

## 基于诱捕的软件异常检测综述

傅建明<sup>1,2</sup>, 刘畅<sup>1,2</sup>, 解梦飞<sup>1,2</sup>, 罗陈可<sup>1,2</sup>

(1. 空天信息安全与可信计算教育部重点实验室, 湖北 武汉 430072;

2. 武汉大学国家网络安全学院, 湖北 武汉 430072)

**摘要:** 高级持续威胁 (APT, advanced persistent threats) 会使用漏洞实现攻击代码的自动加载和攻击行为的隐藏, 并通过复用代码攻击绕过堆栈的不可执行限制, 这是网络安全的重要威胁。传统的控制流完整性和地址随机化技术虽然有效抑制了 APT 的步伐, 但软件的复杂性和攻击演化使软件仍存在被攻击的时间窗口。为此, 以资源为诱饵的诱捕防御是确保网络安全的必要补充。诱捕机制包含诱饵设计和攻击检测两部分, 通过感知与诱饵的交互行为, 推断可能的未授权访问或者恶意攻击。针对文件、数据、代码 3 种诱饵类型, 设计诱饵的自动构造方案并进行部署, 从真实性、可检测性、诱惑性等方面对诱饵的有效程度进行度量。基于诱捕防御的勒索软件检测注重诱饵文件的部署位置, 在漏洞检测领域, 通过注入诱饵代码来检测代码复用攻击。介绍了在 APT 攻击各个阶段实施诱捕防御的相关研究工作, 从诱饵类型、诱饵生成、诱饵部署、诱饵度量方面刻画了诱捕防御的机理; 同时, 剖析了诱捕防御在勒索软件检测、漏洞检测、Web 安全方面的应用。针对现有的勒索软件检测研究在诱饵文件设计与部署方面的不足, 提出了用于检测勒索软件的诱饵动态更新方法。讨论了诱捕防御面临的挑战, 希望诱捕防御可以为发现未知攻击、溯源攻击意图提供理论和技术支持。

**关键词:** 高级持续威胁; 代码复用攻击; 控制流完整性; 地址随机化; 诱捕防御

**中图分类号:** TP393

**文献标志码:** A

**DOI:** 10.11959/j.issn.2096-109x.2022003

## Survey of software anomaly detection based on deception

FU Jianming<sup>1,2</sup>, LIU Chang<sup>1,2</sup>, XIE Mengfei<sup>1,2</sup>, LUO Chenke<sup>1,2</sup>

1. Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, Wuhan 430072, China

2. School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China

**Abstract:** Advanced persistent threats (APT) will use vulnerabilities to automatically load attack code and hide attack behavior, and exploits code reuse to bypass the non-executable stack & heap protection, which is an essential threat to network security. Traditional control flow integrity and address space randomization technologies have effectively prevented the pace of APT. However, the complexity of the software and the evolution of attacks make the software still being vulnerable. For this reason, deception defense with resources as bait is an indispensable supple-

收稿日期: 2021-01-06; 修回日期: 2021-03-23

通信作者: 傅建明, jmfu@whu.edu.cn

基金项目: 国家自然科学基金 (61972297, 62172308, 62172144)

**Foundation Item:** The National Natural Science Foundation of China (61972297, 62172308, 62172144)

**引用格式:** 傅建明, 刘畅, 解梦飞. 等. 基于诱捕的软件异常检测综述[J]. 网络与信息安全学报, 2022, 8(1): 15-29.

**Citation Format:** FU J M, LIU C, JIE M F, et al. Survey of software anomaly detection based on deception[J]. Chinese Journal of Network and Information Security, 2022, 8(1): 15-29.

ment for network security. The trapping mechanism consists of bait design and attack detection, which infer possible unauthorized access or malicious attacks by sensing the interaction behavior with the bait. According to the three types of bait, which are file, data and code, the automatic construction scheme of bait is designed and deployed, and the effectiveness of bait is measured from the aspects of believability, detectability and enticement, etc. Ransom ware detection based on deception defense focuses on the deployment location of bait files, and in the area of vulnerability detection, code reuse attacks are detected by injecting bait code. Research work related to the implementation of deception defense in each phase of APT attacks was introduced, and the mechanism of deception defense from bait type, bait generation, bait deployment, and bait measurement was described. Simultaneously, deception defense applications in ransom ware detection, vulnerability detection, and Web security were analyzed. In response to the shortcomings of existing ransom ware detection research in terms of bait file design and deployment, a dynamic update method of bait for ransom ware detection was proposed. The deception defense challenges were discussed and hoped that deception defense can provide theoretical and technical support for discovering unknown attacks and attack attribution.

**Keywords:** advanced persistent threat, code reuse attack, control flow integrity, address randomization, deception defense

## 0 引言

互联网和物联网的发展,以及智能设备的广泛应用,提高了人们的生活质量和工作效率,使人类的工作和生活逐渐数字化、自动化和智能化。随着时间的推移,网络世界的信息资产和财富不断累积。这些资产和财富吸引了大量的攻击者或者攻击集团,网络安全事件不断涌现。高级持续威胁(APT, advanced persistent threat)<sup>[1]</sup>指的是拥有丰富专业知识和大量资源的攻击者,能够使用多种攻击载体来实现攻击目标。APT攻击一般会使用漏洞(如Windows漏洞CVE-2018-8611、Flash漏洞CVE-2018-15982)实现攻击代码的自动加载和攻击行为的隐藏。而且,网络攻击从过去的隐蔽潜行逐渐走向公开对抗<sup>[2]</sup>。例如,Wannacry勒索病毒使用永恒之蓝漏洞(CVE-2017-0143)实现了病毒自动化的网络传播。防病毒工具<sup>[3]</sup>、入侵检测系统、防火墙都是软件外部的安全机制,它们利用签名或者启发式知识检测已知威胁,如木马、计算机病毒、不合规的流量等。这些预定义的策略和知识独立于软件的内部逻辑,难以检测零日漏洞攻击和新的恶意软件。

蜜罐(HoneyPot)和蜜网(HoneyNet)是网络安全的一种主动防御机制<sup>[4-7]</sup>。该防御机制通过部署虚假的目标(Honey-token)和模糊化真实目

标,干扰攻击者对目标和防御者的认知,引诱攻击者使用其部署的诱饵服务和诱饵网络,动态捕获其攻击行为和攻击工具,解析其攻击方法,推测攻击者的攻击意图和攻击动机。网络诱捕<sup>[4]</sup>利用主机、服务和信息作为诱饵,检测网络入侵、恶意代码、僵尸网络和蠕虫传播、垃圾邮件传播,并提取网络攻击特征,为入侵检测系统和防火墙提供威胁情报。目前,有许多企业研制出了基于诱骗的主动防御产品,如Rapid7(美国)、LogRhythm(美国)、Attivo Networks(以色列)、Cymmetria(以色列)、Smokescreen Technologies(印度)、北京元支点(中国)、长亭科技(中国)、默安科技(中国)、锦行网络科技(中国)等。他们利用软件定义网络或虚拟机实现终端、网络、应用、工控等蜜罐产品,这些产品可以从系统或网络的外部异常行为上感知攻击过程,但缺少与主机软件和主机数据关联的诱捕机制。

蜜罐和蜜网作为一个独立于业务系统的诱捕环境,其目的是诱使攻击者攻击暴露的主机和服。把蜜罐思想引入软件安全,在主机环境部署诱饵文件。一旦软件访问这些诱饵文件,就触发软件异常报警。此时,诱捕防御是一种基本的安全能力,与已有的防病毒、防火墙、入侵检测系统互为补充。诱捕防御可以模糊化防守方的代码资源和数据资源,增强防守方的情报能力和防御筹码,感知未知的安全威胁。

## 1 背景与相关工作

APT 攻击的基本流程包括目标探测、攻击代码传输、攻击代码触发、攻击代码执行和敏感资源获取<sup>[8]</sup>, 如图 1 所示。这里省略了逃逸、提权、边界移动等过程, 逃逸是为了对抗主机安全检测和网络安全检测, 提权是为了获得更多的敏感资源, 边界移动的目的是探测和获取外网不可见的资源, 深度挖掘目标的高价值资源。

目标探测可分为远程探测 (remote probing) 和本地探测 (local probing) 两种类型。远程探测可以通过手动或自动化工具进行, 如直接猜测目标的登录凭证 (user name & password)、扫描系统的版本或脆弱性<sup>[8]</sup>, 扫描或探测数据包中的特有签名、探测频率或行为模式都可以作为检测目标探测攻击的依据<sup>[9]</sup>。一旦攻击者熟悉这些依据, 就可以绕过这类检测。针对远程探测, 可以部署一些虚假的登录凭证、人为植入的漏洞。当这些凭证或漏洞被利用时, 则可以提前感知攻击行为<sup>[10-11]</sup>。

攻击者借助远程探测的结果, 如弱口令或系统漏洞, 可以初步远程访问目标系统, 但攻击者在目标系统中执行的代码是目标系统定制的, 这种定制受限于目标软件的内部逻辑, 无法满足攻击者获取敏感资源的需求。为此, 攻击者需要远程注入自己精心构造的代码到目标系统中, 然后执行注入代码, 以达到攻击的目的。

远程注入的代码分为 ROP (returned-oriented programming)、Shellcode、可执行代码模块 (如 PE、ELF) 3 类。ROP 复用软件中的已有代码, 构建攻击者的攻击载荷<sup>[12-13]</sup>, 以绕过栈和堆中代码不可执行的限制 (DEP, data execution prevention)。ROP 可以让 Shellcode 获得可执行权限。Shellcode 由可执行的代码片段构成, 其作用是加载可执行代码模块<sup>[14]</sup>。可执行代码模块是攻击者完成攻击任务的核心, 用来探测软件环境、

加载任务模块, 并最终从磁盘和内存中获取目标的文档资源、网络资源或者各种登录凭证等。大多数 ROP 防御聚焦于控制流完整性 (CFI, control flow integrity) 保护<sup>[15]</sup>和地址空间布局随机化防御 (ASLR, address space layout randomization)<sup>[16-17]</sup>, 前者阻止软件间接代码指针的非预期跳转, 后者破坏 ROP 代码地址不变的假设。针对 ROP 的构造, 可以在软件中部署一些诱饵代码<sup>[18]</sup>。一旦这些诱饵代码被利用, 则可以检测未知的攻击。

计算机病毒、勒索软件、木马等都属于模块级的攻击代码<sup>[19]</sup>, 这些攻击代码探测磁盘中的敏感文件、内存中的敏感数据, 这种探测属于本地探测。针对本地探测, 可以部署一些诱饵文件, 一旦这些诱饵文件被访问, 就可以检测未知的攻击<sup>[20]</sup>。

在控制了内网机器后, 攻击者往往会进行横向移动, 如读取其他系统用户的数据、扫描当前网段内的其他主机, 以便控制包含高价值目标的主机。另外, 攻击者会尝试攻击域服务器, 从而获取域管理员权限。对于域控攻击, 同样可以通过布置诱饵进行检测。例如, 对于 Kerberoast 攻击, 攻击者会查询高权限域用户下的服务主体名称 (SPN), 导出并尝试破解服务票据。因此, 防守方可以创建一个诱饵 Kerberoast 服务账户, 并伪造 SPN, 检测对于该诱饵服务票据的请求行为。

为了隐藏攻击行为, 攻击者会使用加密通信、匿名代理等方式来传输数据, 从而加大溯源的难度。另外, 在完成攻击任务后, 攻击者会清理目标主机和网络中的相关记录, 如删除系统日志和文件、修改注册表等。

APT 攻击流程的每个阶段都可以部署诱饵资源, 实施诱捕防御。多级防御或纵深防御同样可以应用于诱捕防御。文献[21]提出了由网络服务仿真、安全漏洞伪造、操作行为控制和文件系统镜像组成的深度欺骗策略, 以突破单层欺骗防御

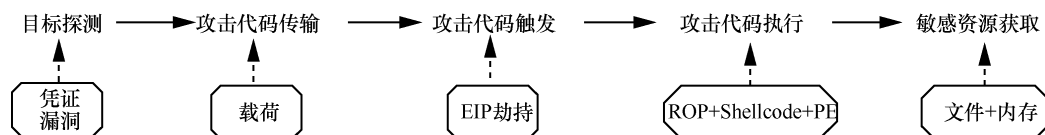


图1 APT 攻击的基本流程  
Figure 1 The basic flow of APT attacks

的局限性。文献[22]针对 APT 侦察、破袭主机、破袭服务器等多阶段攻击过程,设计用户层、本地层、全局层的诱饵资源(诱饵用户、诱饵文件和诱饵服务器),构建了一种纵深的诱捕防御体系。

文献[7]认为网络欺骗利用骗局干扰或误导攻击者的认知,以抑制攻击活动。同时,描述了网络欺骗的生命周期、工作流程,剖析了设备层、网络层、数据层、应用层上的欺骗技术。该文献从认知层面介绍诱骗防御,讨论了网络层面和系统层面诱骗防御存在的挑战。

文献[23]从社会工程学角度定义信息欺骗。信息欺骗利用用户兴趣偏好、利益互惠、摆脱恐惧等方面的弱点,误导用户操作,获取敏感信息、下载和执行恶意代码等。同时,该文献详细分析了社交欺骗、网站欺骗、应用欺骗、邮件欺骗等欺骗场景。

本文聚焦软件或代码访问的资源,从诱饵类型、诱饵生成、诱饵部署、诱饵度量等方面刻画了诱捕防御的机理;同时,剖析了诱捕防御在勒索软件检测和漏洞检测等方面的应用。与文献[7]相比,本文属于应用层和数据层交叉的欺骗,从软件安全角度介绍诱骗防御机制。文献[23]以攻击者的信息欺骗为主,分析应对的防御措施,与之对比,本文采用信息欺骗的思想以确保软件安全,其视角和内容都不同。

## 2 诱捕机制

文献[7]定义了网络欺骗的 5 个要素 {Defender, Asset, Trick, Attacker, Profit}, 其中 Trick 表示骗局,是 5 个要素的核心。骗局有两种策略:以真似假和以假拟真。

**以真似假:**改变已有资源(Asset)的特征,或者伪装为已知的诱饵系统。该掩饰策略让攻击者难以发现目标,最终因失去攻击目标自动放弃攻击。

**以假拟真:**构建虚假资源,吸引攻击者访问这些虚假资源,同时检测和观察攻击活动,挖掘攻击意图。该模拟策略让攻击者陷于骗局,消耗攻击者的资源,延缓或消化攻击。

本文探索以假拟真策略,其诱捕机制包括骗局中诱饵设计和攻击检测两部分。

**定义 1** 诱饵(Bait)<sup>[24-25]</sup>:一种防御资源,  $B=\{R, T, L\}$ 。其中,  $R$  为诱饵资源集合,  $T$  为资源的类型,  $L$  为资源的位置。

诱饵与域名一样,是一种 IOC(indicator of compromise)。传统 IOC,如域名、邮件地址等,都是攻击代码自身携带的、固有的签名。而诱饵是防御者设计和部署的,是一种非对称的防御资源,攻击者难以绕过。

**定义 2** 轨迹:在给定时间窗,进程访问资源的轨迹  $S=\{P, R, O, C\}$ 。其中,  $P$  是用户进程集;  $R$  是用户使用或访问的资源集合;  $O$  是资源访问类型,包括读、写、执行等;  $C$  是资源访问的上下文,包括时间、触发条件等。

**定义 3** 异常:在给定的时间窗,进程  $p$  的资源访问轨迹与诱饵资源关联,则提示进程  $p$  异常。换言之,  $S.R \cap B.R$  不为空,表示进程异常。

因此,只要资源访问的轨迹符合定义 3,就可以检测到可疑攻击。

### 2.1 诱饵类型

诱饵是诱捕防御的战略资源。在日常数字世界中,用户或软件使用的任何信息都可以作为诱饵,如网络服务的特定端口、数据库/网站/远程主机的登录凭证、E-mail 地址、电话号码、文档等。诱捕防御的基本思路是感知与诱饵的交互行为,推断可能的未授权访问或者恶意攻击。

HoneyFile<sup>[26]</sup>引入了 3 类诱饵,即文件诱饵、用户名和口令诱饵、用户邮件名诱饵,以此提高入侵检测的准确性。口令诱饵和邮件名诱饵用以弥补文件诱饵的单一性。Decoy-document<sup>[27]</sup>采用了更多的 Honey token(诱饵的一种表示),如 contract.pdf 含有 Beacon(也称 Web Bug)。Beacon 是一种自动远程访问防御者服务器的方式,如文档中的远程链接。攻击者一旦打开该文档,Beacon 会自动连接远程服务器,借此可以用来检测内部攻击和异常。此外, Bogus-program<sup>[28]</sup>作为 Honey token 可以有效检测程序盗版。Honey-token 也广泛应用于其他安全防御,如口令猜测攻击检测<sup>[10]</sup>、漏洞触发的攻击流量转发检测<sup>[11]</sup>、诈骗电话和广告电话检测<sup>[29-30]</sup>、勒索软件检测<sup>[31-32]</sup>等。

Honey-token 同样可以被引入软件安全中防御代

码复用攻击<sup>[18]</sup>。通过向软件中注入一些指令序列作为 Trap（陷阱），这些指令序列被定义为 HoneyGadget<sup>[33-34]</sup>，其功能是警示用户，终止软件执行等。一旦攻击者使用这些注入的指令，就会触发报警。Readactor++在函数表中注入指向 HoneyGadget 的虚假指针<sup>[33]</sup>，增加 HoneyGadget 触发的机会。

诱饵类型、使用方式和用途如表 1 所示。诱饵类型分为 File、Data、Code 这 3 种类型。File 表示诱饵文件（文档），是指部署在文件系统中的特定格式文件，如桌面上的通信录文件或者文档文件夹中的技术报告、软件文件等。一旦发现有对这些文件的访问操作，安全工具就提示异常。而 Data 属于一种元数据，表示有语义的一串字节序列，指可以从公开网络环境获取、从远程网络访问的诱饵，包括但不限于网络端口、登录凭证、电话号码等，数据诱饵可以嵌入文件、程序、数据库中。防御方有意或无意在社交平台、论坛留言板等地方泄露这些诱饵，诱使攻击者使用，从而观察攻击行为和攻击意图。Code 表示可以直接执行的机器代码。攻击者一旦使用这些代码就会触发报警。诱饵使用方式描述了攻击者使用诱饵的位置，分为本地（Local）和远程（Remote）。前者表示攻击代码已经在目标主机运行，准备获取目标中的敏感信息；后者表示攻击者远程使用这些诱饵，试图远程访问目标系统。诱饵用途描述诱饵在软件安全中的作用，如检测网络异常访问、文件非授权访问、软件盗版、勒索软件、漏洞攻击等。以 honeytable<sup>[35]</sup>为例，它部署 Data 格

式的诱饵，用于检测攻击者从远程发起的针对数据库的非授权访问。

## 2.2 诱饵的生成与部署

诱饵的生成直接关系到诱饵的有效性。目前大多数文献对诱饵生成没有特别说明，仅仅说明诱饵的属性更新，诱饵内容需要包含一些敏感数据<sup>[25]</sup>，如登录凭证、社会保险号、邮件地址等。如果这些敏感数据在流量日志中被发现，则判定为从网络层检测到异常或攻击。文献[26]单独区分文件诱饵、登录凭证诱饵、邮箱诱饵。文献[25-26]均未介绍诱饵构造的方法。

文献[27-28]引入了一种新的诱饵——Beacon，可以远程检测主机系统受到了入侵，其基本思路是利用办公软件的宏或 PDF 的 Javascript 将一个 URL 植入文档或软件中，该 URL 随文档或软件的打开自动远程访问防御者服务。

个人身份信息（PII, personally identifiable information）包含个人隐私，如电话号码、住址等，一直是攻击者希望获取的信息。文献[36]分析了姓名、性别、职业、收入等属性的分布，以及收入与职业、地域（经纬度）的关联，解析了身份证号码和电话号码的构成，设计和实现了一种自动的个人身份信息生成方法。利用搜集的信息，该方法可以直接输出常见的个人身份信息，这些信息满足各个属性的分布关系以及属性之间的关联约束。因数据缺失和隐私保护等，该文献没有讨论信用卡、邮件地址、医疗记录、购买偏好、娱乐偏好的生成。

数据库在各个组织和单位广泛应用，如存放

表 1 诱饵类型、使用方式和用途  
Table 1 Types, usages and uses of bait

| 诱饵                | 诱饵类型      | 使用方式         | 用途         | 备注              |
|-------------------|-----------|--------------|------------|-----------------|
| E-mail 地址、登录凭证    | Data      | Remote       | 异常告警       |                 |
| HoneyFile         | File+Data | Local+Remote | 文件和数据非授权访问 |                 |
| Decoy-document    | File+Data | Local+Remote | 文件和数据非授权访问 | Beacon          |
| honeytable        | Data      | Remote       | DB 的非授权访问  |                 |
| Bogus-program     | File+Data | Local+Remote | 软件盗版       | Beacon          |
| HoneyWord         | Data      | Remote       | 口令猜测       | 口令              |
| Honey-patches     | Data      | Remote       | 漏洞攻击重定向    |                 |
| Honey-pot、Mobipot | Data      | Remote       | 钓鱼/推销电话    | 电话号码            |
| R-locker          | File      | Local        | 勒索软件       | 管道              |
| Trap、Codearmor    | Code      | Local        | 漏洞检测       | ROP gadget 虚假指针 |

企业的客户数据、大学的学生成绩等。从网络诱饵元素,如网络端口、IP 地址、域名等,很容易延伸到数据库的数据记录。HoneyGen<sup>[37]</sup>给出了一种伪造数据库元组的构造方法,利用伪造的数据记录感知异常和入侵。该方法希望产生高质量的伪造数据记录,让攻击者无法识别真伪。该方法分为约束抽取、元组生成和元组选择 3 个阶段。约束抽取收集数据表的字段属性,区分敏感字段和一般字段,构建外键、主键、字段之间的关联约束、连续字段的边界、离散字段的值集。元组生成利用混淆和泛化技术生成候选元组,同时,这些元组满足抽取的约束。元组选择利用对数似然估计来计算候选元组与真实元组的相似度,选择相似度较高的生成元组作为诱饵数据。图 2 描述了 Honeytoken DB 的生成过程,其约束生成和混淆泛化直接影响诱饵的质量。

文献[28]研究利用程序变换 (code obfuscation) 从给定软件 (proprietary source code) 生成 bogus software, 其人工构造的 Beacon 隐藏在文档说明 (pdf)、API 使用说明 (html) 以及代码库中,其程序变换包括符号变换 (类、方法、变量),代码等价变换,垃圾代码注入,结构变化 (增加新类、方法、变量、控制流混淆) 等。当攻击者获取了这些 bogus software, 执行编译后的程序,或者打开含有 Beacon 的文档时, Beacon 自动向服务器发送告警,表明该 Beacon 对应的程序 (软件) 被盗。如果被盗的文档或软件在断网或沙箱环境中打开,则此 Beacon 失效。如果 Beacon 有较强的触发约束,则攻击者难以检测到 Beacon,

从而提升诱饵的价值。

文献[10]提出了 Honeywords 方案来保护口令文件,通过在口令文件中增加额外  $N-1$  个假的身份凭据来增强安全性。该方案在已有的口令尾部追加给定长度随机值作为诱饵口令,也可以分析已有口令的模式 (如 W4D1W5, 表示口令前 4 位为字母、中间 1 位为数字、后 5 位为字母),随机生成符合该模式的诱饵口令。如果口令文件被偷并被破解,攻击者将会面对  $N$  个不同的口令,其中,只有一个是正确的。如果攻击者使用任意假口令,将会触发一个告警,系统管理员就会知道口令数据库已经被盗并被破解。

### 2.3 诱饵的度量

诱捕防御本身是一种概率防御机制。如果部署的诱饵没有任何访问行为,则该防御失效。该机制的有效性依赖于诱饵的构造和诱饵的部署。诱饵应该尽可能部署在攻击者必经的搜索路径上,同时诱饵对攻击者而言是难以辨识的。HoneyFile<sup>[26]</sup>讨论了诱饵文件 MAC(modified time, access time, created time) 时间属性和文件访问日志的更新,防止攻击者从文件的时间属性鉴别出诱饵文件。诱饵的数量和位置直接影响异常检测的准确性和及时性<sup>[38]</sup>。因此,攻击者和防御者在诱捕防御上不断对抗、演化,诱饵度量定性地反映了诱饵的质量。

当诱饵生成和部署后,攻击者、用户都可能会访问它,而安全工具会探测它。对攻击者而言,诱饵是真实的、动态的、有诱惑且容易发现的。对用户而言,诱饵是可辨识的、无干扰的。对检

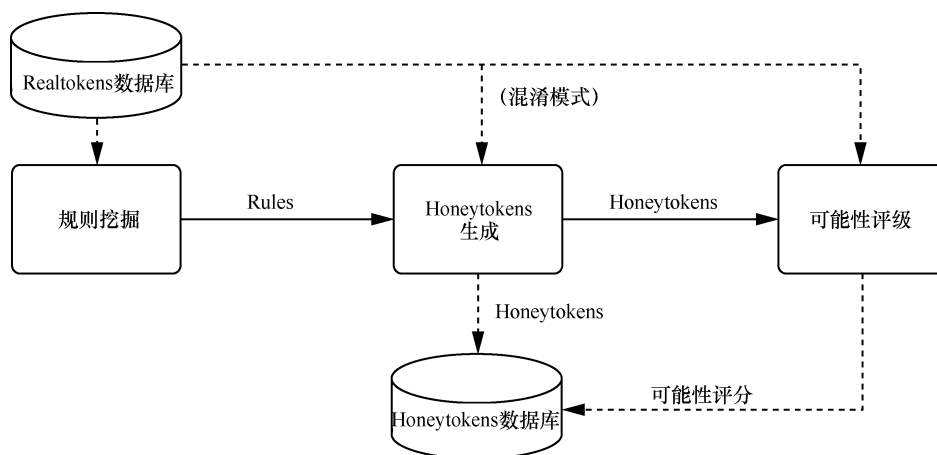


图 2 数据表诱饵的生成框架  
Figure 2 The generation framework of the data table baits

测工具而言, 诱饵的访问是可检测的、可区分的。文献[26]给出了诱饵度量的6个维度。

1) 真实性 (believability): 诱饵从形式和内容上无法与真实资源区分。与真实文件相比, 文件诱饵的文件名、MAC、访问日志是无差别的, 且文件诱饵的内容也是无差别的。在实际检测中, 只要诱饵被访问, 就会发出告警。有时, 诱饵内容的真实性在现实系统中会忽略。

2) 动态性 (variability): 诱饵需要随着时间和用户活动而变化。诱饵的属性, 如访问时间会更新; 诱饵的数量随用户活动而更新。同时, 不同系统上部署的诱饵不同。

3) 诱惑性 (enticement): 诱饵的诱惑性依赖于攻击者的兴趣或意图, 即诱饵必须蕴含攻击者的目标。攻击者关注高价值的文档, 如包含口令、社会保险号、银行账户信息、在线支付信息等。Windows 操作系统的资源管理器提供文档的关键词搜索, 攻击者利用该搜索功能可以定位到其认为的高价值文档。此外, 攻击者可以通过浏览目录发现诱惑性高的文件名, 如 password.txt。因此, 从诱饵的属性和内容两个方面体现诱惑性。

4) 醒目性 (conspicuousness): 诱饵命中攻击者搜索的次数或概率。当攻击者在受害者主机中不断进行文档搜索, 防御者希望其每次搜索结果中都含有诱饵。换言之, 诱饵容易被攻击者发现并访问。醒目性依赖于诱饵部署的位置 (目录)、诱饵的诱惑性、攻击者的搜索策略 (文件的属性搜索、内容搜索)。

5) 无干扰 (non-interference): 诱饵不会影响正常用户的活动。用户可有效识别出系统中的诱饵, 从而避免访问诱饵。同时, 即使用户无意访问诱饵, 用户也能及时发现, 并终止对诱饵的访问。换言之, 尽量减少诱饵在系统中的副作用。

6) 可检测性 (detectability): 检测性是指安全工具及时检测到诱饵访问的能力。一旦发现诱饵访问, 能立刻进行捕获。诱饵的真实性、动态性、诱惑性、醒目性、无干扰是保证攻击检测的基础。这种检测需要排除正常用户的误用, 以及对抗攻击者的绕过技术。

文献[7]定义了网络欺骗的4个属性: 机密性 (confidentiality)、可鉴别性 (authentication)、可

用性 (availability)、可控性 (controllability)。其中 confidentiality 对应 believability, authentication 对应 detectability, availability 对应 non-interference。特殊情况下, 攻击者可能识别出诱饵和伪造的环境, 并利用伪造环境提升攻击能力。为此, 防御者必须确保诱饵和伪造环境的可控性。这里, 可控性超出了软件安全确保的范围, 本文不单独讨论。

攻击者可能在获取诱饵资源后进行分析和识别, 从而绕过诱捕防御, 本文将鉴别诱饵真实性的能力分为低级 (low)、中级 (medium)、高级 (high) 3类。

低级: 攻击者没有鉴别诱饵, 默认这些资源都是真实的。遇到诱饵资源时, 会直接访问这些资源。

中级: 攻击者获得这些文档以及文档中的敏感数据后, 进行简单的验证, 如时间属性的验证<sup>[39]</sup>、折旧验证<sup>[40]</sup>, 然后利用网络搜索引擎等途径验证敏感数据 (凭证) 的真实性。

高级: 攻击者采用人工智能和大数据分析, 鉴别资源的真实性。

在诱捕防御设计中, 一般考虑攻击者具备中级鉴别能力, 以此提高系统的有效性。

### 3 应用场景

在软件安全领域, 诱捕防御首先用于检测单位内部的威胁<sup>[25]</sup>。内部人员知晓单位网络结构和安全策略, 有便捷访问内部资源的优势。当安全策略存在缺点时, 内部攻击者会出现非授权或不当的资源访问。防御者把高价值的文件、数据库记录、医疗记录、包含登录凭证和社会保险号等作为诱饵, 部署在文件系统和数据库系统中。一旦检测到对诱饵的访问行为, 就报警。此外, HoneyFile<sup>[26]</sup>作为诱饵部署在文件服务器, 辅助入侵检测系统 (IDS), 降低入侵检测的误报率和漏报率。Decoy document<sup>[38]</sup>的误报源于合法用户利用资源管理器显式浏览或者应用软件隐式浏览诱饵, 白名单机制可以降低误报率。本节重点介绍诱捕防御在勒索软件检测、漏洞攻击检测等方面的应用。

#### 3.1 勒索软件检测

根据搜索策略, 勒索软件会遍历文件系统,

搜索和加密符合搜索策略的文件,直到遍历结束。有研究从勒索软件的文件访问行为和文件内容的熵两个角度检测勒索软件<sup>[31,41-43]</sup>。它们的检测思路是:前者利用文件访问行为频繁、访问会跨越不同的路径,以及存在明显的文件内容加密操作和文件写操作等特点;后者则依据加密文件的熵值普遍偏高。这两种判据在压缩工具和正常加密工具环境下存在误报的情况。而且这些实时检测工具依赖于采集的数据量,采集足够的数据量才能触发报警。当报警发生时,部分文件已经加密,存在检测响应时延。

文献[39]分析了 1 000 多个勒索软件,解析了它们的文件加密、文件删除和锁屏勒索行为。同时,讨论了利用文件访问活动检测勒索软件的措施,但攻击者可以模仿正常用户的文件访问绕过行为检测。例如,勒索软件可以选择访问和加密部分文件。为此,提到可以设置诱饵资源以捕获勒索软件。UNVEIL<sup>[44]</sup>设计了诱饵资源环境,监控诱饵的删除操作和写操作,实现了检测勒索软件的沙箱。其诱饵资源环境包括文档、密钥和证书、压缩文件、图片、音频、视频,并介绍了文件内容、文件名、文件时间、文件目录、文件个数的构建思路和原则。每一个待分析的样本对应一种诱饵资源环境,以此增强环境的多样性、减少诱饵环境被识别的机会。

文献[45]在用户主目录创建一个命名管道作为核心诱饵文件。为了提高诱饵在文件系统中的覆盖率,在每个目录创建该命名管道的符号链接。发现有对该管道的读行为时,系统将判定检测到勒索软件并报警。如果勒索软件识别出该类管道文件,则该检测方法容易被绕过。

RWGuard<sup>[31]</sup>在文件的实时监控中部署了 txt、doc、pdf、ppt、xls 这 5 类诱饵文件,其文件内容来源于文件夹中的同类型文件,以增强诱饵的真实性。诱饵文件与正常文件的相似性可能导致用户难以区分诱饵文件,影响用户对诱饵的识别。

文献[46]根据 Windows 系统的文件遍历机制,设计以“sw”(最先遍历的符号)和“ew”(最后遍历的符号)开始的文件名和目录名作为诱饵。不管勒索软件采用何种文件遍历方式,都可以最

大概率捕获到该样本。文献[47]把 ASCII 表中字节值小的打印字符‘!’‘”’‘#’作为诱饵文件名的第一个字符,并在用户的桌面和文档文件夹、磁盘根目录以及网络驱动器中部署这些诱饵。然而,其添加文件名前缀的方式单一,容易被攻击者猜测到。

文献[32]提出了一种栈式文件系统 DcyFS,对可信进程提供真实的文件资源,对半可信进程提供部分真实的文件资源和诱饵资源,利用这些资源检测数据盗用和数据勒索。

文献[48]为了首要保护用户关键文件数据,提出在用户最近访问的 10 个文件夹和文件数量最多的 10 文件夹中部署诱饵文件。但是文件数量最多的文件夹不一定是重要的文件夹,如日志文件夹。同时,该部署方式难以预测用户对文件的更新活动。

文献[38]通过大量用户调查和实验分析表明,当单个文件夹内诱饵文件数量与所有文件总数之比小于 20%时,诱饵文件对正常用户的工作干扰可以忽略不计。

针对诱饵的生成、部署和检测技术,文献[20]提出识别诱饵文件的 3 种特征。其一是诱饵文件的静态特征。诱饵文件可能包含某些共有且易观测的特征,如文件的隐藏属性、文件内容为空、文件类型为管道、文件名或文件内容中的特定字符或字符串。其二是诱饵文件的文件修改时间特征。诱饵文件一般不会被用户或正常软件修改,诱饵文件的最后修改时间通常远早于系统中正常文件的最后修改时间,可以将最后修改时间远小于系统文件最后访问时间的文件标记为诱饵文件。其三是诱饵文件的位置特征。诱饵文件通常部署在文件夹的首部或尾部(按照文件名的首字节的 ASCII 值排序),以确保诱饵优先被勒索软件访问。因此,勒索软件可以采取随机选择文件夹的文件加密,以绕过诱饵文件的定向部署。

根据上文提到的诱饵文件衡量指标,现有针对勒索软件的各种诱饵文件的对比如表 2 所示。

基于诱饵文件的勒索软件检测研究目前仍未提出系统性的诱饵文件设计与部署方法,且往往由于诱饵文件部署不理想,检测误报率上升。本文从诱捕防御出发,重点关注诱饵文件的动态更

表 2 现有诱饵文件对比  
Table 2 Comparison of existing bait files

| 诱饵文件                        | 真实性 | 动态性 | 诱惑性 | 醒目性 | 无干扰 | 可检测性 |
|-----------------------------|-----|-----|-----|-----|-----|------|
| RWGuard <sup>[31]</sup>     | ✓   | ×   | ✓   | ✓   | ×   | ✓    |
| DcyFS <sup>[32]</sup>       | ✓   | ✓   | ✓   | ✓   | ×   | ✓    |
| UNVEIL <sup>[44]</sup>      | ✓   | ×   | ✓   | ✓   | ×   | ✓    |
| R-Locker <sup>[45]</sup>    | ×   | ✓   | ×   | ✓   | ×   | ✓    |
| Decoy Files <sup>[46]</sup> | ✓   | ×   | ×   | ✓   | ✓   | ✓    |
| IRDS <sup>[47]</sup>        | ✓   | ×   | ×   | ✓   | ×   | ✓    |
| DDT <sup>[48]</sup>         | ✓   | ×   | ✓   | ✓   | ×   | ✓    |

注：“✓”表示工具满足该项指标，“×”表示不满足该项指标。

新,提出了 Windows 平台上一种基于诱饵文件和进程文件行为的勒索软件早期检测方法。基于诱捕的勒索软件检测框架如图 3 所示,该框架主要由进程事件监控模块、诱饵部署与检测模块构成<sup>[49]</sup>。进程事件监控模块主要通过 Windows 事件追踪 (ETW) 和内核文件事件监控来记录系统中所有进程产生的各类事件,对收集到的事件进行类型过滤,并将处理后的事件交由诱饵部署与检测模块进行进一步分析。诱饵部署与检测模块主要负责设计并在系统中的关键位置动态部署诱饵文件,同时检测对诱饵文件的可疑访问行为,任何对诱饵文件进行写入或删除的进程都被认定为勒索软件。

为满足诱饵的设计要求,本节从诱饵文件的

文件命名、文件数量配置、文件类型配置、文件内容填充以及文件元数据配置等方面来阐述具体设计方式。

文件命名:为保证诱饵文件名与正常文件名尽可能相似。对于真实文件名样本库,过滤出勒索软件所加密的文件类型并分类,并进行分词处理,得到各类型诱饵文件的命名词库。构造诱饵文件名时,查询需要部署的诱饵文件类型所对应的分词词库,从中随机选取若干单词,然后从特殊字符的分词词库中随机选取若干字符与之前选取的单词进行组合,从而构成诱饵文件名。

文件数量配置:假设文件夹中诱饵数量与文件夹内正常文件总数的比值为  $R$ ,正常文件总数

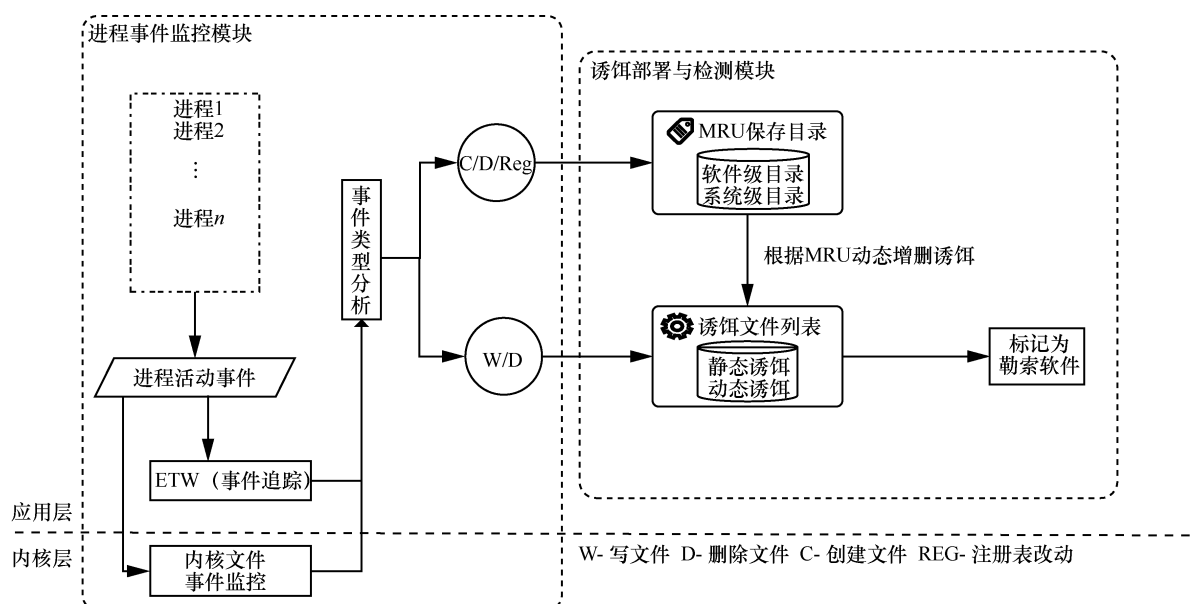


图 3 基于诱捕的勒索软件检测框架  
Figure 3 Ransomware detection framework based on deception

为  $N$ ，实际部署诱饵数量为  $D$ 。理论上来说， $R$  值越大，文件夹中部署诱饵的数量越多，勒索软件逃避检测的可能性就越小，但同时用户对用户的干扰越大。本文将  $R$  值设定为 20%，且当理论上需要部署的诱饵数量少于两个时， $D$  值为 2，即至少部署首尾两个诱饵文件，如下所示。

$$D = \begin{cases} 2, N \cdot R \leq 2 \\ N \cdot R, N \cdot R > 2 \end{cases}$$

文件类型配置：勒索软件会对文件类型进行过滤，其主要加密的文件类型主要有项目工程文件、文档类文件以及音视频类文件，本文在设计时仅随机选取一类文件作为诱饵的文件类型，在保证高吸引力的同时保证诱饵的多变性。

文件内容填充：对于诱饵文件的文件内容设计，本文首先根据文件类型填充与之相匹配的文件头部，然后根据文件类型填充随机数据，如在文档类文件中填充随机英文字符串，在多媒体类文件中则填充随机的音视频字节流数据。

文件元数据配置：文件元数据主要指文件访问权限、创建日期、访问日期、修改日期和创建者等信息。本文首先将创建者配置为系统当前用户，模拟正常文件的从属关系；由于勒索软件可能通过检查文件时间来跳过用户长期未访问的文件，本文按照“创建日期<修改日期≤访问日期<当前日期”的逻辑条件配置文件时间，并将文件时间设定为当前时间。在检测过程中，系统每隔一段时间自动将已部署诱饵的文件时间更新至最新时间，从而将所有诱饵文件伪装为最近才被用户访问过的状态。

在检测前，首先由诱饵部署与检测模块进行诱饵部署。诱饵部署包含两部分操作：其一是在勒索软件最先访问的文件夹中部署静态诱饵；其二是在当前用户最近访问过的文件夹中部署动态诱饵。部署完成后则将所有已部署诱饵添加至诱饵监控列表用于勒索软件检测。在检测过程中，由进程事件监控模块实时监控来自进程的各类事件。若捕获到的事件与用户最近访问文件夹的变化有关，则通知诱饵部署与检测模块，使其根据变化情况实时地增加或删除动态诱饵，并更新诱饵监控列表。若事件是写入或删除已部署的诱饵文件，则认为该事件来自勒索软件，由诱饵部署与检测模块收

集相关进程的信息并将其标记为勒索软件。

为了对基于诱饵文件的勒索软件检测进行评估，首先通过勒索软件文件遍历实验，确定勒索软件最先访问的文件夹，然后根据上述内容设计并部署诱饵文件，针对不同勒索软件家族的样本，计算对勒索软件的检测率和检测出时对应的平均已加密文件数，检测结果如表 3 所示。

表 3 勒索软件的检测结果  
Table 3 Detection results of ransomware

| 勒索软件家族（样本量）    | 平均已加密文件数 |
|----------------|----------|
| Reveton(102)   | 4.8      |
| CoinVault(87)  | 0.0      |
| Onion(115)     | 5.2      |
| WannaCry(204)  | 0.0      |
| Ryuk(48)       | 11.3     |
| Sodinokibi(90) | 0.6      |

由实验结果可知，对测试集中 6 类勒索软件家族的检测准确率均为 100%，且大部分勒索软件家族对应的平均已加密文件数在 5 个以下。进一步分析发现，WannaCry 和 CoinVault 等平均已加密文件数为 0 的勒索软件家族的文件遍历类型为直接遍历，即最先访问的是按字符顺序排序最小的磁盘的根目录，实验时该目录中已部署诱饵文件，因此这类勒索软件会在未加密任何正常文件的情况下就被检测出来；另外，平均已加密文件数不为 0 的勒索软件家族的文件遍历类型为特殊文件夹遍历和优先遍历类型，同家族间变种样本的遍历列表存在一定差异，每个变种可能根据自身需求配置一部分特殊文件夹，而这些文件夹不在诱饵静态部署目录中，因此导致部分文件被加密。

### 3.2 漏洞攻击检测

除数据诱饵和文件诱饵外，还有代码诱饵。ROP 攻击会重用软件的代码。通过在软件中部署代码诱饵来实施诱捕防御，以检测 ROP 攻击。同时，当发现漏洞攻击后，可以把攻击流量或攻击载荷重定向到蜜罐系统中，在蜜罐系统中进一步观察攻击轨迹、识别攻击意图。

#### 3.2.1 A-ROP 防御

攻击者一般利用内存错误<sup>[50]</sup>在目标软件中注入 ROP 攻击代码。ROP 编程从起初为复用库

函数的返回到库函数 (return-to-libc) 的攻击<sup>[51]</sup>, 逐渐发展为返回导向编程<sup>[52-53]</sup>、跳转导向编程<sup>[54]</sup>、虚表导向编程<sup>[55]</sup>等。不管 ROP 的编程如何演化, 其攻击成功都是依赖于 EIP 的跳转和 ROP 预设链接地址的准确性<sup>[13]</sup>。控制流完整性保护 (CFI) 技术<sup>[56]</sup>确保 EIP 会跳转到合法的代码区域, 但细粒度的 CFI 的开销比较大, 且难以应用到复杂结构和动态代码的代码指针保护<sup>[57]</sup>。

ROP 需要复用已知的代码和数据。随机化防御就是扰动目标软件的内存布局, 使得攻击者获得的代码地址无效<sup>[58]</sup>, 但内存泄露打破了内存布局随机化的假设<sup>[59-60]</sup>, 提高了 ROP 攻击的成功率。直观的对抗方法是缩短内存泄露信息的时效, 即动态随机化技术<sup>[61-64]</sup>。但动态指针的跟踪和修订会带来较多的时间开销<sup>[61]</sup>。此外, 动态随机化的空隙为攻击者提供了可乘之机。

文献[18]根据 ROP 复用目标软件中的代码这一特性, 讨论了在目标软件中注入诱饵代码 (即 HoneyGadget) 的设计思路。当检测到这些 HoneyGadget 被使用时, 系统可以警告用户或终止进程。代码诱饵即使在内存泄露环境下, 仍然是防御 ROP 攻击的一种有效选择。

CodeArmor<sup>[34]</sup>在保留初始版本外, 随机化整个代码区域, 并复制多个随机化代码版本, 利用系统调用和信号机制触发随机化版本的切换, 其时延较小。每次只有一个激活版本是有效的, 其他版本都是诱饵。CodeArmor 在切换版本时, 将原位置替换为 honey gadget 来吸引攻击者。Readactor++<sup>[33]</sup>在虚函数表中插入了大量虚假的虚函数指针作为诱饵引诱攻击者, 一旦攻击者执行了这些虚假指针指向的代码, 就会落入其陷阱中。

文献[65]提出在二进制文件中插入 nop 指令和 honey gadget, 并维护一个记录了插入 gadget 的地址表。一旦目标应用程序执行了敏感的函数调用, HoneyGadget 将暂停其执行并读取最近分支记录 (LBR) 缓冲区, 以检查 LBR 记录是否与地址表中的地址匹配。如果记录匹配, 则说明可能存在 ROP 攻击。

### 3.2.2 B-漏洞攻击劫持

攻击者攻击打补丁的 Web 服务时, 其漏洞探测和漏洞攻击会失效, 此时, 无法感知攻击者的攻击意图, 也没有办法收集攻击工具 (远程控制工具) 的信息。为了吸引攻击者和获取攻击载荷, 文献[11]借助蜜罐的思想, 设计了 honey-patches 和 honey-patching。honey-patches 在补丁代码中注入诱骗服务的触发点。当攻击者利用漏洞攻击载荷时, honey-patching 把攻击载荷转发给诱骗服务器, 实现引诱攻击者的目的。同时, 利用内存修订和网络连接保持以确保 Web 自身的安全性和诱骗服务的真实性。该方法需要人工辅助才能设计已知漏洞对应的 honey-patches, 难以自动化部署, 不能转发零日漏洞攻击。

### 3.3 Web 安全

为了评估钓鱼攻击, 研究受害者、钓鱼者以及研究者的行为, Phisheye<sup>[66]</sup>构建一个包含脆弱性的 Web 服务网站 (即蜜罐服务), 诱使攻击者攻击该网站, 并在该网站安装钓鱼攻击包。从近 5 个月数据日志中, 发现了 474 个网络钓鱼攻击包, 初步刻画了钓鱼攻击包的完整生命周期。此外, 29% 的攻击者通过搜索引擎查询到部署的蜜罐, 超过 40% 攻击者通过社交网络来共享蜜罐服务。同时, 反网络钓鱼服务 GSB 和 PhishTank 对蜜罐服务上钓鱼网站的平均检测时延约 12 天。

Tripwire<sup>[67]</sup>构建邮件系统的登录诱饵凭证 (用户名和口令), 然后利用这些诱饵凭证自动注册为第三方网站的用户。一旦有人利用网站的凭证登录给定邮件服务器, 就可以检测到攻击, 并推断出该凭证关联的网站被攻破。

DorkPot<sup>[68]</sup>利用 GHDB (Google hacking DataBase) 的 dork 构建蜜罐网页, 并把这些蜜罐网页部署在云服务平台, 对外提供 Web 访问。攻击者利用谷歌的搜索引擎, 搜索并访问这些蜜罐网页, 文献[68]统计了 dork 的使用频率, 由统计结果可知, 攻击者偏好搜索网络摄像头和路由器等在线设备、密码和数据库等重要数据。

## 4 讨论

传统的防御方式, 如流量入侵检测等, 属于

被动的防护手段。而诱捕防御属于主动防御技术,通过在 APT 攻击的各个阶段部署诱饵,可以快速检测和响应对主机的攻击。与传统防御手段对比,诱捕防御并不局限在特定的位置,而贯穿于防御体系的各个环节。它可以与传统防御相结合,如作为控制流完整性和地址随机化技术的补充手段,以对抗利用内存破坏漏洞的攻击,形成纵深的安全体系。

攻击者实施攻击时利用的资源和需要获取的资源都可以作为诱饵资源。IOT 设备广泛应用于智能家居、智慧工厂和智慧城市。为了获取攻击 IOT 设备的样本,可以虚拟各种 IOT 设备作为诱饵<sup>[69-70]</sup>,并部署到云端,实时监控这些诱饵设备的访问情况,捕获攻击样本,并跟踪识别攻击意图。电话号码和设备同样可以用作诱饵<sup>[29-30]</sup>,利用人工智能技术实施自动电话应答,尽可能多收集电话滥用信息,检测和发现语音钓鱼、电话营销、拒绝服务等电话威胁。此外, HoneyUSB 可以转发恶意 USB 固件访问主机的流量,可以识别攻击意图<sup>[71]</sup>; Honey page<sup>[72-73]</sup>用于检测浏览器扩展的信息泄露。

工业界从基于蜜罐和蜜网的网络安全防御逐渐延伸到基于诱捕的主机安全保护。为对抗勒索软件,安全工具在主机上部署文件诱饵,但这些诱饵文件命名方式和文件所在位置比较单一,容易被攻击者绕过。

同时,诱捕防御存在一定的局限性,诱捕防御本身是一种概率机制。为了提高其成功概率,需要确保诱饵与真实资源的相似性、诱饵的多样性、部署位置的随机性。这些将是诱捕防御机制的研究热点。文献[7]指出诱饵资源与被保护的业务系统高度一致。为了确保诱饵资源的相似性和多样性,诱饵资源还需要与用户行为高度一致,确保诱饵的动态性和新鲜性,使攻击者难以识别诱饵资源。部署位置的随机性是指诱饵所在目录、路径是难以被攻击者猜测的。同时,这种随机性可以确保攻击者尽可能访问这些目录和路径。另外,诱捕防御有被绕过或者利用的可能。例如,攻击者通过识别虚拟化环境和系统监控工具来判断当前环境是否安全。在部署诱饵的主机被控制后,攻击者可以将它作为跳板,进而攻击其他主

机和域服务器。

## 5 结束语

控制流完整性、地址随机化和诱捕防御构建了内存破坏漏洞的立体防御体系。前两者是软件自身完整性和内存布局的多样性的结合,目的是抑制攻击代码的执行。诱捕防御部署在代码上,可以抑制攻击代码的执行;部署在数据资源、文件资源上,可以从系统层面主动感知攻击。因此,从防御层看,诱捕防御是控制流完整性和地址随机化的有效补充。

本文从 APT 攻击流程出发,分析和总结了 APT 攻击的每一个环节都可以部署诱饵,并实施诱捕防御。同时,总结了诱饵类型,剖析了诱饵的生成、部署和度量。接着,分析了诱捕防御在勒索软件检测、漏洞检测、Web 安全等安全场景中的应用;最后,诱捕防御还可以应用到物联网安全、电话安全等领域。确保诱饵的真实性、动态性、新鲜性、多样性是未来研究工作的重点。

## 参考文献:

- [1] FORCE J T, INITIATIVE T. Security and privacy controls for federal information systems and organizations[J]. NIST Special Publication, 2013, 800(53): 8-13.
- [2] MONIKA, ZAVARSKY P, LINDSKOG D. Experimental analysis of ransomware on windows and android platforms: evolution and characterization[J]. Procedia Computer Science, 2016, 94: 465-472.
- [3] YE Y F, LI T, ADJEROH D, et al. A survey on malware detection using data mining techniques[J]. ACM Computing Surveys, 2017, 50(3): 1-40.
- [4] 诸葛建伟, 唐勇, 韩心慧, 等. 蜜罐技术研究与应用进展[J]. 软件学报, 2013, 24(4): 825-842.  
ZHUGE J W, TANG Y, HAN X H, et al. Honey-pot technology research and application[J]. Journal of Software, 2013, 24(4): 825-842.
- [5] 杨德全, 刘卫民, 俞宙. 基于蜜罐的主动防御应用研究[J]. 网络与信息安全学报, 2018, 4(1): 57-62, 78.  
YANG D Q, LIU W M, YU Z. Research on active defense application based on honeypot[J]. Chinese Journal of Network and Information Security, 2018, 4(1): 57-62, 78.
- [6] JAJODIA S, SUBRAHMANYAN V S, SWARUP V, et al. Cyber deception[M]. Cham: Springer, 2016.
- [7] 贾召鹏, 方滨兴, 刘潮歌, 等. 网络欺骗技术综述[J]. 通信学报, 2017, 38(12): 128-143.  
JIA Z P, FANG B X, LIU C G, et al. Survey on cyber deception[J]. Journal on Communications, 2017, 38(12): 128-143.
- [8] MILAJERDI S M, GJOMEMO R, ESHETE B, et al. HOLMES:

- real-time APT detection through correlation of suspicious information flows[C]//Proceedings of 2019 IEEE Symposium on Security and Privacy (SP). 2019: 1137-1152.
- [9] FU J M, LI L, WANG Y J, et al. Web scanner detection based on behavioral differences[M]//Communications in Computer and Information Science. 2019: 1-16.
- [10] JUELS A, RIVEST R L. Honeywords: making password-cracking detectable[C]//Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. 2013: 145-160.
- [11] ARAUJO F, HAMLEN K W, BIEDERMANN S, et al. From patches to honey-patches: lightweight attacker misdirection, deception, and disinformation[C]//Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. 2014: 942-953.
- [12] 乔向东, 郭戎潇, 赵勇. 代码复用对抗技术研究进展[J]. 网络与信息安全学报, 2018, 4(3): 1-12.
- QIAO X D, GUO R X, ZHAO Y. Research progress in code reuse attacking and defending[J]. Chinese Journal of Network and Information Security, 2018, 4(3): 1-12.
- [13] 彭国军, 梁玉, 张焕国, 等. 软件二进制代码重用技术综述[J]. 软件学报, 2017, 28(8): 2026-2045.
- PENG G J, LIANG Y, ZHANG H G, et al. Survey on software binary code reuse technologies[J]. Journal of Software, 2017, 28(8): 2026-2045.
- [14] 梁玉, 傅建明, 彭国军, 等. S-Tracker: 基于栈异常的 Shellcode 检测方法[J]. 华中科技大学学报(自然科学版), 2014, 42(11): 39-46.
- LIANG Y, FU J M, PENG G J, et al. S-Tracker: attribution of Shellcode exploiting stack[J]. Journal of Huazhong University of Science and Technology (Natural Science Edition), 2014, 42(11): 39-46.
- [15] ABADI M, BUDIU M H, ERLINGSSON Ú, et al. Control-flow integrity[C]//Proceedings of the 12th ACM conference on Computer and Communications Security. 2005: 340-353.
- [16] 王丰峰, 张涛, 徐伟光, 等. 进程控制流劫持攻击与防御技术综述[J]. 网络与信息安全学报, 2019, 5(6): 10-20.
- WANG F F, ZHANG T, XU W G, et al. Overview of control-flow hijacking attack and defense techniques for process[J]. Chinese Journal of Network and Information Security, 2019, 5(6): 10-20.
- [17] SNOW K Z, MONROSE F, DAVI L, et al. Just-in-time code reuse: on the effectiveness of fine-grained address space layout randomization[C]//Proceedings of 2013 IEEE Symposium on Security and Privacy. 2013: 574-588.
- [18] CRANE S, LARSEN P, BRUNTHALER S, et al. Booby trapping software[C]//Proceedings of the 2013 Workshop on New Security Paradigms Workshop - NSPW'13. 2013: 95-106.
- [19] DING S, FU J M, PENG B C. ModuleGuard: a gatekeeper for dynamic module loading against malware[J]. Wuhan University Journal of Natural Sciences, 2013, 18(6): 489-498.
- [20] GENÇ Z A, LENZINI G, SGANDURRA D. On deception-based protection against cryptographic ransomware[M]//Detection of Intrusions and Malware, and Vulnerability Assessment. Cham: Springer, 2019: 219-239.
- [21] 姚兰, 王新梅. 基于欺骗的网络主动防御技术研究[J]. 国防科技大学学报, 2008, 30(3): 65-69.
- YAO L, WANG X M. A study on the network active defense technology based on deception[J]. Journal of National University of Defense Technology, 2008, 30(3): 65-69.
- [22] WANG W, BICKFORD J, MURYNETS I, et al. Detecting targeted attacks by multilayer deception[J]. Journal of Cyber Security and Mobility, 2013, 2(2): 175-199.
- [23] 刘秀文, 傅建明, 黎琳, 等. 面向用户交互场景的信息欺骗分类及其威胁抑制机制[J]. 武汉大学学报(理学版), 2019, 65(2): 126-138.
- LIU X W, FU J M, LI L, et al. Taxonomy and threat suppression mechanism of user interactive scenario oriented information deception attacks[J]. Journal of Wuhan University (Natural Science Edition), 2019, 65(2): 126-138.
- [24] LIAO X J, YUAN K, WANG X F, et al. Acing the IOC game: toward automatic discovery and analysis of open-source cyber threat intelligence[C]// Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. 2016: 755-766.
- [25] SPITZNER L. Honey pots: catching the insider threat[C]//Proceedings of 19th Annual Computer Security Applications Conference, 2003. 2003: 170-179.
- [26] YUILL J, ZAPPE M, DENNING D, et al. Honeyfiles: deceptive files for intrusion detection[C]//Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop, 2004. 2004: 116-122.
- [27] BOWEN B M, HERSHKOP S, KEROMYTIS A D, et al. Baiting inside attackers using decoy documents[M]//Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009: 51-70.
- [28] PARK Y, STOLFO S. Software-based decoy system for insider threats[C]//ASIACCS'12. 2012.
- [29] GUPTA P, SRINIVASAN B, BALASUBRAMANIYAN V, et al. Phoney pot: data-driven understanding of telephony threats[C]//Proceedings of 2015 Network and Distributed System Security Symposium. 2015.
- [30] BALDUZZI M, GUPTA P, GU L, et al. Mobipot: understanding mobile telephony threats with honeycards[C]//Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security. 2016: 723-734.
- [31] MEHNAZ S, MUDGERIKAR A, BERTINO E. RWGuard: A real-time detection system against cryptographic ransomware[M]//Research in Attacks, Intrusions, and Defenses. Cham: Springer, 2018: 114-136.
- [32] TAYLOR T, ARAUJO F, KOHLBRENNER A, et al. Hidden in plain sight: filesystem view separation for data integrity and deception[M]//Detection of Intrusions and Malware, and Vulnerability Assessment. Cham: Springer, 2018: 256-278.
- [33] CRANE S J, VOLCKAERT S, SCHUSTER F, et al. It's a TRaP: table randomization and protection against function-reuse attacks[C]//Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. 2015: 243-255.
- [34] CHEN X, BOS H, GIUFFRIDA C. CodeArmor: virtualizing the

- code space to counter disclosure attacks[C]//Proceedings of 2017 IEEE European Symposium on Security and Privacy (EuroS&P). 2017: 514-529.
- [35] ČENYS A, RAINYS D, RADVILAVIČIUS L, et al. Implementation of honeypot module in DBMS oracle 9i2 enterprise edition for internal malicious activity detection[J]. IEEE Computer Society's TC on Security and Privacy, 2005: 1-13.
- [36] WHITE J. Creating personally identifiable honeypots[M]//Innovations and Advances in Computer Sciences and Engineering. Dordrecht: Springer Netherlands, 2009: 227-232.
- [37] BERCOVITCH M, RENFORD M, HASSON L, et al. HoneyGen: an automated honeypots generator[C]//Proceedings of 2011 IEEE International Conference on Intelligence and Security Informatics. IEEE, 2011: 131-136.
- [38] BEN SALEM M, STOLFO S J. Decoy document deployment for effective masquerade attack detection[M]//Detection of Intrusions and Malware, and Vulnerability Assessment. Berlin, Heidelberg: Springer, 2011: 35-54.
- [39] KHARRAZ A, ROBERTSON W, BALZAROTTI D, et al. Cutting the gordian knot: A look under the hood of ransomware attacks[M]//Detection of Intrusions and Malware, and Vulnerability Assessment. Cham: Springer, 2015: 3-24.
- [40] MIRAMIRKHANI N, APPINI M P, NIKIFORAKIS N, et al. Spotless sandboxes: evading malware analysis systems using wear-and-tear artifacts[C]//Proceedings of 2017 IEEE Symposium on Security and Privacy (SP). 2017: 1009-1024.
- [41] SCAIFE N, CARTER H, TRAYNOR P, et al. CryptoLock (and drop it): stopping ransomware attacks on user data[C]//Proceedings of 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS). 2016: 303-312.
- [42] CONTINELLA A, GUAGNELLI A, ZINGARO G, et al. ShieldFS: a self-healing, ransomware-aware filesystem[C]//Proceedings of the 32nd Annual Conference on Computer Security Applications. 2016: 336-347.
- [43] KHARRAZ A, KIRDA E. Redemption: real-time protection against ransomware at end-hosts[M]//Research in Attacks, Intrusions, and Defenses. Cham: Springer, 2017: 98-119.
- [44] KHARRAZ A, ARSHAD S, MULLINER C, et al. {UNVEIL}: A large-scale, automated approach to detecting ransomware[C]//Proceedings of 25th USENIX Security Symposium. 2016: 757-772.
- [45] GÓMEZ-HERNÁNDEZ J A, ÁLVAREZ-GONZÁLEZ L, GARCÍA-TEODORO P. R-Locker: thwarting ransomware action through a honeyfile-based approach[J]. Computers & Security, 2018, 73: 389-398.
- [46] LEE J, LEE J, HONG J M. How to make efficient decoy files for ransomware detection[C]//Proceedings of the International Conference on Research in Adaptive and Convergent Systems. 2017: 208-212.
- [47] EL-KOSAIRY A, AZER M A. Intrusion and ransomware detection system[C]//Proceedings of 2018 1st International Conference on Computer Applications & Information Security (ICCAIS). 2018: 1-7.
- [48] VORIS J, SONG Y B, SALEM M B, et al. Active authentication using file system decoys and user behavior modeling: results of a large scale study[J]. Computers & Security, 2019, 87: 101412.
- [49] 杨铮, 傅建明, 罗陈可, 等. 一种基于诱饵文件的勒索软件及时检测方法[J]. 武汉大学学报(理学版), 2020, 66(5): 473-482.
- YANG Z, FU J M, LUO C K, et al. A method of timely detection of ransomware based on decoy file[J]. Journal of Wuhan University (Natural Science Edition), 2020, 66(5): 473-482.
- [50] SZEKERES L, PAYER M, WEI T, et al. SoK: eternal war in memory[C]//Proceedings of 2013 IEEE Symposium on Security and Privacy. 2013: 48-62.
- [51] SHACHAM H. The geometry of innocent flesh on the bone: return-into-libc without function calls (on the x86)[C]//Proceedings of the 14th ACM Conference on Computer and Communications Security. 2007: 552-561.
- [52] BUCHANAN E, ROEMER R, SAVAGE S, et al. Return-oriented programming: Exploitation without code injection[J]. Black Hat, 2008, 8.
- [53] ROEMER R, BUCHANAN E, SHACHAM H, et al. Return-oriented programming: systems, languages, and applications[J]. ACM Transactions on Information and System Security, 2012, 15(1): 2.
- [54] BLETSCH T, JIANG X X, FREEH V W, et al. Jump-oriented programming: a new class of code-reuse attack[C]//Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security. 2011: 30-40.
- [55] SCHUSTER F, TENDYCK T, LIEBCHEN C, et al. Counterfeit object-oriented programming: on the difficulty of preventing code reuse attacks in C++ applications[C]//Proceedings of 2015 IEEE Symposium on Security and Privacy. 2015: 745-762.
- [56] HU H, QIAN C X, YAGEMANN C, et al. Enforcing unique code target property for control-flow integrity[C]//Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. 2018: 1470-1486.
- [57] XU X, GHAFARINIA M, WANG W, et al. {CONFIRM}: evaluating compatibility and relevance of control-flow integrity protections for modern software[C]//28th USENIX Security Symposium. 2019: 1805-1821.
- [58] 傅建明, 林艳, 刘秀文, 等. 云计算环境下基于随机化的安全防护研究[J]. 计算机学报, 2018, 41(6): 987-1004.
- FU J M, LIN Y, LIU X W, et al. Survey of randomization defenses on cloud computing[J]. Chinese Journal of Computers, 2018, 41(6): 987-1004.
- [59] SEIBERT J, OKHRAVI H, SÖDERSTRÖM E. Information leaks without memory disclosures: remote side channel attacks on diversified code[C]//Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. 2014: 54-65.
- [60] 傅建明, 刘秀文, 汤毅, 等. 内存地址泄漏分析与防御[J]. 计算机研究与发展, 2016, 53(8): 1829-1849.
- FU J M, LIU X W, TANG Y, et al. Survey of memory address leakage and its defense[J]. Journal of Computer Research and Development, 2016, 53(8): 1829-1849.
- [61] BIGELOW D, HOBSON T, RUDD R, et al. Timely rerandomization for mitigating memory disclosures[C]//Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. 2015: 268-279.

- [62] CHEN Y, WANG Z, WHALLEY D, et al. Remix: on-demand live randomization[C]// Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy. 2016: 50-61.
- [63] LU K J, NÜRNBERGER S, BACKES M, et al. How to make ASLR win the clone wars: runtime Re-randomization[C]// Proceedings of 2016 Network and Distributed System Security Symposium. Reston, VA: Internet Society, 2016.
- [64] WILLIAMS-KING D, GOBIESKI G, WILLIAMS-KING K, et al. Shuffler: fast and deployable continuous code re-randomization[C]//Proceedings of 12th USENIX Symposium on Operating Systems Design and Implementation {OSDI}. 2016: 367-382.
- [65] HUANG X, YAN F, ZHANG L Q, et al. HoneyGadget: A deception based ROP detection scheme[M]//Science of Cyber Security. Cham: Springer, 2019: 121-135.
- [66] HAN X, KHEIR N, BALZAROTTI D. Phisheye: live monitoring of sandboxed phishing kits[C]//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. 2016: 1402-1413.
- [67] DEBLASIO J, SAVAGE S, VOELKER G M, et al. Tripwire: Inferring internet site compromise[C]//Proceedings of the 2017 Internet Measurement Conference. 2017: 341-354.
- [68] QUINKERT F, LEONHARDT E, HOLZ T. Dorkpot: a honeypot-based analysis of google dorks[C]//Proceedings of the Workshop on Measurements, Attacks, and Defenses for the Web (MADWeb '19). 2019.
- [69] PA Y M P, SUZUKI S, YOSHIOKA K, et al. IoT POT: analysing the rise of IoT compromises[C]// USENIX WOOT 2015. 2015.
- [70] DANG F, LI Z H, LIU Y H, et al. Understanding fileless attacks on linux-based IoT devices with HoneyCloud[C]// Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services. 2019: 482-493.
- [71] TIAN D J, BATES A, BUTLER K. Defending against malicious USB firmware with GoodUSB[C]//Proceedings of the 31st Annual Computer Security Applications Conference. 2015: 261-270.
- [72] XIE M F, FU J M, HE J, et al. JTaint: finding privacy-leakage in chrome extensions[M]//Information Security and Privacy. 2020: 563-583.
- [73] CHEN Q, KAPRAVELOS A. Mystique: uncovering information leakage from browser extensions[C]//Proceedings of the 2018 ACM

SIGSAC Conference on Computer and Communications Security. 2018: 1687-1700.

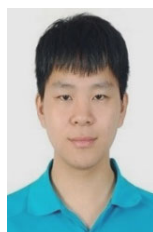
#### [作者简介]



傅建明（1969—），男，湖南宁乡人，博士，武汉大学教授、博士生导师，主要研究方向为系统安全、移动安全。



刘畅（1998—），男，河南商丘人，武汉大学硕士生，主要研究方向为移动安全。



解梦飞（1997—），男，山东青岛人，武汉大学硕士生，主要研究方向为系统安全。



罗陈可（1996—），男，湖南郴州人，武汉大学博士生，主要研究方向为系统安全、二进制安全。