验证码的内容。由于 AI 模拟的即是用户识别验证码的过程，当用户可以识别验证码时，AI 也能识别验证码。这个过程彷佛陷入了死循环，成为了一个无解的过程。这是由人的参与带来的不确定性。

从表格中也可以看出，不能防御住 AI 赋能的网络威胁的场景通常和人密切相关，如目标定位涉及到人在各类网络平台发布的各种信息，敏感信息恢复取决于人将信息破坏到什么程度，信息伪造更是针对人而进行的社会工程攻击。上文提到的网络安全管理也和人息息相关。由此可见，人是网络攻击中的薄弱一环。有人参与的环节往往会具备独特的不确定性和脆弱性。因此，减少网络安全中的包括人为因素在内的不确定性也是防御工作的一个方向。

5.2 针对性防御方法探讨

当前针对 AI 赋能的网络安全威胁的防御工作相对较少，由于各类型攻击的技术手段、作用对象以及攻击目的都有所不同，因此很难形成一套通用的防御方法，需要根据实际的威胁类型和情况构建具有针对性的防御方法。Alavizadeh 等[186]提出根据 AI 的作用将 AI 赋能的网络威胁分为 AI 协助的（AI-aided）和 AI 嵌入的（AI-embedded）两类。AI 协助的威胁指使用 AI 技术让攻击更高效，AI 嵌入的威胁指利用 AI 本身来执行威胁任务。在 AI 协助的威胁中，攻击者要对目标系统有先验知识，要有效应对资源限制，主要包括信息搜集、目标选择、攻击准备等任务。在 AI 嵌入的攻击中，AI 的能力被嵌入到恶意代码或威胁中，例如 DeepLocker 和 DeepC2。对应 AI 的内生特性，AI 辅助类使用的主要是 AI 的学习能力和决策能力，而 AI 嵌入类使用的是 AI 模型的性质，把 AI“嵌入”到了威胁任务中。作者在[186]中针对AI协助类网络威胁提出了基于马尔可夫博弈模型（Markov Game Model）的缓解方法，从理论上证明了防御基于 AI 的网络威胁的可行性。针对 AI 嵌入类的攻击场景 DeepC2，作者在[187]中也从博弈理论入手，对攻防参与者及元素进行建模，给出了在双方实现纳什均衡（Nash Equilibrium）的条件下能够成功进行防御的条件。

针对 AI 赋能的网络安全威胁防御方法，本文提出从场景、技术和系统等三个方面入手展开防御工作。

5.2.1 针对场景的防御

针对攻击场景展开分析，可以发现攻击面的薄弱环节，从而进行针对性地防御。一般而言，一个攻防场景需要包含多个元素，如果攻击者引入 AI 对某些元素进行了赋能，那么防御者可以从其他元素入手展开防御。比如命令控制场景有控制端、被控端和通信信道三个元素。在 DeepC2 中，攻击者引入 AI 完成控制端的命令发布和被控端的寻址，增强了通信信道的防溯源和隐匿性。然而，对于控制端所在的社交网络平台和被控端所在的宿主环境，攻击者却没有进行 AI 赋能。防御者可以在被控端上检测恶意程序的静态特征或动态特征、在社交网络平台上检测控制端的自动化行为等，从两端入手，结合场景来打击恶意活动。正如表 7 中提到的防御措施一样，在 DeepC2 的场景中，攻击者无法绕过社交网络平台或其他平台，这是 DeepC2 的一个特色，因此防御者能从平台入手进行防御。

同理，在另外一类场景中，攻击者对模型的训练过程需要对目标系统进行不断试错，而真实场景中，一旦试错失败，攻击者面临的可能是访问行为的受限和模型训练的终止。例如，在防火墙绕过的案例中，攻击者会生成不同的载荷来对 WAF 规则进行持续地试探，这势必会引起 WAF 的告警；防御者则可以调整 WAF 对探测行为的容忍阈值，在识别到有规律的连续探测行为后，对探测者和探测行为进行拦截或进一步的限制，以此消减此类威胁。

5.2.2 针对技术的防御

针对技术的防御主要指针对威胁场景中使用的各种技术进行防御。比如，针对威胁场景中的 AI 技术，防御者可以使用基于 AI 的防御方法，把 AI 引入防御工作中，用 AI 防 AI。例如，针对检测逃逸行为，防御者可以使用多分类器增加逃逸难度、使用权重正则或 Dropout 等方法来降低模型对对抗样本的敏感性、使用对抗样本对模型进行增量训练以及使用更健壮的特征来训练检测模型等[188]。在实际防御应用中，Grosse 等[136]使用增量训练的方式，在使用良性样本和恶意样本训练检测模型后，构建对抗样本并标记为恶意样本，用对抗样本对检测模型进行迭代增量训练，以提升模型的泛化性能。实验表明，在选取适当的参数时，模型对恶意样本的误报率有明显的下降。Zhang 等[189]通过选择更健壮的训练特征来防御恶意 PDF 逃逸攻击，能够在相同条件下提高模型对恶意样本识别的准确率。对于 DeepLocker 来说，攻击者使用 AI 模型进行定向攻击，那么防御者可以针对 AI 模型展开防御，比如借助模型逆向攻击[190]提取模型中有关目标的特征，在组织范围内寻找潜在的目标，解开载荷。

此外，防御者也可以使用一些技巧性的操作来进行防御。比如，同样针对检测逃逸类恶意代码，防御者可以使用冗余异构和多模表决的方式来增强检测结果的可靠性、对模型的输入进行变形（如去混淆、增加偏移等）来弱化可学习的特征、使用神经网络蜜罐[191]判断输入是否属于对抗样本等方式来进行增强防御能力。Chen 等[192]通过对模型的输入进行变形来防御逃逸攻击，作者提取 Android 应用的特征并转换为二进制特征向量，量化表示，将其转换为实数值，并使用压缩来减少对抗操作的影响。从实验效果看，该方法能防御住基于特征选择的逃逸攻击和 FGSM 攻击。

5.2.3 针对系统的防御

“木桶理论”和“世上没有绝对安全的系统”对攻击系统也适用，攻击系统也一定有其薄弱的方面。针对系统的防御主要指对攻击系统进行检测、对攻击行为进行发现和对信息系统进行防护。防御者在掌握了攻击系统的一些信息后，可以对攻击系统的薄弱环节进行分析，甚至也可以用 AI 进行攻击系统的漏洞发现，分析攻击系统的脆弱性，进而找出其短板，进行防御。例如，集群智能恶意代码往往会构建自己的通信系统和信息交流协议，会与网络中的其它节点进行状态信息交换和命令传递，这些恶意代码构成了一个智能集群系统。针对这类系统，防御者可以分析它们使用的网络协议，从异常网络流量的角度来进行检测。防御者也可以在宿主设备上通过 API 调用序列来判断异常的网络行为和文件行为，进而找出信息系统中“潜伏”的恶意代码。集群恶意代码可能涉及到状态转换，防御者还可以分析此类恶意代码有无协议上或实现上的漏洞，采用类似女巫攻击（Sybil Attack）的方式，主动向其投递攻击载荷，使其失效。

在日常的网络安全维护中，防御者应当严格遵守网络安全管理各项规定，对各级各类型网络设备和服务进行安全检查和防护。在面对真实攻击场景时，防御者可对攻击的实际情况进行分析，采用上述的一项或多项措施，从多个角度寻找可能的防御方案。

6 讨论与展望

6.1 未来的发展趋势

人工智能本身是一个数据分析方法，是一个能兼容多种数据类型、发现数据规律的分析方法，而

不是解决所有问题的“良药”。人工智能的复杂结构拟合了一些高级数学表达式，因此能更好地发现数据规律，预测数据趋势。这一特性也带来了人工智能的可解释性问题。在统计学习领域，各类问题能用简明的数学原理进行解释。而在深度学习领域，研究人员暂时不能将复杂的模型转换为清晰明了的数学公式，只能形而上地分析它的原理。这种对神经网络模型认识的模糊性带来了人工智能的脆弱性，进而可以将人工智能引入网络威胁中，将网络威胁智能化。想要防御和消减这类威胁，需要安全人员对网络威胁及 AI 原理和模型都有深入的了解。这无疑是对安全人员的一个挑战和考验。本文认为，AI 赋能的网络威胁有以下发展趋势。

基于 AI 的网络威胁场景将随着新技术的出现不断更新。一方面，AI 与传统网络安全技术结合，例如与僵尸网络技术结合的攻击场景 DeepC2、流量伪装和域名混淆等，将进一步加剧现有网络安全的风险，甚至将网络威胁带到物理世界。另一方面，新兴技术和概念不断出现，如区块链、元宇宙、NFT 等，AI 与新兴技术结合，在带来新应用的同时，也可能产生新的网络安全威胁。

AI 自身的安全问题也可能被攻击者利用。人工智能自身存在大量安全风险[193]，如数据投毒、后门攻击、模型窃取等。这些安全风险具备被恶意利用的可能，比如对抗样本攻击、针对 OSINT 进行数据投毒[52]等。还有一类将 AI 的安全风险当作“特性”进行攻击的场景，即“把 bug 当 feature”。例如，当用在积极面时，AI 的后门攻击可以是保护 AI 模型知识产权的模型水印；当用在消极面时，后门攻击可以成为攻击者的跳板。此外，在数据安全与隐私保护大背景下，针对 AI 模型的数据安全攻击也将成为一个重要方向，比如成员推理攻击[194]、模型逆向攻击[190]、梯度隐私泄露[195]等，可以从模型或梯度中泄露包含敏感内容的训练数据，使包括联邦学习[196]在内的 AI 系统面临数据安全威胁。

利用 AI 攻击 AI 系统可能成为趋势。随着人工智能技术被广泛应用在各个领域，未来可能会有大规模的针对自然语言处理、图像分类、语音分析和视频追踪等系统的攻击，而在这些攻击中起到主要作用的是 AI。比如，在针对图像分类的对抗样本攻击中，对抗样本中添加的噪声数据可以由 AI 生成；在针对自然语言处理系统的投毒攻击中，毒化数据也可以由 AI 生成，从而形成 AI 攻击 AI 的趋势。

6.2 AI 网络威胁有效性探讨

AI 能够在网络威胁中起到重要的推动作用，能够弥补当前网络攻击在各阶段的短板。然而，我们也要看到，囿于软硬件和计算资源，在短时期内，AI 赋能的网络威胁不会大规模出现在实际的网络攻击活动中。在此，我们要探讨 AI 赋能的网络威胁究竟能在实际攻击中起到多大作用。

本文列举的工作大多始于 2016 年至 2018 年。如果说 2016 至 2018 是 AI 网络威胁起步阶段的话，近些年 AI 在网络威胁里的应用场景更新几乎止步。除了上述的外部条件限制外，还应有其他原因。本文假设其为动力原因。究其本，本文虽提及众多应用场景，但对一个网络威胁活动起到至关重要作用的场景却很少。若对这些应用场景进行评级划分，让攻击者选择适用于特定攻击任务的应用场景，则恐难择一二。

为何会出现如此结果呢？从图 3 可以发现，在准备阶段，AI 起到的作用主要是提升攻击者的效率，比如目标定位是为了更快发现目标，密码分析是为了更快破解密文，敏感信息恢复更是为了更快得到敏感信息。从攻击路径选择到验证码识别和防火墙绕过，这类工作无一例外都是为了提升攻击者的效率，使攻击更快速精准。诚然，在当今时代，效率的提升也是一大进步，可这里使用的 AI 终究没有起到无可替代的作用，AI 对攻击本身没有实质性的影响。如果说准备阶段攻击者还处于信息系统的外部，无法接触到信息系统的数据，不能执行关键攻击任务的话，那在入侵阶段和执行阶段的大部分案例中，AI 的作用却也显得有限。这里就存在一个悖论。尽管恶意代码能在目标系统接触内部数据，但攻击者并不能预先获取这些数据，且不同信息系统之间的网络架构、布局和功能差异较大，攻击者难以拥有足够的可以模拟目标系统的数据来训练相应的模型。这势必会限制 AI 网络威胁的发展。从文中的案例也能看出，18 种已有的案例中，准备阶段占据 9 种，入侵和执行阶段缺少更丰富的应用场景。与此同时，也应注意到，AI 是发现规律学习规律的工具之一，而其他具备此类功能的工具也能用来替代 AI，进而起到相同的效果。在这些场景中，AI 能对攻击起推动作用，但都不是决定性作用。

从防御角度来看，表 7 中不能防御住 AI 赋能的网络威胁的方法多集中在准备阶段，这也是因为防御者只能在自己力所能及的范围内展开防御，而不能将手臂伸至系统之外。反之，能防御住 AI 威胁的方法通常能直接对信息系统内部的配置、流量、文件等数据进行操作，在威胁发生时有较强的自主性，能对此类威胁进行及时有效应对。

从另一个角度来看，前文也提到 Alavizadeh 等[186]对 AI 赋能的网络威胁进行分类，其中 AI 嵌入类网络威胁把 AI“嵌入”到威胁任务中。Wang[197]也提出类似的分类方法，并指出在类似 AI 嵌入的网络威胁中，AI 起到的作用是“不可替代”的，而在类似 AI 协助的网络威胁中，AI 的作用可以被其他方法所替代的。在二者的分类中，AI 嵌入类均有 DeepLocker 和 DeepC2 两个案例。在本文所列举的案例中，这两个也是为数不多的主要利用 AI 模型性质的案例。那么使用 AI 模型的性质就能做到“不可替代”吗？不尽然。DeepLocker 中要保证信息处理过程的不可逆，要保证处理方法具有健壮性和稳定性。在实际操作中，攻击者可以使用哈希的衍生算法来实现此类功能。在 DeepC2 中，若攻击者使用社交网络平台的其它用户属性进行控制端身份识别，也能规避神经网络模型的使用。

攻击者欲使用 AI 赋能网络威胁，就要知道哪些操作可以被 AI 赋能。诸如建立 TCP 连接、扫描、删除文件这种原子类型的操作不会也没必要使用 AI 进行赋能，而决定什么时候发起连接、扫描时发送哪些载荷、删除什么文件则存在被 AI 赋能的可能。由此可见，AI 的赋能点一定是有学习和决策参与的点，这也是前文总结的由 AI 增强的学习能力和决策能力。由此对应到赋能矩阵，防御人员可以更好地理解 AI 在网络威胁中的作用，同时加强对未知的威胁进行研判和预防。

7 结束语

本文从 AI 的能力和 AI 模型的性质出发，结合杀伤链模型，系统梳理了人工智能赋能的网络威胁研究现状，同时提出了防御建议，并对未来的发展和 AI 网络威胁的有效性进行了探讨。众所周知，攻击和防御是对立统一的，并在一定条件下能相互转换。目前，攻防双方的能力、掌握的信息量、可用的资源都存在不对等的关系，这给智能化网络攻防带来了机遇与挑战。一方面，攻击者可以利用人工智能不断赋能网络威胁，增强恶意代码的能力；另一方面，软硬件及数据等限制条件阻碍了智能化网络威胁的应用，使防御者有机会进行防御准备。然而，无论攻防如何发展，其核心依旧是人。不管是密码学算法、虚拟化方法还是人工智能方法，它们都是攻防人员的工具，被用来增强双方的能力。未来，人工智能等技术将得到进一步发展，当前在计算资源、存储资源、模型性能、框架安全等方面的能力限制也将得到解决。此时，网络威胁不会消失，攻防博弈双方仍将互相促进，互相牵制，互相发展。我们期望未来的防御能力会构筑在信息系统自身中，以增强系统安全性，创建更安全可靠的网络世界。

参考文献

[1] Brundage M , Avin S , Clark J , et al. The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation [EB/OL]. 2018: *ArXiv Preprint ArXiv*: 1802.07228.

[2] Beyondtrust. BeyondTrust Releases Cybersecurity Predictions for 2021 and Beyond[EB/OL]. https://www.globenewswire.com/news-release/2020/10/28/2115996/0/en/BeyondTrust-Releases-Cybersecurity-Predictions-for-2021-and-Beyond.html. 2020.

[3] Cearley D, Jones N, Smith D, et al. Top 10 Strategic Technology Trends for 2020[EB/OL]. https://emtemp.gcom.cloud/ngw/globalassets/en/doc/documents/432920-top-10-strategic-technology-trends-for-2020.pdf. 2019.

[4] Fortiguard. New Cybersecurity Threat Predictions for 2021[EB/OL]. https://www.fortinet.com/blog/threat-research/new-cybersecurity-threat-predictions-for-2021. 2020.

[5] Kirat D, Jang J, Stoecklin M. DeepLocker–Concealing Targeted Attacks with AI Locksmithing[J]. *Black Hat USA*, 2018.

[6] Isao Takaesu. Deep Exploit[EB/OL]. https://github.com/13o-bbr-bbq/machine_learning_security/tree/master/DeepExploit. 2018.

[7] Manky D. Order vs. Mad Science Analyzing Black Hat Swarm Intelligence[EB/OL]. https://published-prd.lanyonevents.com/published/rsaus18/sessionsFiles/8423/HT-W02_Order-vs-Mad%20Science-Analyzing-Black-Hat-Swarm-Intelligence.pdf. 2018.

[8] Eugene Ching. Understanding the 6 major capabilities of AI[EB/OL]. https://medium.com/qavar/understanding-the-6-major-capabilities-of-ai-efea8e361d06. 2020.

[9] Neil Sahota. The Next Frontier is here: 3 Key Capabilities that Make AI so Valuable[EB/OL]. https://news.itu.int/the-next-frontier-is-here-3-key-capabilities-that-make-ai-so-valuable/. 2017.

[10] Schmid, T., Hildesheim, W., Holoyad, T. et al. The AI Methods, Capabilities and Criticality Grid[J]. *Künstl Intell*, 2021, 35: 425–440. DOI: 10.1007/s13218-021-00736-4.

[11] 邱锡鹏. 神经网络与深度学习[M/OL]. https://nndl.github.io/. 北京: 机械工业出版社, 2020: 3-4.

[12] Yisroel Mirsky and Wenke Lee. 2020. The Creation and Detection of Deepfakes: A Survey[J]. *ACM Computing Survey*, 2020: 54, 1, Article 7.

[13] Wang Z., Liu C., Cui X., et al. DeepC2: AI-Powered Covert Command and Control on OSNs[C]. In *24th International Conference on Information and Communications Security (ICICS)*, Springer, 2022.

[14] Ping Guolou, Ye Xiaojun. A Survey of Research on Network Attack Model[J]. *Chinese Journal of Information Security Research*, 2020, 6(12): 1058-1067.
(平国楼 叶晓俊. 网络攻击模型研究综述[J]. 信息安全研究, 2020, 6(12): 1058-1067.)

[15] Hutchins, Eric M., Michael J. Cloppert, and Rohan M. Amin. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains[J]. *Leading Issues in Information Warfare & Security Research*, 2011, 1(1): 113-125.

[16] Caltagirone S., Pendergast A., Betz C. The Diamond Model of Intrusion Analysis [R]. Center for Cyber Intelligence Analysis and Threat Research Hanover Md, Tech Rep: 0704-0189, 2013.

[17] Strom B. E., Applebaum A., Miller D. P., et al. MITRE ATT&CK: Design and Philosophy[R]. The MITRE Corporation, Tech Rep: MP180360, 2018.

[18] ODNI. A Common Cyber Threat Framework: A Foundation for Communication[EB/OL]. https://www.dni.gov/files/ODNI/documents/features/A_Common_Cyber_Threat_Framework_Overview.

pdf. 2017.
[19] Liu D., Wu Q., Han W., et al. User Identification across Multiple Websites Based on Username Features[J]. *Chinese Journal of Computers*, 2015, 38(10):13.
(刘东, 吴泉源, 韩伟红,等. 基于用户名特征的用户身份同一性判定方法[J]. 计算机学报, 2015, 38(10):13.)
[20] Li S., Wang J., Zhou G. Method and system for identifying same user under two different platforms[P]. Chinese Patent: CN 104778388A, 2015-07-15.
(李寿山, 王晶晶, 周国栋. 一种两个不同平台下同一用户识别方法及系统[P]. 中国专利: CN104778388A, 2015-07-15.)
[21] ThoughtfulDev. EagleEye[EB/OL]. https://github.com/ThoughtfulDev/EagleEye. 2018.
[22] SEYMOUR J, TULLY P. Weaponizing Data Science for Social Engineering: Automated E2E Spear Phishing on Twitter[J]. *Black Hat USA*, 2016, 37:1-39.
[23] Liu Y., Luo X., Li H. Microblog User Location Inference Based on POI and Query Likelihood Model[C]. *International Conference on Information and Communications Security (ICICS 2021)*, Springer, 2021: 464-480.
[24] Ge Z., Hu H. Confluence of Neural Networks and Cryptography: A Review[J]. *Chinese Journal of Cryptologic Research*, 2021, 8(2): 215-231.
(葛钊成, 胡汉平. 神经网络与密码学的交叉研究[J]. 密码学报, 2021, 8(2): 215-231.)
[25] Alallayah K. M., Amin M., Abd El-Wahed W. F., et al. Attack and Construction of Simulator for Some Cipher Systems using Neuro-Identifier. *Int. Arab J. Inf. Technol.*, 2010, 7(4): 365-372.
[26] Mohammed M. Alani. Neuro-Cryptanalysis of DES and Triple-DES[C]. *19th International Conference on Neural Information Processing*, Springer, 2012: 637-646.
[27] Sam Greydanus. Learning the Enigma with Recurrent Neural Networks[EB/OL]. 2017: *ArXiv Preprint ArXiv*: 1708.07576.
[28] 腾讯安全 AI Lab. 利用 AI 技术进行碎纸拼接复原[OL]. https://mp.weixin.qq.com/s/gDdehK3tBkVWG54-PkKf4g. 2021.
[29] 腾讯安全 AI Lab. 用 AI 去除马赛克[OL]. https://mp.weixin.qq.com/s/U41iuhluGepdYKZRR6R8rQ. 2020.
[30] aisecstudent. TextDemosaicing[OL]. https://github.com/aisecstudent/TextDemosaicing. 2020.
[31] 腾讯安全 AI Lab. 从联邦学习中恢复隐私数据[OL]. https://mp.weixin.qq.com/s/YRvw7SJzELZyQWOzQibWVQ. 2021.
[32] Yousefi M., Mtetwa N., Zhang Y., et al. A Reinforcement Learning Approach for Attack Graph Analysis[C]. *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, IEEE, 2018: 212-217.
[33] Wu R., Gong J., Tong W., et al. Network Attack Path Selection and Evaluation Based on Q-Learning[J]. *Applied Sciences*, 2021, 11(1): 285.
[34] D. Chen, Z. Zhao, X. Qin, et al. MAGLeak: A Learning-Based Side-Channel Attack for Password Recognition With Multiple Sensors in IIoT Environment[J]. *IEEE Transactions on Industrial Informatics*, 2022, 18(1): 467-476. DOI: 10.1109/TII.2020.3045161.
[35] Hospodar, G., Gierlichs, B., De Mulder, E., et al. Machine learning in side-channel analysis: a first study[J]. *Journal of Cryptographic Engineering*, 2011, 1(4): 293-302.
[36] Yu, W. and Chen, J. Deep learning-assisted and combined attack: a novel side-channel attack[J]. *Electronics Letters*, 2018, 54(19): 1114-1116.
[37] Yu, W. Convolutional neural network attack on cryptographic circuits[J]. *Electronics Letters*, 2019, 55(5): 246-248.
[38] Wen, Y. and Yu, W. Boosting the efficacy of power attacks on cryptographic circuits with autoencoder[J]. *Electronics Letters*, 2019, 55(23): 1221-1224.
[39] Yu, W. Hardware Trojan attacks on voltage scaling-based side-channel attack countermeasure[J]. *IET Circuits, Devices & Systems*, 2019, 13(3): 321-326.
[40] Carré, S., Dyseryn, V., Facon, A., et al. End-to-end automated cache-timing attack driven by Machine Learning[C]. *Proceedings of 8th International Workshop on Security Proofs for Embedded Systems*, 2019, 11: 1-16.
[41] Herrmann, D., Wendolsky, R., Federrath, H. Website Fingerprinting: Attacking Popular Privacy Enhancing Technologies with the Multinomial Naïve-Bayes Classifier[C]. *Cloud Computing Security Workshop*, 2009:.31-42.
[42] T. Wang, and I. Goldberg. Improved website fingerprinting on Tor[C]. *ACM Workshop on Workshop on Privacy in the Electronic Society*, ACM, 2013: 201-212.
[43] T. Wang, X. Cai, R. Nithyanand, et al. Effective Attacks and Provable Defenses for Website Fingerprinting[C]. *USENIX Security Symposium*, USENIX, 2014: 143–157.
[44] A. Panchenko, F. Lanze, A. Zinnen, et al. Website Fingerprinting at Internet Scale[C], *Network and Distributed System Security Symposium (NDSS)*, IEEE, 2016: 1–15.
[45] Rimmer V., Preuveneers D., Juarez M., et al. Automated Website Fingerprinting through Deep Learning[C]. *Network and Distributed System Security Symposium*, 2018.
[46] Jansen, R., Vaidya, T., Sherr, M. Point Break: A Study of Bandwidth Denial-of-Service Attacks against Tor[C]. *28th USENIX security symposium*, USENIX, 2019: 1823-1840.
[47] Giovanni Cherubin, Rob Jansen, Carmela Troncoso. Online Website Fingerprinting: Evaluating Website Fingerprinting Attacks on Tor in the Real World[C]. *31st USENIX Security Symposium*, USENIX, 2022.
[48] La Cour A.S., Afridi K.K., Suh G.E. Wireless Charging Power Side-Channel Attacks[C]. *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, ACM, 2021: 651-665.
[49] Gong J., Zhang X., Ren J., et al. The Invisible Shadow: How Security Cameras Leak Private Activities[C]. *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, ACM, 2021: 2780-2793.
[50] 腾讯安全 AI Lab. 利用 AI 听音辨踪[OL]. https://mp.weixin.qq.com/s/CAHXfiue_nPRJmPl0uteAw. 2021.
[51] Baki S, Verma R M, Mukherjee A, et al. Scaling and Effectiveness of Email Masquerade Attacks: Exploiting Natural Language Generation[C]. *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security (AsiaCCS 2017)*, ACM, 2017: 469-482.
[52] Ranade P., Piplai A., Mittal S., et al. Generating Fake Cyber Threat Intelligence Using Transformer-Based Models[EB/OL]. 2021: *ArXiv Preprint ArXiv*: 2102.04351.
[53] 腾讯安全 AI Lab. 使用 AI 模仿人类笔迹[EB/OL]. https://mp.weixin.qq.com/s/Og7hshjr7KR2oOzMCqg-Cw. 2021.
[54] Rebryk Y., Beliaev S. ConVoice: Real-Time Zero-Shot Voice Style Transfer with Convolutional Network[EB/OL]. 2020: *ArXiv Preprint ArXiv*: 2005.07815.
[55] Yuan X, Chen Y, Zhao Y, et al. CommanderSong: A Systematic Approach for Practical Adversarial Voice Recognition[C]. *27th USENIX Security Symposium*, Usenix, 2018: 49-64.
[56] Emily Wenger, Max Bronckers, Christian Cianfarani, et al. "Hello, It's Me": Deep Learning-based Speech Synthesis Attacks in the Real World[C]. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, ACM: 235-251.
[57] Catherine Stupp. Fraudsters Used AI to Mimic CEO's Voice in Unusual Cybercrime Case[EB/OL]. *The Wall Street Journal*. https://www.wsj.com/articles/fraudsters-use-ai-to-mimic-ceos-voice-in-unusual-cybercrime-case-11567157402. 2019.
[58] GoUpSec. 3500万美元的深度伪造第一大案：高管语音被克隆[EB/OL]. 安全内参. https://www.secrss.com/articles/35142. 2021.
[59] Yu Chen, Bin Ma, Zhuo Ma. Biometric Authentication Under Threat: Liveness Detection Hacking[J]. *Black Hat USA*, 2019.
[60] Simen Thys, Wiebe Van Ranst, Toon Goedeme. Fooling Automated Surveillance Cameras: Adversarial Patches to Attack Person Detection[EB/OL]. 2019: *ArXiv Preprint ArXiv*: 1904.08653.
[61] 机器之心. GeekPwn 大赛又出新招，各路极客在 AI 之眼下实现“隐身”[OL]. https://www.jiqizhixin.com/articles/2019-10-24-11.

2019.
[62] 朱芸阳. 合法合规，才能“ZAO”[EB/OL]. https://mp.weixin.qq.com/s/hbNe3K3aQ5oLqTIazsV1pw. 2019.
[63] Twitter (Mikael Thalen). https://twitter.com/MikaelThalen/status/1504123674516885507
[64] Bahnsen A C, Torroledo I, Camacho L D, et al. DeepPhish: Simulating Malicious AI[C]. *2018 APWG Symposium on Electronic Crime Research (eCrime)*. 2018: 1-8.
[65] Salminen J., Jung S., Jansen B. The Future of Data-driven Personas: A Marriage of Online Analytics Numbers and Human Attributes[C]. ICEIS (1), 2019: 608-615.
[66] Tamaghna Basu. How I Created My Clone Using AI - Next-Gen Social Engineering[J]. *Black Hat USA*, 2020.
[67] Hitaj B., Gasti P., Ateniese G., et al. PassGAN: A Deep Learning Approach for Password Guessing[C]. *17th International Conference on Applied Cryptography and Network Security*, Springer, 2019: 217-237.
[68] Melicher W., Ur B., Segreti S. M., et al. Fast, Lean, and Accurate: Modeling Password Guessability Using Neural Networks[C]. *25th USENIX Security Symposium (USENIX Security 16)*, 2016: 175-191.
[69] Xia Z., Yi P., Liu Y., et al. GENPass: A Multi-Source Deep Learning Model for Password Guessing. *IEEE Transactions on Multimedia*, 2019, 22(5): 1323-1332.
[70] D. Pasquini, A. Gangwal, G. Ateniese, et al. Improving Password Guessing via Representation Learning[C]. *2021 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2021: 1382-1399. DOI: 10.1109/SP40001.2021.00016.
[71] Ming Xu, Chuanwang Wang, Jitao Yu, et al. Chunk-Level Password Guessing: Towards Modeling Refined Password Composition Representations[C]. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, ACM, 2021: 5-20.
[72] P. Bontrager, A. Roy, J. Togelius, et al. DeepMasterPrints: Generating MasterPrints for Dictionary Attacks via Latent Variable Evolution[C]. *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, IEEE, 2018: 1-9. DOI: 10.1109/BTAS.2018.8698539.
[73] Meriem Guerar, Luca Verderame, Mauro Migliardi, et al. Gotta CAPTCHA ’Em All: A Survey of 20 Years of the Human-or-computer Dilemma[J]. *ACM Computing Survey*, 2021: 54, 9, Article 192. DOI: 10.1145/3477142.
[74] Kopp, Martin, Matej Nikl, and Martin Holena. Breaking CAPTCHAs with Convolutional Neural Networks[C]. *Proceedings of the 17th Conference on Information Technologies – Applications and Theory (ITAT 2017)*, 2017, 93-99.
[75] P. Wang, H. Gao, Q. Rao, et al. A Security Analysis of Captchas With Large Character Sets[J]. *IEEE Transactions on Dependable and Secure Computing*, 2021, 18(6): 2953-2968. DOI: 10.1109/TDSC.2020.2971477.
[76] Jennifer Tam, Jirí Simsa, Sean Hyde, et al. Breaking Audio CAPTCHAs[C]. *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems (NIPS)*, 2008: 1625-1632.
[77] Sivakorn, S., Polakis, J. and Keromytis, A.D. I’m not a human: Breaking the Google reCAPTCHA[J]. *Black Hat*, 2016: 14.
[78] Fatmah H. Alqahtani, Fawaz A. Alsulaiman. Is image-based CAPTCHA secure against attacks based on machine learning? An experimental study[J]. *Computers & Security*, 2020, 88.
[79] Stark, F., Hazırbas, C., Triebel, R., & Cremers, D. Captcha recognition with active deep learning[J]. *New Challenges in Neural Computation*, 2015, 94.
[80] Du, F. L., Li, J. X., Yang, Z., et al. CAPTCHA recognition based on faster R-CNN[C]. *International Conference on Intelligent Computing*, Springer, 2017: 597-605.
[81] Matthew D Zeiler, Rob Fergus. Visualizing and Understanding Convolutional Networks[EB/OL]. 2013: *ArXiv Preprint ArXiv*: 1311.2901.
[82] Lin, D., Lin, F., Lv, Y., et al. Chinese character CAPTCHA recognition and performance estimation via deep neural network[J]. *Neurocomputing*, 2018, 288: 11-19.
[83] Tang, M., Gao, H., Zhang, Y., et al. Research on Deep Learning Techniques in Breaking Text-Based Captchas and Designing Image-Based Captcha[J]. *IEEE Transactions on Information Forensics and Security*, 2018, 13(10): 2522-2537.
[84] Wu, X., Dai, S., Guo, Y., et al. A machine learning attack against variable-length Chinese character CAPTCHAs[J]. *Applied Intelligence*, 2019, 49(4): 1548-1565.
[85] Y. Zi, H. Gao, Z. Cheng, et al. An End-to-End Attack on Text CAPTCHAs[J]. *IEEE Transactions on Information Forensics and Security*, 2020, 15: 753-766. DOI: 10.1109/TIFS.2019.2928622.
[86] Tian S., and Tao Xiong. A Generic Solver Combining Unsupervised Learning and Representation Learning for Breaking Text-Based Captchas[C]. *Proceedings of The Web Conference 2020*, 2020: 860-871.
[87] Sukhani, K., Sawant, S., Maniar, S., et al. Automating the Bypass of Image-based CAPTCHA and Assessing Security[C]. *12th International Conference on Computing Communication and Networking Technologies*. IEEE, 2021: 1-8.
[88] Isao Takaesu. Automatic Generation of Injection Codes using Genetic Algorithm[EB/OL]. https://www.mbsd.jp/blog/20170921.html. 2017.
[89] XunSu, KeYunLuo. Deep X-Ray: 一种机器学习驱动的 WAF 规则窃取器[EB/OL]. http://t.cn/A65ZGOyL. 2020.
[90] Demetrio, L., Valenza, A., Costa, G., et al. WAF-A-MoLE: Evading Web Application Firewalls Through Adversarial Machine Learning[C]. *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, ACM, 2020: 1745-1752.
[91] Zhenqing Qu, Xiang Ling, and Chunming Wu. AutoSpear: Towards Automatically Bypassing and Inspecting Web Application Firewalls[J]. *Black Hat Asia*, 2022.
[92] Fu Zhangjie, Li Enlu, Cheng Xu, et al. Recent Advances in Image Steganography Based on Deep Learning[J]. *Chinese Journal of Computer Research and Development*, 2021, 58(3): 548-568.
(付章杰, 李恩露, 程旭, 黄永峰, 胡雨婷. 基于深度学习的图像隐写研究进展[J]. 计算机研究与发展, 2021, 58(3): 548-568.)
[93] Volkhonskiy D., Nazarov I., Borisenko B., et al. Steganographic Generative Adversarial Networks[EB/OL], 2017: *arXiv preprint arXiv*:1703.05502.
[94] Zhou Zhili, Cao Yi, Sun Xingming. Coverless Information Hiding Based on Bag-of-Words Model of Image[J]. *Chinese Journal of Applied Sciences*, 2016, 34(5): 527-536.
(周志立, 曹談, 孙星明. 基于图像 Bag-of-Words 模型的无载体信息隐藏[J]. 应用科学学报, 2016, 34(5): 527-536.)
[95] Zhu J., Kaplan R., Johnson J, et al. HiDDeN: Hiding Data with Deep Networks[C]. *Proceedings of the European conference on computer vision (ECCV)*, 2021: 657-672.
[96] Zhu Dingju. Hiding Information in Big Data based on Deep Learning[EB/OL]. 2019: *arXiv preprint arXiv*:1912.13156.
[97] Liu T., Liu Z., Liu Q., et al. StegoNet: Turn Deep Neural Network into a Stegomalware[C]. *Annual Computer Security Applications Conference (ACSAC)*, ACM, 2020: 928-938.
[98] Wang Z., Liu C., Cui X. EvilModel: Hiding Malware Inside of Neural Network Models[C]. *2021 IEEE Symposium on Computers and Communications (ISCC)*, 2021: 1-7.
[99] Wang Z., Liu C., Cui X., et al. EvilModel 2.0: Bringing Neural Network Models into Malware Attacks[EB/OL]. 2021: *arXiv preprint arXiv:2109.04344*.
[100] Li Z., Zou D., Xu S., et al. VulDeePecker: A Deep Learning-Based System for Vulnerability Detection[EB/OL]. 2018: *arXiv preprint arXiv:1801.01681*.
[101] She D., Pei K., Epstein D., et al. Neuzz: Efficient Fuzzing with Neural Program Smoothing[C]. *2019 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2019: 803-817.
[102] Peiyuan Zong, Tao Lv, Dawei Wang, et al. FuzzGuard: Filtering out Unreachable Inputs in Directed Grey-box Fuzzing through Deep Learning[C]. In *29th USENIX Security Symposium*, Usenix, 2020: 2255-2269.
[103] Zhou Y., Liu S., Siow J., et al. Devign: Effective Vulnerability Identification by Learning Comprehensive Program Semantic

s via Graph Neural Networks[C]. *Advances in Neural Information Processing Systems*, 2019, 32.
[104]Isao Takaesu, Takeshi Terada. SAIVS (Spider Artificial Intelligence Vulnerability Scanner)[J]. *Black Hat Asia 2016 Arsenal*, 2016.
[105]Bishop Fox. DeepHack. https://github.com/BishopFox/deephack. 2017.
[106]Masafumi Masuya, Isao Takaesu, Toshitsugu Yoneyama, et al. GyoiThon: Next generation penetration test tool[OL]. https://github.com/gyoisamurai/GyoiThon. 2018.
[107]McAfee. 利用高级恶意软件检测技术增强威胁检测功能[OL]. https://www.mcafee.com/enterprise/zh-cn/products/security-analytics-products.html.
[108]Fortinet. FortiAI: Virtual Security Analyst[OL]. https://www.fortinet.com/cn/products/fortiai.
[109]奇安信. 安全 DNS 防御系统[OL]. https://www.qianxin.com/product/detail/pid/424.
[110]深信服. 深信服终端检测响应平台 EDR[OL]. https://edr.sangfor.com.cn/#/introduction/edr.
[111]Norton. 防病毒和恶意软件防护[OL]. https://cn.norton.com/antivirus.
[112]Kaspersky. 卡巴斯基反欺诈平台[OL]. https://www.kaspersky.com.cn/enterprise-security/fraud-prevention.
[113]瑞星. 瑞星 AI 网络威胁检测引擎[OL]. http://www.rising.com.cn/avsdk/.
[114]华为. HiSec Insight 安全态势感知系统[OL]. https://e.huawei.com/cn/products/enterprise-networking/security/bigdata-apt/cis.
[115]H3C. H3C SecPath F1000-AK 系列 AI 防火墙[OL]. http://www.h3c.com/cn/Products___Technology/Products/IP_Security/FW_VPN/F1000/F1000-AK1000/.
[116]观成科技. 瞰云-加密威胁智能检测系统（ENS）[OL]. https://www.viewintech.com/html/product.html.
[117]Weilin Xu, Yanjun Qi, David Evans. Automatically evading classifiers[C]. *Proceedings of the 2016 Network and Distributed Systems Symposium*, 2016, 10.
[118]Dang H., Huang Y., Chang E.C. Evading Classifiers by Morphing in the Dark[C]. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ACM, 2017: 119-133.
[119]Dey, S., Kumar, A., Sawarkar, M., et al. EvadePDF: Towards Evading Machine Learning Based PDF Malware Classifiers[C]. *International Conference on Security & Privacy*, Springer, 2019: 140-150.
[120]Li, Y., Wang, Y., Wang, Y., et al. A feature-vector generative adversarial network for evading PDF malware classifiers[J]. *Information Sciences*, 2020, 523: 38-48.
[121]Bae H., Lee Y., Kim Y., et al. Learn2Evade: Learning-Based Generative Model for Evading PDF Malware Classifiers[J]. *IEEE Transactions on Artificial Intelligence*, 2021, 2(4): 299-313.
[122]Hu W, Tan Y. Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN[EB/OL]. 2017: *arXiv preprint arXiv: 1702.05983*.
[123]Anderson H S, Kharkar A, Filar B, et al. Learning to Evade Static PE Machine Learning Malware Models via Reinforcement Learning[EB/OL]. 2018: *arXiv preprint arXiv: 1801.08917*.
[124]Hu W., Tan Y. Black-box Attacks Against RNN Based Malware Detection Algorithms[C]. *Workshops at the 32nd AAAI Conference on Artificial Intelligence*, 2018.
[125]Luca D., Biggio B., Giovanni L., et al. Explaining Vulnerabilities of Deep Learning to Adversarial Malware Binaries[C]. *3rd Italian Conference on Cyber Security (ITASEC)*, 2019, 2315.
[126]Yuan J, Zhou S, Lin L, et al. Black-box Adversarial Attacks Against Deep Learning Based Malware Binaries Detection with GAN[C]. *24th European Conference on Artificial Intelligence*, IOS Press, 2020: 2536-2542.
[127]Amich A., Eshete B. Explanation-Guided Diagnosis of Machine Learning Evasion Attacks[C]. *International Conference on Security and Privacy in Communication Systems (SecureCom)*, Springer, 2021: 207-228.
[128]Chen B., Ren Z., Yu C., et al. Adversarial Examples for CNN-Based Malware Detectors[J]. *IEEE Access*, 2019, 7: 54360-54371.
[129]Anderson H.S., Kharkar A., Filar B., et al. Evading machine learning malware detection[J]. *Black Hat*, 2017.
[130]Li X. and Li Q. An IRL-Based Malware Adversarial Generation Method to Evade Anti-Malware Engines[J]. *Computers & Security*, 2021, 104: 102118.
[131]Ebrahimi M., Pacheco J., Li W., et al. Binary Black-Box Attacks Against Static Malware Detectors with Reinforcement Learning in Discrete Action Spaces[C]. *2021 IEEE Security and Privacy Workshops (SPW)*, IEEE, 2021: 85-91.
[132]Grosse K., Papernot N., Manoharan P., et al. Adversarial Perturbations Against Deep Neural Networks for Malware Classification[EB/OL]. 2016: *arXiv preprint arXiv:1606.04435*.
[133]Xu P., Kolosnjaji B., Eckert C., et al. MANIS: Evading Malware Detection System on Graph Structure[C]. *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, ACM, 2020: 1688-1695.
[134]Li D., Li Q. Adversarial Deep Ensemble: Evasion Attacks and Defenses for Malware Detection[J]. *IEEE Transactions on Information Forensics and Security*, 2020, 15: 3886-3900.
[135]Li C., Chen X., Wang D., et al. Backdoor Attack on Machine Learning Based Android Malware Detectors[J]. *IEEE Transactions on Dependable and Secure Computing*, IEEE, 2021.
[136]Grosse K., Papernot N., Manoharan P., et al. Adversarial Examples for Malware Detection[C]. *European Symposium on Research in Computer Security (ESORICS)*, Springer, 2017: 62-79.
[137]Chen X., Li C., Wang, D., et al. Android HIV: A Study of Repackaging Malware for Evading Machine-Learning Detection[J]. *IEEE Transactions on Information Forensics and Security*, 2019, 15: 987-1001.
[138]Kaifa Zhao, Hao Zhou, Yulin Zhu, et al. Structural Attack against Graph Based Android Malware Detection[C]. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, ACM, 2021: 3218-3235.
[139]Abusnaina A., Khormali A., Alasmary H., et al. Adversarial Learning Attacks on Graph-Based IoT Malware Detection Systems[C]. *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, IEEE, 2019: 1296-1305.
[140]Khasawneh K.N., Abu-Ghazaleh N., Ponomarev D., et al. RHMD: Evasion-Resilient Hardware Malware Detectors[C]. *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*, 2017: 315-327.
[141]Nozawa K., Hasegawa K., Hidano S., et al. Generating Adversarial Examples for Hardware-Trojan Detection at Gate-Level Netlists[J]. *Journal of Information Processing*, 2021, 29: 236-246.
[142]Rigaki M. and Garcia S. Bringing a GAN to a Knife-fight: Adapting Malware Communication to Avoid Detection[C]. *2018 IEEE Security and Privacy Workshops (SPW)*, IEEE, 2018: 70-75.
[143]Apruzzese G., Colajanni M. Evading Botnet Detectors Based on Flows and Random Forest with Adversarial Samples[C]. *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*, IEEE, 2018: 1-8.
[144]Wu D., Fang B., Wang J., et al. Evading Machine Learning Botnet Detection Models via Deep Reinforcement Learning[C]. *2019 IEEE International Conference on Communications (ICC)*, IEEE, 2019: 1-6.
[145]Yongjin Hu, Yuanbo Guo, Jun Ma, et al. Method to Generate Cyber Deception Traffic Based on Adversarial Sample[J]. *Chinese Journal on Communications*, 2020, 41(9): 59-70.
(胡永进, 郭渊博, 马骏, 等. 基于对抗样本的网络欺骗流量生成方法[J]. 通信学报, 2020, 41(9): 59-70.)
[146]Wang J., Liu Q., Wu D., et al. Crafting Adversarial Example to Bypass Flow-&ML-based Botnet Detector via RL[C]. *24th International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*, 2021: 193-204.
[147]Lin Z., Shi Y., Xue, Z. IDSGAN: Generative Adversarial Networks for Attack Generation Against Intrusion Detection[EB/OL]. 2018: *arXiv preprint arXiv:1809.02077*.

[148]Novo C., Morla R. Flow-Based Detection and Proxy-Based Evasion of Encrypted Malware C2 Traffic[C]. *Proceedings of the 13th ACM Workshop on Artificial Intelligence and Security (AISec)*, ACM, 2020: 83-91.

[149]Han Y., Fang B., Cui X., et al. StealthyFlow: A Framework for Malware Dynamic Traffic Camouflaging in Adversarial Environment[J]. *Chinese Journal of Computers*, 2021, 44(5): 948-962.
(韩宇, 方滨兴, 崔翔, 等. StealthyFlow: 一种对抗条件下恶意代码动态流量伪装框架[J]. 计算机学报, 2021, 44(5): 948-962.)

[150]Plohmann D., Yakdan K., Klatt M., et al. A Comprehensive Measurement Study of Domain Generating Malware[C]. *25th USENIX Security Symposium*, USENIX, 2016: 263-278.

[151]Yadav S., Reddy A. K. K., Reddy A. L. N., et al. Detecting Algorithmically Generated Domain-Flux Attacks With DNS Traffic Analysis[J]. *IEEE/ACM Trans. Netw*, 2012, 20(5): 1663-1677.

[152]Antonakakis M., Perdisci R., Nadji Y., et al. From Throw-Away Traffic to Bots: Detecting the Rise of DGA-Based Malware[C]. *21th USENIX Security Symposium*, USENIX, 2012: 491-506.

[153]Schiavoni S., Maggi F., Cavallaro L., et al. Phoenix: DGA-Based Botnet Tracking and Intelligence[C]. *11th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*, Springer, 2014: 192-211.

[154]Anderson H. S., Woodbridge J., Filar B. DeepDGA: Adversarially-Tuned Domain Generation and Detection[C]. *Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security (AISec@CCS 2016)*, ACM, 2016: 13-21.

[155]Corley I., Lwowski J., Hoffman J. DomainGAN: Generating Adversarial Examples to Attack Domain Generation Algorithm Classifiers[EB/OL]. 2019: *arXiv preprint arXiv:1911.06285*.

[156]Sidi L., Nadler A., Shabtai A. MaskDGA: An Evasion Attack Against DGA Classifiers and Adversarial Defenses[J]. *IEEE Access*, 2020, 8: 161580-161592.

[157]Peck J., Nie C., Sivaguru R., et al. CharBot: A Simple and Effective Method for Evading DGA Classifiers[J]. *IEEE Access*, 2019, 7: 91759-91771.

[158]Zheng Y., Yang C., Yang Y., et al. ShadowDGA: Toward Evading DGA Detectors with GANs. *2021 International Conference on Computer Communications and Networks (ICCCN)*, IEEE, 2021: 1-8.

[159]Alfonso Muñoz. uriDeep[OL]. https://github.com/mindcrypt/uriDeep. 2019.

[160]YSYY. Punycode[EB/OL]. http://www.arkteam.net/?p=3049. 2017.

[161]FireEye. Uncovering a Malware Backdoor that Uses Twitter[OL]. https://www.fireeye.com/current-threats/apt-groups/rpt-apt29.html. 2015.

[162]F-Secure. The Dukes: 7 Years of Russian Cyberespionage[EB/OL]. https://blog-assets.f-secure.com/wp-content/uploads/2020/03/18122307/F-Secure_Dukes_Whitepaper.pdf. 2015.

[163]Matthieu Faou. From Agent.Btz to Comrat V4: A Ten-Year Journey. Technical Report, ESET, 2020.

[164]Kirat D. H., Jang J., Stoecklin M. P. AI-Powered Cyber Data Concealment and Targeted Mission Execution[P]. US Patent: US20200007512A1, 2020-01-02.

[165]Ji T., Fang B., Cui X., et al. The First Step Towards Modeling Unbreakable Malware[EB/OL]. 2020: *arXiv preprint arXiv:2008.06163*.

[166]Yu N., Tuttle Z., Thurnau C.J., et al. AI-Powered GUI Attack and Its Defensive Methods[C]. *Proceedings of the 2020 ACM Southeast Conference*, ACM, 2020: 79-86.

[167]MinJae Kwon. Flog[OL]. https://github.com/mingrammer/flog. 2018.

[168]Kirit Basu. Fake Apache Log Generator[OL]. https://github.com/kiritbasu/Fake-Apache-Log-Generator. 2015.

[169]Toemmel C. Catch Me If You GAN: Using Artificial Intelligence for Fake Log Generation. 2021: *arXiv preprint arXiv:2112.12006*.

[170]Beni G., Wang J. Swarm Intelligence in Cellular Robotic Systems[C]. *Robots and Biological Systems: Towards a New Bionics?. NATO ASI Series (Series F: Computer and Systems Sciences)*, 1993, 102.

[171]Zeng P., Hua L., Chen J., et al. Research Progress of DARPA UAV Swarm Technology in the US[J]. *Military Digest*, 2020, (05): 23-27.
(曾鹏, 花梁修宇, 陈军燕, 等. 美国 DARPA 无人机集群技术研究进展[J]. 军事文摘, 2020, (05): 23-27.)

[172]Bangchu Zhang, Jian Liao, Yu Kuang, et al. Research Status and Development Trend of the United States UAV Swarm Battlefield[J]. *Aero Weaponry*, 2020, 27(6): 7-12.
(张邦楚, 廖剑, 匡宇, 等. 美国无人机集群作战的研究现状与发展趋势[J]. 航空兵器, 2020, 27(6): 7-12.)

[173]Zelinka I., Das S., Sikora L., et al. Swarm Virus-Next-Generation Virus and Antivirus Paradigm?[J]. *Swarm and Evolutionary Computation*, 2018, 43: 207-224.

[174]XJ. “进击”的僵尸网络[OL]. http://www.arkteam.net/?p=3468. 2018.

[175]Castiglione A., De Prisco R., De Santis A., et al. A Botnet-Based Command and Control Approach Relying on Swarm Intelligence[J]. *Journal of Network and Computer Applications*, 2014, 38: 22-33.

[176]Cani A., Gaudesi M., Sanchez E., et al. Towards Automated Malware Creation: Code Generation and Code Integration[C]. *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, ACM, 2014: 157-160.

[177]Danziger M., Henriques M.A.A. Attacking and Defending with Intelligent Botnets[C]. *XXXV Simpósio Brasileiro de Telecomunicaç oes e Processamento de Sinais-SBrT*, 2017: 457-461.

[178]Truong T.C., Zelinka I., Senkerik, R. Neural Swarm Virus[C]. *Swarm, Evolutionary, and Memetic Computing and Fuzzy and Neural Computing*, Springer, 2019: 122-134.

[179]Paar, Christof, and Jan Pelzl. Understanding Cryptography: A Textbook for Students and Practitioners[M]. Springer Science & Business Media, 2009.

[180]Antesar M. Shabut, Khin T. Lwin, and M. Alamgir Hossain. Cyber Attacks, Countermeasures, and Protection Scheme: A State of the Art Survey[C]. In *2016 10th International Conference on Software, Knowledge, Information Management & Applications (SKIMA)*, IEEE, 2016: 37-44.

[181]J. Rosenberg, Chapter e6-Embedded security, Rugged Embedded Systems[M]. Morgan Kaufmann, 2017: e1-74.

[182]Joshua Reynolds, Nikita Samarin, Joseph Barnes, et al. Empirical Measurement of Systemic 2FA Usability[C]. In *29th USENIX Security Symposium*, Usenix, 2020: 127-143.

[183]Michael Howard, and Steve Lipner. The Security Development Lifecycle. Vol. 8[M]. Redmond: Microsoft Press, 2006.

[184]Samuel Schüppen, Dominik Teubert, Patrick Herrmann, et al. FANCI: Feature-based Automated NXDomain Classification and Intelligence[C]. In *27th USENIX Security Symposium*, Usenix, 2018:1165-1181.

[185]Microsoft. Audit Sensitive Privilege Use[OL]. https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/audit-sensitive-privilege-use.

[186]Alavizadeh H., Jang-Jaccard J., Alpcan T., et al. A Markov Game Model for AI-based Cyber Security Attack Mitigation[EB/OL]. 2021: *arXiv preprint arXiv:2107.09258*.

[187]Alavizadeh, H., Jang-Jaccard, J., Alpcan, T. et al. A Game-Theoretic Approach for AI-based Botnet Attack Defence[EB/OL]. 2021: *arXiv preprint arXiv:2112.02223*.

[188]Li, D., Li, Q., Ye, Y., et al. Arms Race in Adversarial Malware Detection: A Survey[J]. *ACM Computing Surveys (CSUR)*, 2021, 55(1): 1-35.

[189]Zhang, F., Chan, P.P., Biggio, B., et al. Adversarial Feature Selection Against Evasion Attacks[J]. *IEEE Transactions on Cybernetics*, 2016, 46(3): 766-777.

[190]Fredrikson M., Jha S., Ristenpart, T. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures[C]. *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, ACM, 2015: 1322-1333.

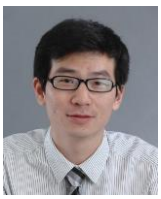
[191]Shan, S., Wenger, E., Wang, B., et al. Gotta Catch'em All: Using Honeypots to Catch Adversarial Attacks on Neural Networks[C]. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 2020: 67-83.
[192]Chen, L., Hou, S., Ye, Y., et al. Droideye: Fortifying Security of Learning-Based Classifier Against Adversarial Android Malware Attacks[C]. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE. 2018: 782-789.
[193]Tencent AI Lab. AI 安全威胁矩阵 AI Sec Matrix[OL]. https://matrix.tencent.com/.
[194]Shokri R., Stronati M., Shmatikov, V. Membership Inference Attacks against Machine Learning Models[EB/OL]. 2016: *arXiv preprint arXiv:1610.05820*.
[195]Ligeng Zhu, Zhijian Liu, Song Han. Deep Leakage from Gradients[C]. *Annual Conference on Neural Information Processing Systems 2019 (NeurIPS 2019)*, 2019: 14747-14756.
[196]Brendan McMahan, Eider Moore, Daniel Ramage, et al. Communication-Efficient Learning of Deep Networks from Decentralized Data[C]. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017: 1273-1282.
[197]Wang Z. Analysis and Classification on AI-based Attacks[EB/OL]. 2022: *ChinaXiv preprint ChinaXiv:202201.00085*.

王志 于 2017 年在中国民航大学信息专业获得工学学士学位，现在中国科学院大学网络空间安全专业攻读博士学位，研究领域为网络攻防技术，研究兴趣包括人工智能安全、僵尸网络。Email：wangzhi@iie.ac.cn

崔翔 于 2012 年在中国科学院研究生院信息安全专业获得工学博士学位，现任广州大学网络空间先进技术研究院教授，研究领域为网络空间安全，主要研究方向为网络安全。Email：cuixiang@gzhu.edu.cn

刘潮歌 于 2019 年在中国科学院大学信息安全安全专业获得工学博士学位，现任中国科学院信息工程研究所副研究员，研究兴趣包括恶意代码、网络攻击追踪溯源和 Web 安全。Email: liuchaoge@iie.ac.cn

尹捷 于 2021 年在中国科学院大学网络空间安全专业获得工学博士学位，现任中国科学院信息工程研究所工程师，研究领域为网络攻防技术、僵尸网络，研究兴趣包括人工智能安全。Email：yinjie@iie.ac.cn



刘奇旭 于 2011 年在中国科学院研究生院信息安全专业获得博士学位，现任中国科学院信息工程研究所研究员、中国科学院大学网络空间安全学院教授，主要研究方向为网络攻防技术、网络安全评测。Email: liuqixu@iie.ac.cn

汪旭童 于 2020 年在中国矿业大学信息安全专业获得学士学位，现在中国科学院大学网络空间安全专业攻读硕士学位，研究领域为 Web 安全、网络溯源、机器学习。Email：wangxutong@iie.ac.cn