一种面向 N 变体系统的时延隐蔽信道攻击及其对策研究

曾 威[1*], 扈红超[1], 霍树民[1], 周大成[1]

[1]国家数字程控交换中心 解放军信息工程大学 郑州 中国 450001

摘要 N 变体系统具有高可靠性和高安全性的特点, 能够有效防御多种安全风险, 现已广泛应用于金融、医疗、军事和网络空间等多个具有高安全性需求的领域。但是 N 变体系统特有的裁决机制为实施时延隐蔽信道攻击提供了潜在的实现途径。针对这种潜在的安全威胁, 本文首先分析了一种面向 N 变体系统的时延隐蔽信道攻击方法, 该攻击方法以信息论为基础, 利用 N 变体系统响应时延的差异特征来泄露系统信息。进而, 推导出了攻击者使用响应时延样本均值和样本方差作为特征统计量时的检出率公式。然后针对该时延隐蔽信道攻击方法, 从减少攻击者利用响应时延差异特征的角度上提出随机加扰策略、自适应加扰策略和先到先裁决策略三种防御策略, 随机加扰策略通过引入延迟使响应时延具有相同的统计特征, 自适应加扰策略通过动态调整裁决策略以平衡系统运行效率, 先到先裁决策略通过优化裁决算法以减少攻击者利用时延差异特征来泄露系统信息, 同时提升一定的系统性能。最后, 开发了基于 Nginx 的原型系统并进行了广泛的实验, 实验部分证明了该时延隐蔽信道攻击对 N 变体系统的安全威胁, 同时验证了三种防御策略的可行性与有效性, 性能对比测试结果表明先到先裁决策略相较于原裁决策略降低了 10%的系统响应时延, 吞吐量提升了 18%, CPU 利用率提升了 3%。

关键词 响应时延; 隐蔽信道攻击; N 变体系统; 信息泄露

中图法分类号 TN 915.08 DOI 号 10.19363/J.cnki.cn10-1380/tn.2022.05.06

A Time-delayed Covert Channel Attack and Its Countermeasures for N-Variant Systems

ZENG Wei[1*], HU Hongchao[1], HUO Shumin [1], ZHOU Dacheng [1]

[1]NDSC, People's Liberation Army Strategic Support Force Information Engineering University, Zhengzhou 450001, China

Abstract The N-variant system has the characteristics of high reliability and high security, and can effectively defend against a variety of security risks. It has been widely used in many fields with high security requirements, such as finance, medical treatment, military, and cyberspace. However, the unique adjudication mechanism of the N-variant system provides a potential way to implement time-delayed covert channel attacks. In response to this potential security threat, this article first analyzes a time-delayed covert channel attack method for N-variant systems. The attack method is based on information theory and uses the difference characteristics of the N-variant system response time to leak system information. Furthermore, the formula of the detection rate when the attacker uses the response delay sample mean and sample variance as characteristic statistics is derived. Then, for this time-delayed covert channel attack method, three defensive strategies, random scrambling strategy, adaptive scrambling strategy, and first-come-first-ruling strategy, are proposed from the perspective of reducing the attacker's use of response delay difference characteristics. The random scrambling strategy introduces a delay to make the response delay have the same statistical characteristics. The adaptive scrambling strategy dynamically adjusts the ruling strategy to balance system operating efficiency. The first-come-first- ruling strategy optimizes the adjudication algorithm to reduce the attacker's use of delay difference characteristics to leak system information, and at the same time improve certain system performance. Finally, a prototype system based on Nginx was developed and extensive experiments were carried out. The experimental part proved the security threat of the time-delayed covert channel attack to the N-variant system, and verified the feasibility and effectiveness of the three defense strategies. The performance comparison test results show that compared with the original ruling strategy, the first-come-first-ruling strategy reduces the system response delay by 10%, the throughput increases by 18%, and the CPU utilization rate increases by 3%.

Key words response time delay; covert channel attack; N-variant system; information leakage

通讯作者: 曾威, 硕士生, Email:zengwei19970605@163.com。

本课题得到国家自然科学基金项目(No. 62002383)、国家重点研发计划课题(No. 2018YFB0804004)资助。

收稿日期: 2021-04-02; 修改日期: 2021-07-22; 定稿日期: 2022-03-15

1 引言

随着人们越来越关注个人隐私安全, 当前数据中心网络环境下已经提出了各种隐私增强技术来保证用户隐私信息安全[1]。在一些特殊行业和关键设备中通常使用冗余架构系统来保证用户隐私安全, 同时提供高可靠性和高安全性。基于冗余架构构建的系统采用两套及以上相对独立的配置设计, 可以有效地防止由单点故障所引起的信息泄露和系统宕机。在网络空间安全领域, 冗余架构系统[2-3]通过在系统中引入冗余性和异构性, 以此来扰乱网络攻击链的构造和生效过程, 使攻击者攻击成功的代价倍增[4]。N 变体系统作为一种典型的冗余架构系统, 将相同的输入同步到具有异构性的多个副本变体上执行, 并监视多个副本变体输出的一致性以检测安全威胁, 实现目标程序的入侵容忍[5-6]。N 变体系统凭借其良好的可靠性和安全性, 已广泛应用于金融、医疗、军事、网络空间等具有高安全需求的信息系统中。

隐蔽信道攻击已成为当前造成用户信息泄露的主要安全威胁之一[7]。在隐蔽信道攻击场景中, 攻击者主要是利用信息发送方和接收方所共享的资源特征来编码隐蔽信息, 并借助本意不是用来传输信息的信道传递隐蔽信息[8]。通常攻击者可以通过修改网络通信协议或控制机制来嵌入秘密信息。隐蔽信道攻击一种典型的形式是基于时间的攻击, 攻击者可以通过测算观测时间内受害者产生的扰动来获取机密信息。目前, 业界已经对隐蔽信道攻击相关的安全威胁进行了广泛的研究。研究表明, 网络流量容易受到隐蔽信道攻击的安全威胁, 攻击者可以通过检查数据包大小、数据包时序和网络载荷数据, 从网络流量中推断出敏感信息。基于此, Devanathan 等[9]提出了一种隐私保护流量填充模型, 该模型可以兼顾用户隐私需求和流量填充成本。ML Wen 等[10]基于攻击者利用特定的数据包大小来识别敏感用户输入, 提出了一种新颖的数据包随机填充方法, 通过增加数据包填充过程中的不确定性以扰乱攻击者获取用户输入信息。王翀等[11]对近年来国内外与隐蔽信道相关的研究工作进行了系统的梳理、分析和总结, 旨在为后续开展与隐蔽信道相关的研究工作提供系统的参考。值得注意的是, 隐蔽信道攻击的攻击目标和攻击形式往往具有极大的不确定性, 因此很难从根源上对此类攻击进行防御。

本文发现 N 变体系统存在一种基于时延的隐蔽信道攻击安全威胁, 该攻击手段的出现为攻破 N 变体系统提供了新的思路, 但目前尚未发现相关研究, 且现有的防御方法难以防御此类安全威胁。对此本文利用 N 变体系统中特有的裁决机制, 提出一种面向 N 变体系统的时延隐蔽信道攻击方法, 该方法通过利用响应时延差异特征来传输额外信息, 造成系统信息泄露。

基于上述观察, 本文提出了 N 变体系统中一种基于响应时延的隐蔽信道攻击。在该隐蔽信道攻击场景中, 攻击者控制系统中一部分变体(如 N 变体系统开放白盒测试验证裁决模块安全性或者因供应链问题被植入木马或后门时), 动态地调整变体的输出时延以控制系统整体响应时延。在客户端, 攻击者通过对请求和响应数据包的时延特征进行分析, 可以解码出预传输的相关信息, 造成目标系统信息逃逸[12]。本文首先从信息论的角度出发来研究 N 变体系统中基于响应时延的信息泄露, 同时使用样本均值和样本方差的特征统计量来进行理论分析, 并推导出了检出率公式。其次, 为了防御基于时延隐蔽信道的安全威胁, 本文提出了 3 种防御策略: 随机加扰策略、自适应性加扰策略和先到先裁决策略, 并开发了基于 Nginx 的原型系统。最后, 在该原型系统上进行了广泛的实验, 证明了基于响应时延的隐蔽信道攻击的安全威胁, 同时验证了 3 种防御策略的可行性与有效性并进行了对比测试。实验表明, 先到先裁决策略可以有效地解决 N 变体系统中基于响应时延的信息泄露, 同时降低了系统 10%响应时延、吞吐量提升了 18%, CPU 利用率提升了 3%。

本文的贡献如下:

(1) 提出一种面向 N 变体系统的时延隐蔽信道攻击方法, 该方法以信息论为基础, 利用系统响应时延的差异特征来传输信息。

(2) 使用样本均值和样本方差的特征统计量来进行理论分析, 并推导出了检出率公式。

(3) 提出 3 种防御策略以应对基于时延的隐蔽信道攻击安全威胁: 随机加扰策略通过引入延迟使响应时延具有相同的统计特征; 自适应加扰策略通过动态调整裁决策略以平衡系统运行效率; 先到先裁决策略通过优化裁决算法以减少攻击者利用时延差异特征, 同时提升一定的系统性能。

(4) 开发了基于 Nginx 的原型系统并对本文所提方法进行验证, 实验结果表明3种防御策略有效地防御了时延隐蔽信道安全威胁, 且先到先裁决策略降低了10%的系统响应时延, 吞吐量提升了18%, CPU 利用率提升了3%。

本文后续部分安排如下: 第 2 章梳理了相关工

作, 包括隐蔽信道攻击、侧信道攻击和流量分析攻击等; 第 3 章中详细介绍了 N 变体系统中基于响应时延的隐蔽信道攻击, 包括攻击场景描述和攻击方法介绍, 以信息论为基础得出了当攻击者使用样本均值和方差作为特征统计量时的检出率公式; 在第 4 章中, 针对时延隐蔽信道的攻击特征设计了 3 种防御策略以应对此类型安全威胁; 在第 5 章在基于 Nginx 的原型系统中进行了广泛的实验, 验证了此类型隐蔽信道攻击对N变体系统的危害, 并对比3种防御策略的可行性与有效性; 第 6 章对本文的工作进行了结论, 并提出了未来的研究方向。

2 相关工作

N 变体系统实现了将相同的输入数据同步到具有多样性和异构性的多个副本变体上执行, 并通过监视系统中多个副本变体的输出响应以检测安全威胁[13-14]。根据相对正确公理(True Relatively Axiom, TRA), 攻击者难以同时控制多个变体的响应输出, N 变体系统可以将单个变体输出的不确定性问题转换成多模裁决机制可以感知的共模或者差模问题[15]。N 变体系统凭借其良好的可靠性和安全性, 已广泛应用于金融、医疗、军事、网络空间等具有高安全需求的信息系统中。但值得注意的是, N 变体系统特有的裁决机制为实施时延隐蔽信道攻击提供了实现途径。

目前业界对不同类型隐蔽信道攻击进行了广泛研究, 涉及从被动观察流量的侧信道和主动观察流量的隐蔽信道。隐蔽信道攻击和侧信道攻击已成为数据中心环境下用户信息泄露的主要原因[16-17]。隐蔽信道主要是利用共享的资源特征来编码隐蔽信息, 并借助本意不是用来传输信息的信道传递隐蔽信息[18]。当前常用的隐蔽信道主要包括存储隐蔽信道(Covert Storage Channel, CSC)[19]和时间隐蔽信道(Covert Timing Channel, CTC)[20]。CSC 通过修改共享资源的数值(如数据包中的字段)来编码隐蔽信息, 接收方通过读取共享资源的数值来解码隐蔽信息; CTC 通过动态调节共享资源的时间特性(如数据包间隔时延[21])来编码隐蔽信息, 接收方通过观察共享资源的时间特性来解码隐蔽信息。侧信道攻击也称为边信道攻击[22], 主要是指通过非直接途径泄露系统状态信息, 攻击者通过测量和采集, 恢复出系统敏感数据[23]。从狭义上讲, 侧信道攻击特指针对密码算法领域的非侵入式攻击(如能量分析攻击、计时攻击和电磁分析攻击等), 通过加密电子设备在运行过程中泄露的信息来破解密码算法。从广义上讲, 针对安全系统或设备的非侵入式、半侵入式和侵入式攻击等“旁门左道”的攻击方式都属于侧信道攻击的范畴[24]。

隐蔽信道技术可以将特定的秘密信号嵌入目标流量中进而传递隐蔽信息。考虑到隐蔽信道使用特制的内容或时间特征将内部信息传输给外部攻击者, A Liu 等[25]设计了一种实时隐蔽信道检测系统 Observer, 通过运行一个模拟易受攻击的虚拟机的安全虚拟机, 以便可以实时识别两个虚拟机之间的差异。针对移动设备可能遭受到隐蔽信道攻击, JC Wang 等[26]设计了一种能够检测基于意图的隐蔽存储通道的系统, 并在 Android 平台上验证了该系统在防御存储型隐蔽信道攻击的有效性。考虑到窃听者可以标识实际的应用程序数据, 且对 Web 流量随机填充和分组舍入会产生高昂的成本, Akshaya A.K 等[27]提出了一种正式的隐私保护流量填充(Privacy Preserving Traffic Padding, PPTP)算法, 以降低产生的额外开销并防止基于加密流识别的侧信道攻击。Ling 等[28]提出了一种基于细胞计数器的针对洋葱路由器的攻击, 其中攻击者将信号嵌入到恶意出口 Tor 中目标流量的细胞计数器的变化中。

在侧信道攻击中, 攻击者可以通过记录流量特征进而恢复出敏感数据。S Kadloor 等[29]基于分组网络中用户间共享路由资源提出一种低成本流量分析攻击, 并验证了该攻击方式在获取特定用户流量信息上的有效性。G Xun 等[30]针对流量时序特征来进行流量分析, 并验证了家庭数字用户线路由器内部基于定时侧信道攻击的可行性。研究表明, Web 流量容易受到侧信道攻击的安全威胁, 攻击者可以通过检查数据包大小和时序模式, 从加密的网络流量中推断出敏感信息。基于此, Devanathan 等[9]提出了一种隐私保护流量填充模型, 该模型可以兼顾用户隐私需求和流量填充成本。ML Wen 等[10]基于攻击者利用特定的数据包大小来识别敏感用户输入, 提出了一种新颖的数据包随机填充方法, 通过增加数据包填充过程中的不确定性以扰乱攻击者获取用户输入信息。Ke Li 等[31]基于攻击者可以推断成对用户或者用户社交关系之间的实时通信, 提出了一种方法框架来验证侧信道信息泄漏, 该框架实现了通过匹配流量的成对时间序列来识别实时通信的用户。Xingrui Fei 等[32]从网络的角度提出一种独立于加密的方案, 该方案采用在网络边界处收集的 Web 流量来识别由 Web 用户的点击操作生成的 HTTP(S)请求。L Zhen 等[33]研究了一种基于网络延迟的新型侧信道攻击以推断出用户访问的网站, 并提出了基于 k-均值聚类和 K-匿名性的策略以确保流量整形在提供匿

名性的同时不会引起过多的延迟。

在时延隐蔽信道攻击中, 攻击者需要对流量数据进行分析以解码出相关机密信息。为了应对流量分析攻击, Mehta 等[34]介绍了一种用于防御 IaaS 云中网络侧信道泄漏的新系统 Pacer, 该系统通过在访客外部整形用户流量, 增加攻击者利用流量特征的不确定性, 实验验证了该系统可以有效地平衡系统安全性和系统开销。George Stergiopoulos 等[35]提出了一种使用机器学习来检测 TCP/IP 数据包的侧信道特征的网络流量监控系统, 实验表明该系统能够在广泛的攻击范围内有效地区分正常流量和恶意流量, 其恶意流量检出率约为 94%。Majid Sabbagh 等[36]提出了一种新的检测微体系结构侧信道攻击的方法, 实验表明该方法在面对指令高速缓存和数据高速缓存等侧信道攻击时提供了高效性和准确性。考虑到传统的入侵检测系统深度数据包检查非常消耗资源的特点, Hongda Li 等[37]开发一种轻量级的基于侧信道的异常检测系统, 用于流量分选, 以减少由入侵检测系统监控的流量。李莉等[38]提出了一种基于 Web 防火墙(Web Application Firewall, WAF)主动防御的策略, 在客户端到服务器端使用主动性安全防护机制来加固 HTTP 交互流程。胡洋瑞等[39]针对网络流量缺乏一定的标记数据集的问题, 提出了一种无监督的异常流量检测方法, 通过度量网络流量与正常行为的偏离距离来识别出异常流量。

总之, 隐蔽信道就是借助一条本意不是用来传输正常数据的通信信道, 侧信道是指在通信的信道中, 存在着多种能泄露系统或设备信息的物理状态方式。其中侧信道可能被用来构建隐蔽信道。通常来说, 许多侧信道都有可能成为隐蔽信道, 或者伴随着隐蔽信道。Ullrich J 等[40]确定了侧信道成为潜在隐蔽信道的必要性, 并对 20 个侧信道场景进行调研, 确定了其中 18 个侧信道有可能成为隐蔽信道, 其中 5 个侧信道甚至允许发展成为两个隐蔽信道。值得注意的是, 区分隐蔽信道攻击和普通的侧信道攻击的依据主要是看攻击者是否与被攻击者系统共谋。隐蔽信道攻击通常伴随着内外协同式攻击, 即被攻击者系统与攻击者共谋, 进而发动攻击, 泄露系统隐私信息。侧信道攻击方式通常对流量处于被动监听状态, 不涉及与被攻击者系统进行协同式攻击。

上述方法一定程度上有效地防御了不同类型的隐蔽信道攻击安全威胁, 但并未考虑到 N 变体系统中可能存在的基于时延的隐蔽信道攻击。这种时延隐蔽信道攻击方式能够有效地绕过 N 变体系统的裁决机制, 造成系统信息泄露。目前 N 变体系统中常用的裁决机制包括: 全体一致裁决机制[41]、大数裁决机制[42-43]、最大近似裁决机制[44]和基于历史信息的加权裁决机制[45]等。这些常用的裁决机制都是基于响应内容进行裁决处理的, 难以防御基于时延的隐蔽信道攻击安全威胁, 且很难被检测发现。因此, 本文研究了 N 变体系统中一种基于时延的隐蔽信道攻击, 推导出了基于特征统计量的检出率公式, 并通过实验验证了所提的三种防御策略的可行性与有效性。

3 威胁模型

本节中首先介绍 N 变体系统中存在的基于响应时延的隐蔽信道攻击威胁模型。进而, 在网络环境相对稳定的情况下, 从攻击者角度对基于时延的隐蔽信道攻击进行分析, 包括往返时延的分解和攻击者的攻击策略。最后得出了当攻击者使用样本均值和方差作为特征统计量的检出率公式。

3.1 攻击场景描述

N 变体系统实现了将相同的客户端输入同步到多个异构变体上执行, 裁决模块同时监视多个变体的执行, 并对多个变体的响应数据进行裁决。基于此, 本节发现攻击者可以利用响应时延的差异特征来发起基于时延的隐蔽信道攻击, 造成 N 变体系统裁决逃逸。N 变体系统中基于时延的隐蔽信道攻击威胁模型如图 1 所示, 模型包括客户端、攻击者、代理服务器和应用服务器。在一次完整访问流程中, 代理服务器在接收到客户端请求时, 将请求分发至后端多个应用服务器, 同时对多个应用服务器的输出进行裁决, 并将一致性响应内容传输给客户端。在该隐蔽信道攻击场景中, 假设攻击者能够控制后端少部分变体应用服务器, 通过对预传输信息进行编码, 有规

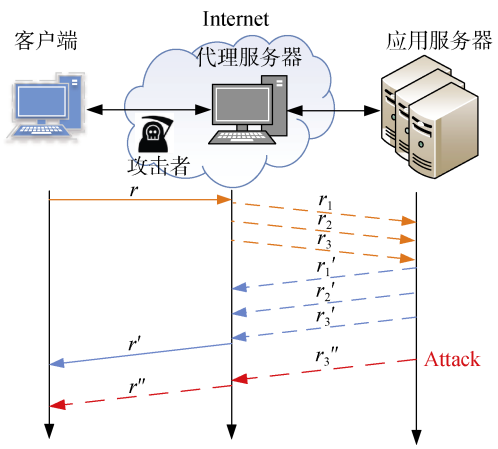


图 1 基于时延的隐蔽信道攻击威胁模型图

Figure 1 Covert channel attack threat model diagram based on time delay

律地控制变体侧应用服务器请求 r_3 的响应时延 $r_3^{''}$, 进而控制系统响应时延。同时在客户端侧, 攻击者可以通过记录请求和响应数据包的时间戳和长度来解码出预传输信息, 造成目标系统信息逃逸。

为了测量客户端与应用服务器之间请求的往返时延(Round-Trip Time, RTT), 攻击者需要观察通信过程并分析相关数据包以得出 RTT。假设客户端 C 通过代理服务器 P 来访问应用服务器 S, 且代理服务器没有故意引入额外的延迟以扰乱业务流量。此时客户端 C 与代理服务器 P 之间的 RTT 表示为 T_{CP}, 代理服务器 P 与应用服务器 S 之间的 RTT 表示为 T_{PS}, 代理服务器对数据流量的处理时间为 T_P。则客户端 C 与应用服务器 S 之间的 RTT 表示为:

$$T_{CS} = T_{CP} + T_P + T_{PS} \tag{1}$$

攻击者可以通过多次记录 HTTP GET 请求和响应之间的时间间隔来获得整个系统的响应时间分布。由于 N 变体系统的系统响应时延具有“反木桶原理”, 即系统响应时延取决于最长的变体服务器响应时延[46]。系统白盒测试时, 攻击者通过控制少部分应用服务器以控制系统响应时延, 按照一定的信道编码规律(如相移键控和频移键控)增加 T_{PS}, 导致系统响应时延 T_{CS} 也会按照相应的信道编码规律而增加。攻击者通过记录和分析数据包的 RTT, 可以解码出预传输的信息, 造成系统裁决逃逸。

这里举一个简单的例子来说明 N 变体系统信息泄露。假设攻击者控制客户端 C 正常访问应用服务器 S 上某一资源 S_i, 此时请求的 RTT 为 T_{S_i}。在攻击者控制少部分应用服务器响应延迟输出的情况下, 此时系统的 RTT 为 T_{S_j}。正常响应情况下 T_{S_i} 的分布和延迟响应情况下 T_{S_j} 的分布的重叠区域记作 N_{ij}, 则在该隐蔽信道攻击场景中服务器端泄露信息的概率为 $1-\mathrm{N}_{ij}$。本文主要使用到的数学符号与其含义归纳如表 1。

表 1 主要数学符号及其含义

Table 1 The main mathematical symbols and their meanings

数学符号	含义
C	客户端
P	代理服务器
S	应用服务器
T	系统响应时延
i	第 i 次请求为正常响应
j	第 j 次请求为延迟响应
N_{ij}	正常响应和延迟响应分布的重叠区域
r	正常响应和延迟响应分布方差比率
$G(s)$	高斯核函数
$f(s)$	具有核 G 的密度函数的核估计量
P_{err}	检测失败的概率
$Pd(X)$	检出率公式
D	Hellinger 距离的平方
C_i	第 i 个应用服务器的响应时延状态集
α	安全域值

3.2 攻击方法介绍

本节将通信过程分成三个阶段来研究系统响应时延特征。接着从攻击者的角度介绍了隐蔽信道攻击的攻击策略, 并得出了具有统计意义的贝叶斯决策规则。最后, 将检出率定义为攻击者正确识别预传输信息的概率, 并得出了当攻击者使用样本均值和方差作为统计特征量时的检出率近似公式。

3.2.1 往返时延 RTT 的分解

客户端请求的 RTT 包括三个部分: 客户端与代理服务器的 RTT 表示为 T_C, 代理服务器排队和处理时延记作 T_P, 代理服务器与远程服务器的 RTT 表示为 T_S。则客户端请求的往返时延 T 可表示为:

$$T = T_C + T_P + T_S \tag{2}$$

正态分布通常适合用来描述单位时间或单位空间内离散随机事件发生的次数。由于多个变体服务器的整体响应时延通常受限于传输延迟和变体服务器处理逻辑, 变体服务器整体响应时延具有一定的离散性, 服从正态分布的条件。代理服务器 P 和应用服务器 S_i 之间的响应时延用 t_i 表示, 则 $T_S \sim N\left(\max(t_1,t_2,...,t_n),\ \sigma_S\right)$。

本节聚焦攻击者控制代理服务器和应用服务器之间的交互, 而客户端与代理服务器之间的交互、代理服务器对流量的排队和处理时延不在本文主要研究范围内, 因此假设 T_C 和 T_P 也满足正态分布条件, 即 $T_C \sim N\left(\mu_c,\sigma_c^2\right)$, $T_P \sim N\left(\mu_p,\sigma_p^2\right)$。

攻击者控制响应数据正常返回到客户端时的响应时延表示为 T_i, 攻击者控制响应数据延迟返回到客户端时的响应时延表示为 T_j。μ_i 和 μ_j 表示为 T_i 和 T_j 的平均值, σ_i^2 和 σ_j^2 表示为 T_i 和 T_j 的方差。此时 $\mu_i = \mu_c + \mu_p + \mu_{s_i}$, 且 $\sigma_i^2=\sigma_c^2+\sigma_p^2+\sigma_{s_i}^2$, 同理 $\mu_j = \mu_c + \mu_p + \mu_{s_j}$, 且 $\sigma_j^2=\sigma_c^2+\sigma_p^2+\sigma_{s_j}^2$。

定义 T_i 和 T_j 的方差比率为 r, 表示为

$$r=\frac{\sigma_i^2}{\sigma_j^2}=\frac{\sigma_c^2+\sigma_p^2+\sigma_{s_i}^2}{\sigma_c^2+\sigma_p^2+\sigma_{s_j}^2} \tag{3}$$

3.2.2 攻击策略

本节从攻击者的角度进行分析, 研究实施此隐蔽信道攻击的具体流程。假设攻击者在客户端采用基于贝叶斯决策理论的模式识别策略对数据流量进行分析, 则攻击过程可以分为两个阶段: 离线分析阶段和实施隐蔽信道攻击阶段。

阶段 1: 离线分析阶段

在实施隐蔽信道攻击之前, 攻击者需要尽可能多地掌握系统状态信息以及运行信息。攻击者通常会捕获大量数据流量来生成样本状态集。本文主要考虑由数据包响应时延差异特征所引入的安全威胁, 因此本节选取样本均值和方差作为特征统计量。本节从信息论的角度研究传输信道特征, 具有核 G 的密度函数的核估计量定义为

$$f(s)=\frac{1}{Mh}\sum_{i=1}^{M}G\left(\frac{s-S_i}{h}\right) \tag{4}$$

其中 h 是窗口宽度, 也称为平滑参数或带宽, S_i 是特征的第 i 个度量, M 是此类度量的数量, 函数 G 基于高斯核, 即

$$G(s)=\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{s^2}{2}\right) \tag{5}$$

根据对某一资源请求的 RTT 进行分析, 得出贝叶斯决策规则。贝叶斯决策规则可以表述如下:

$$P(T_i\mid s)\geqslant P(T_j\mid s) \tag{6}$$

即 $$f(s\mid T_i)P(T_i)\geqslant f(s\mid T_j)P(T_j) \tag{7}$$

其中 $P(T_i)$ 是应用服务器正常响应的先验概率, $P(T_i\mid s)$ 是用户在收集到的样本特征 RTT 为 s 时应用服务器侧正常响应的先验概率。

图 2 展示了正常响应情况 T_i 和延迟响应情况 T_j 下系统响应时延分布图, 令 d 为以下方程式的解, 即

$$f(T_i\mid s)=f(T_j\mid s) \tag{8}$$

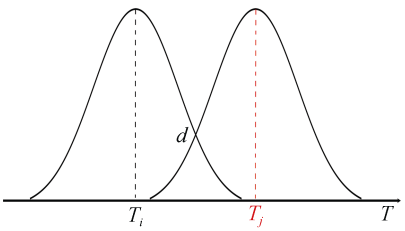


图 2 正常和延迟响应下响应时延分布图

Figure 2 Response time delay distribution diagram under normal and delayed response

假定方程有一个唯一的解。此时贝叶斯决策规则变为: 如果 $s\leqslant d$, 代表应用服务器正常响应; 如果 $s>d$, 则表示应用服务器延迟响应。值得注意的是, T_i 和 T_j 响应时延分布存在重叠区域, 难以进行细粒度判决, 此时检测失败的概率 P_{err} 为:

$$\begin{aligned}P_{\text{err}}=&P(T_j)\int_{-\infty}^{d}f(s\mid T_j)ds+\dots\\&P(T_i)\int_{d}^{+\infty}f(s\mid T_i)ds\end{aligned} \tag{9}$$

则检测成功的概率 P_{suc} 为:

$$\begin{aligned}P_{\text{suc}}=1-P_{\text{err}}=&P(T_i)\int_{-\infty}^{d}f(s\mid T_i)ds+\dots\\&P(T_j)\int_{d}^{+\infty}f(s\mid T_j)ds\end{aligned} \tag{10}$$

阶段 2: 实施隐蔽信道攻击阶段

一旦攻击者充分掌握系统运行特征信息, 攻击者便开始实施隐蔽信道攻击。在攻击者控制少部分应用服务器的情况下, 攻击者通过频移键控或相移键控等信道编码方式将变体侧预传输的信息编码成一串二进制比特流, 同时有规律地控制变体服务器的响应时延以传输二进制比特流信息。攻击者通过捕获客户端的数据包数据, 并基于在分析阶段得出的贝叶斯决策规则对大量样本数据进行分析(如攻击者预传输比特 1 的时候控制应用服务器侧延迟响应, 当攻击者预传输比特 0 的时候保证应用服务器正常响应), 以解码出变体侧预传输的信息。

基于时延的隐蔽信道攻击对于客户端和 N 变体系统裁决模块来说都是透明的。对于客户端, 变体服务器侧通常引入毫秒级的延迟, 客户端难以感知这一变化。对于裁决模块来说, 现有的裁决策略大多是基于响应内容进行投票表决, 这种攻击方式能够绕过现有裁决逻辑, 导致系统信息泄露, 对用户隐私信息产生了极大的威胁。

3.3 检出率

本节我们将检出率定义为攻击者正确识别预传输信息的概率。假设攻击者使用样本均值和样本方差作为特征统计量对数据流量进行分析, 同时给出了仅考虑样本均值、仅考虑样本方差和同时考虑样本均值和方差三种不同特征统计量情况下的检出率公式。

(1) 考虑样本均值的情况: $\{X_1,X_2,\dots,X_n\}$ 表示为一组大小为 n 的 RTT 数据。样本均值 $\overline{X}$ 是样本中元素的平均值, 即

$$\overline{X}=\frac{\sum_{i=1}^{n}X_i}{n} \tag{11}$$

请注意, 样本均值 $\overline{X}$ 是一个随机变量, 并且 μ 是样本 X 均值的无偏估计。当攻击者使用样本均值作为特征统计量时, 检出率的公式可以表示为[33]:

$$Pd\left(\overline{X}\right)\approx e^{-1/4\left(\mu_j-\mu_i\right)^2\frac{n}{\sigma_i^2+\sigma_j^2}}\frac{1}{\sqrt{2\left(\frac{1}{\sqrt{r}}+\sqrt{r}\right)}} \tag{12}$$

通过分析上述公式可以得出以下结论:

- 检出率 $Pd\left(\overline{X}\right)$ 随着样本大小 n 的增加而增加。值得注意的是, 如果攻击者可以收集足够多的响应时延样本, 那么攻击者可以进行充分研究, 攻击成功的概率也会随之上升。
- 检出率 $Pd\left(\overline{X}\right)$ 是延迟情况下和正常响应情况下响应时延均值之差(即 $\mu_j-\mu_i$)的递增函数。两种情况下响应时延均值差值越大, 即样本重叠率越小, 攻击成功的概率也会越高。

(2) 考虑样本方差的情况: 一组 RTT 样本数据 $\{X_1,X_2,...,X_n\}$ 的方差 Y 表示为:

$$Y=\frac{\sum_{i=1}^{n}\left(X_i-\overline{X}\right)^2}{n-1} \tag{13}$$

请注意, 样本方差 Y 是随机变量, 并且是样本 X 方差的无偏估计。当使用样本方差作为特征统计量时, 检出率的公式 $Pd\left(Y\right)$ 可以表示为[33]:

$$Pd\left(Y\right)\approx\max\left(1-\frac{V_y}{n-1},50\%\right) \tag{14}$$

其中

$$V_y=\frac{1}{2\left(1-\frac{\ln r}{r-1}\right)^2}+\frac{1}{2\left(\frac{r\ln r}{r-1}-1\right)^2} \tag{15}$$

通过分析上述公式可以得出以下结论:

- 检出率 $Pd\left(Y\right)$ 是关于样本大小 n 的递增函数。检出率随样本大小 n 的增大而增加, 这意味着攻击者收集的样本数据越多, 样本方差作为特征统计量时的攻击成功率就越高。
- 检出率 $Pd\left(Y\right)$ 是关于比率 r 的递增函数。比率 r 越小, 检出率越低。在代理服务器侧引入具有较大的排队和处理时延的情况下, 即比率 r 趋近于 1 时, 检出率趋近于 50%。这表明在系统存在较大随机延迟的情况下, 攻击者不能直接使用样本方差作为特征统计量来检测信息。

(3) 同时考虑样本均值和样本方差的情况:

本节同时考虑样本均值和样本方差两个特征统计量, 如 3.2.2 节图 2 所示, 攻击者控制响应消息正常返回到客户端时的响应时延表示为 T_i, 攻击者控制响应消息延迟返回至客户端时的响应时延表示为 T_j, 表示为

$$T_i\sim N\left(\mu_c+\mu_p+\mu_{s_i},{\sigma_c}^2+{\sigma_p}^2+{\sigma_{s_i}}^2\right) \tag{16}$$

$$T_j\sim N\left(\mu_c+\mu_p+\mu_{s_j},{\sigma_c}^2+{\sigma_p}^2+{\sigma_{s_j}}^2\right) \tag{17}$$

此时 $\mu_i=\mu_c+\mu_p+\mu_{s_i}$, 且 $\sigma_i^2={\sigma_c}^2+{\sigma_p}^2+{\sigma_{s_i}}^2$, 同理 $\mu_j=\mu_c+\mu_p+\mu_{s_j}$, 且 $\sigma_i^2={\sigma_c}^2+{\sigma_p}^2+{\sigma_{s_j}}^2$。当使用样本均值和样本方差作为特征统计量时, 检出率的公式 $Pd\left(Z\right)$ 可以表示为:

$$\begin{aligned}Pd(Z)=1-\int_{-\infty}^{d}\frac{1}{\sqrt{2\pi}\sigma_j}e^{-\frac{\left(x-u_j\right)^2}{2\sigma_j^2}}dx-\ldots\\ \int_{d}^{+\infty}\frac{1}{\sqrt{2\pi}\sigma_i}e^{-\frac{\left(x-u_i\right)^2}{2\sigma_i^2}}dx\end{aligned} \tag{18}$$

通过分析上述公式可以得出以下结论:

- 检出率 $Pd\left(Z\right)$ 是关于样本均值差值(即 $\mu_j-\mu_i$)的递增函数。样本均值差值越大, 样本分布重叠区域越小, 检出率越高。
- 在样本均值一定的情况下, 样本方差越大, 样本分布重叠区域越大, 检出率 $Pd\left(Z\right)$ 越小。这表明样本波动范围越大, 攻击者基于数据包时延差异特征来分析流量数据的难度越大, 同时也意味着可以给系统响应时延引入一定的不确定性以扰乱攻击者分析, 进而防御基于时延的隐蔽信道攻击安全威胁。

4 防御策略

为了防御 N 变体系统中基于响应时延的隐蔽信道攻击安全威胁, 本节提出三种防御策略: 随机加扰策略、自适应加扰策略与先到先裁决策略。随机加扰策略通过引入网络延迟, 使得所有请求的响应时延具有相似的统计特征, 以扰乱攻击者进行流量分析。自适应性加扰策略通过维护多个变体服务器响应时延状态集, 根据流量特征动态调整裁决策略。先到先裁决策略通过优化裁决算法, 使得攻击者难以利用数据包响应时延差异特征来传输信息, 同时降低了系统响应时延, 提升了系统运行效率。下面对三种防御策略进行详细介绍。

4.1 随机加扰策略

根据 Shannon 的完全保密准则, 如果客户端请求和每个资源之间的 RTT 分配相同, 则攻击者获得的信息将趋近于零。基于这一准则, 本节设计了一种随机加扰策略, 通过在代理服务器侧引入一定延迟, 使得所有响应时延在样本均值和样本方差的特征统计量上保持一致。

通过对应用服务器上某一服务进行多次访问, 记录一段时间内系统响应时延数据, 选择样本中响应时延均值最大的数据作为该服务的响应时延, 该服务的响应时延表示为 T_j。如图 3 所示, 第 i 次请求的响应时延均值为 T_i, 通过在代理服务器上引入延迟使得第 i 次的响应时延接近 T_j, 即满足 $T_i \approx T_j$。引入时延后的样本数据和该服务的响应时延数据重叠区域显著增大, 检出率将大幅下降, 说明引入随机加扰策略后攻击者难以使用样本均值和方差作为特征统计量来传输相关信息。

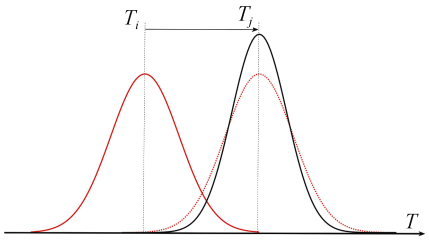


图 3 随机加扰策略原理示意图

Figure 3 Schematic diagram of the principle of random scrambling strategy

这里使用 Hellinger 距离的平方来度量来估计两个正态分布 $P \sim N\left(\mu_i, \sigma_i^2\right)$ 和 $Q \sim N\left(\mu_j, \sigma_j^2\right)$ 之间的距离[47], 表示为

$$D_i = \mathrm{H}^2\left(P,Q\right) = 1 - \sqrt{\frac{2\sigma_j\sigma_i}{\sigma_j^2+\sigma_i^2}} e^{-\frac{1}{4}\frac{\left(\mu_j-\mu_i\right)^2}{\sigma_j^2+\sigma_i^2}} \tag{19}$$

总距离为 $D=\sum D_i$。当采用随机加扰策略时, 此时 $\mu_i \approx \mu_j$, 此时正态分布 P 和 Q 之间 Hellinger 距离的平方表示为

$$\begin{aligned} D = \mathrm{H}^2\left(P,Q\right) &= 1 - \sqrt{\frac{2\sigma_j\sigma_i}{\sigma_j^2+\sigma_i^2}} e^{-\frac{1}{4}\frac{\left(\mu_j-\mu_i\right)^2}{\sigma_j^2+\sigma_i^2}} \\ &\approx 1 - \sqrt{\frac{2\sigma_j\sigma_i}{\sigma_j^2+\sigma_i^2}} \end{aligned} \tag{20}$$

随机加扰策略核心思想如算法 1 所示, 通过在代理服务器侧引入延迟, 使得攻击者在客户端就无法基于响应时延样本均值和样本方差的特征来解码出预传输的信息。

算法 1. 随机加扰算法.

输入: 收集 i 个响应, N 个远程服务器, 裁决函数 $M(x)$, 第 i 个响应的时延 Γ_i;

输出: 选择一致性响应输出, 上报异常消息

1. 初始化: Set 0<=i<=N
2. 最长响应时延 Γ
3. FOR i = 1, 2, ..., N DO
4. 收集并存储 i 个响应数据
5. 记录第 i 个响应的时延 Γ_i
6. IF i=N THEN
7. 使用 $M(x)$函数对 N 个响应数据进行裁决
8. IF N 个响应数据都一致 THEN
9. IF $\Gamma_i \leqslant \Gamma$ THEN
10. 随机选择一个响应数据输出至客户端
11. ELSE
12. $\Gamma=\Gamma_i$, 更新最长响应时延 Γ
13. END IF
14. ELSE
15. 上报相关异常消息
16. END IF
17. END IF
18. END FOR

3.3 节理论分析表明, 即使所有响应时延的样本均值相同, 样本方差仍有可能会泄露一些信息。相较于样本均值统计量, 样本方差统计量携带信息时更容易受到随机因素的干扰, 如代理服务器排队与处理时延存在较大波动的情况下, 基于样本方差的检测概率具有较大的不确定性。因此, 当采取随机加扰策略时, 同样可以选择一个最大样本方差 σ_s^2, 并在代理服务器侧引入延迟的同时使得所有的响应时延数据具有相同的方差。

随机加扰策略该策略有效地解决了基于响应时延的隐蔽信道攻击安全威胁。但在某些特殊领域, 如在用户对响应时延要求较高或存在大量并发请求的情况下, 延迟的引入会造成系统运行效率急剧下降, 影响用户正常访问相关服务。

4.2 自适应加扰策略

考虑到随机加扰策略使所有响应时延具有相同的分布, 缺乏一定的灵活性。本节所讨论的自适应加扰策略是对随机加扰策略上的一种优化方法, 主要用来平衡系统性能和系统安全性。在攻击者离线分

析阶段, 代理服务器侧对应用服务器侧的不同变体服务器响应时延进行分析, 维护 n 个状态集 $\{C_1,C_2,...,C_n\}$, C_i 表示第 i 个应用服务器的响应时延状态集。不失一般性地, 假设每个状态集数据服从正态分布, 即 $C_i \sim N(\mu_i,\sigma_i)$。

对于状态集 C_i, 记录时间 t 内响应时延数据超出自身状态集的次数 N。设定安全域值 α, 即单位时间内系统可容忍的时延数据超出自身状态集的次数。安全阈值取决于系统信息泄露容忍度和系统动态调度概率, 信息泄露容忍度高或动态调度频率高的系统中通常安全域值也高。用户可以调整安全域值 α 的取值以匹配自身安全性需求。自适应加扰策略核心思想如算法 2 所示, 若满足 $\frac{N}{t} \leqslant \alpha$, 说明此时系统基于时延的信息泄漏量在系统可接受范围之内, 此时维护并更新状态集 C_i。若满足 $\frac{N}{t} > \alpha$, 则表示信息泄漏量超出系统的安全域值, 此时代理服务器侧引入随机加扰策略, 使得所有响应时延具有相同的分布以扰乱攻击者的流量分析。代理服务器也可以标记该应用服务器 i 异常, 并将异常消息上报给异常处理模块。若一段时间满足 $\frac{N}{t} \leqslant \alpha$, 代理服务器侧则恢复原裁决策略。在引入自适应加扰策略时, 此时 $\mu_i \approx \mu_j \approx \max\{t_1,t_2,...,t_n\}$, 且 $\sigma_i \approx \sigma_j$, 此时正态分布 P 和 Q 之间 Hellinger 距离的平方趋近于 0。

算法 2. 自适应加扰算法.

输入: 收集 i 个响应, N 个远程服务器, 裁决函数 $M(x)$, 第 i 个响应的时延 Γ_i, 安全阈值 α;

输出: 选择一致性响应输出, 上报异常消息

1．响应时延状态集 C 初始化为 $\varnothing$, Set 0<=i<=N

2．维护自身状态集 $\{C_1,C_2,...,C_n\}$

3．FOR i = 1, 2, ..., N DO

4．收集并存储 i 个响应数据

5．记录第 i 个响应的时延 Γ_i

6．IF i=N THEN

7．使用 $M(x)$函数对 N 个响应数据进行裁决

8．对比与状态集 C_i

9．IF $\Gamma_i \cap C_i = \varnothing$ THEN

10．记录异常响应次数 N

11．IF $\frac{N}{t} > \alpha$ THEN

12．上报执行体 i 异常信息

13．ELSE

14．维护与更新状态集 C_i

15．END IF

16．END IF

17．END IF

18．END FOR

自适应加扰策略是对随机加扰策略的优化, 提供了一定的检错和纠错功能, 且能够自适应地调整系统响应时延, 有效地防御了此类型隐蔽信道攻击安全威胁。但值得注意的是, 自适应加扰策略和随机加扰策略都给引入了一定的延迟, 增加了系统响应时延, 降低了系统运行效率。

4.3 先到先裁决策略

在基于响应时延的隐蔽信道攻击中, 攻击者是通过控制少部分变体服务器的响应时延, 继而控制系统响应时延来传输额外信息。基于 N 变体系统响应时延具有“反木桶原理”, 且部分变体服务器的响应时延会直接影响整个系统的响应时延, 本节设计了一种先到先裁决策略来防御此类型隐蔽信道攻击安全威胁。先到先裁决策略核心思想如算法 3 所示, 通过优化代理服务器的裁决策略, 代理服务器在收集到响应数据个数满足基本大数判决条件时就

算法 3. 先到先裁决算法.

输入: 收集 i 个响应, N 个远程服务器, 裁决函数 $M(x)$;

输出: 选择一致性响应输出, 上报异常消息

1．初始化: Set 0<=i<=N

2．FOR i = 1, 2, ..., N DO

3．收集并存储 i 个响应数据

4．IF i>=1+N/2 THEN

5．使用 $M(x)$函数对 i 个响应数据进行裁决

6．IF 存在 $\lfloor N/2 \rfloor+1$ 个响应一致 THEN

7．从 $\lfloor N/2 \rfloor+1$ 个一致性响应中随机选择一个响应数据输出给客户端

8．ELSE

9．等待其余响应数据

10．END IF

11．END IF

12．IF i=N THEN

13．使用 $M(x)$函数对 N 个响应数据进行裁决

14．IF N 个响应数据存在不一致 THEN

15．上报相关异常消息

16．END IF

17．END IF

18．END FOR

进行裁决操作，同时在收集齐$\lfloor n/2 \rfloor+1$个一致性响应数据时立即将一致性响应数据中任意一个响应数据发送至客户端，同时对已收集的响应数据进行缓存。当代理服务器接收到所有响应数据时，再次对所有响应数据进行裁决，裁决结果一致时不再向客户端发送数据，裁决结果存在不一致时向异常处理模块上报相关异常信息。

在 N 变体系统中，代理服务器和应用服务器请求的 RTT 为$t_s=\max\{t_1,t_2,...,t_n\}$。当代理服务器裁决模块引入先到先裁决策略时，在第一阶段收集到多数响应数据时进行首次裁决，这里用$\{t_1,t_2,...,t_{\lfloor n/2 \rfloor+1}\}$来表示前$\lfloor n/2 \rfloor+1$个一致性响应时延数据，则此时代理服务器和应用服务器请求的 RTT 可表示为$t_{q1}=\max\{t_1,t_2,...,t_{\lfloor n/2 \rfloor+1}\}$。在第二阶段收集齐所有响应数据之后再次进行多模裁决，此时响应时延为$t_{q2}=\max\{t_1,t_2,...,t_n\}$。先到先裁决策略下代理服务器和应用服务器请求的 RTT t_s'表示为

$$\max\{t_1,t_2,...,t_{\lfloor n/2 \rfloor+1}\}\leqslant t_s'\leqslant\max\{t_1,t_2,...,t_n\} \quad (21)$$

公式 21 说明当采用先到先裁决策略时客户端请求的响应时延明显低于原裁决策略下响应时延。当攻击者使用样本均值和样本方差进行流量分析时，此时$\mu_i\approx\mu_j\approx\max\{t_1,t_2,...,t_{\lfloor n/2 \rfloor+1}\}$，且$\sigma_i\approx\sigma_j$，正态分布 P 和 Q 之间 Hellinger 距离的平方可以表示为

$$D_i=\mathrm{H}^2(P,Q)=1-\sqrt{\frac{2\sigma_j\sigma_i}{\sigma_j^2+\sigma_i^2}}e^{-\frac{1}{4}\frac{(\mu_j-\mu_i)^2}{\sigma_j^2+\sigma_i^2}}\approx 0 \quad (22)$$

根据 Shannon 的完全保密准则，当代理服务器使用先到先裁决策略时，系统信息泄露量趋近于 0。先到先裁决策略能够屏蔽少部分应用服务器对整体响应时延的影响，使得攻击者在客户端难以利用样本均值和样本方差作为特征统计量解码预传输信息。相较于原始裁决策略而言，先到先裁决策略既降低了客户端响应时延，增大了系统运行效率，又有效地解决了上述基于响应时延的隐蔽信道攻击安全威胁。

值得一提是，随机加扰策略和自适应加扰策略可以应用于N变体系统中常用的全体一致裁决机制、大数裁决机制、最大近似裁决机制和基于历史信息的加权裁决机制中，即通过引入一定的系统延迟，扰乱攻击者进行流量分析，但延迟的引入会降低一定的系统运行效率。进而，本文以 N 变体系统较为常用的大数裁决机制作为切入点，设计了一种先到先裁决策略来降低攻击者利用响应时延差异特征来传输信息，使得系统响应时延具有较大的随机性和不确定性，极大地增加了攻击者的攻击难度，有效地提升了系统的安全性。值得注意的是，上述全体一致裁决机制、最大似然裁决机制和基于历史信息的加权裁决机制都需要收集齐全部响应数据才能依据裁决策略进行裁决操作，受限于其裁决机制的前提条件，先到先裁决策略均难以适用于上述三种裁决机制。

5 实验结果及分析

为了模拟该攻击场景，并验证所提策略的安全性和有效性，实验部分使用了 Nginx 作为反向代理服务器。Nginx 是基于 c 语言的一款优秀的开源软件，具有占用内存少、并发能力强等优点。本文开发了基于 Nginx 的原型系统，并对其源码进行约 5000 行修改以验证所提 3 种策略的可行性与有效性。

5.1 实验设置

实验场景设置如图 4 所示，包括客户端、代理服务器、远程 Web 服务器。实验环境配置参数如表 2 所示，使用 Firefox18.05 浏览器作为客户端，同时关闭浏览器缓存机制，并安装最新的 Adobe Flash 插件[48]。攻击者使用 TCPdump 工具[49]来捕获和分析系统流量数据，同时可以通过修改 Web 服务器 Apache 的配置文件来引入一定的延迟。代理服务器(28 cores 2.20 GHz,

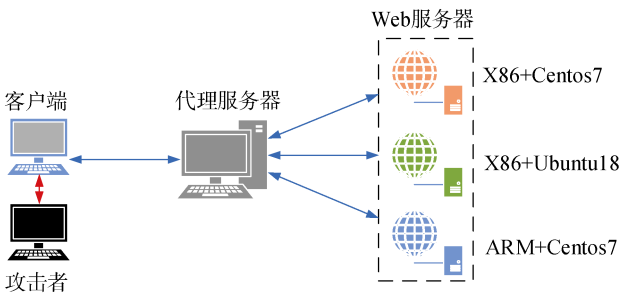


图 4 实验场景图

Figure 4 Experimental scene graph

表 2 实验环境配置参数

Table 2 Experimental environment configuration parameters

组件	基础架构	操作系统	应用程序
客户端	X86-64	Redhat 7	Firefox
攻击者	X86-64	Redhat 7	TCPdump
代理服务器	X86-64	Centos7	Nginx
Web 服务器 1	X86-64	Centos7	Apache
Web 服务器 2	X86-64	Ubuntu18	Apache
Web 服务器 3	ARM	Centos7	Apache

256 GB RAM, 10 Gb NIC)上安装基于 Nginx 的原型系统作为反向代理和负载均衡服务器，能够实现对用户请求进行分发和对多个变体响应数据进行多模裁决。3 台远程 Web 服务器配置不同架构和操作系统来模拟 N 变体系统环境，其中包括 2 台 x86 服务器(28 cores 2.00 GHz, 256 GB RAM, 10 Gb NIC), 1 台 ARM 服务器(16 cores 1.50 GHz, 32 GB RAM)。

5.2 检出率

本节模拟真实的隐蔽信道攻击场景来证明基于响应时延的隐蔽信道攻击对N变体系统的安全威胁。在攻击者控制一个远程 Web 服务器的情况下，控制客户端等间隔的向服务器端发送 50000 次请求，攻击者对预传输的信息进行编码，通过有规律的延迟其中 10000 次请求的响应时延来模拟该时延隐蔽信道攻击场景。客户端使用 libpcap 和 TCPdump 工具来捕获相关请求和响应数据包，并使用 TCPdump 来存储捕获的数据包。通过对大量流量数据进行分析，同时结合 3.3 节理论分析，得到了如图 5 所示的检出率与样本均值和方差特征统计量的关系。如图 5(a)所示，在仅考虑样本均值和同时考虑样本均值和方差的情况下，检出率随样本均值差值的增加而增大，且仅考虑样本均值的检出率趋近于 0.5，同时考虑样本均值和方差的检出率趋近于 1。如图 5(b)所示，在均值差值固定且仅考虑样本方差的情况下，检出率相对稳定，这与理论分析相对应，说明代理服务器侧排队和处理时延存在较大波动的时候，攻击者不能单独使用样本方差来作为特征统计量。而在同时考虑样本均值和方差的情况下，检出率随方差差值的增加而减少，即 $\Delta\sigma$ 越大，正常响应 T_i 和延迟响应 T_j 分布的重叠区域面积越大，此时检出率也会相应降低。值得一提的是，在均值差值 $\Delta\mu$=10 的情况下，攻击者考虑样本均值和样本方差作为特征统计量时，此时检出率趋近于 100%。说明攻击者能够使用样本均值和样本方差作为特征统计量来实施基于时延的隐蔽信道攻击，造成系统信息泄露。

为了验证三种防御策略对系统响应时延时序特征的影响，设置本节在同等测试环境下，设置 $\Delta\mu$=10，$\Delta\sigma$=10，客户端等间隔的向服务器端发送 50000 次请求，分别在代理服务器侧引入随机加扰策略、自适应加扰策略和先到先裁决策略，同时基于 3.3 节同时考虑样本均值和方差时的理论分析，统计不同防御策略下正常响应和延迟响应时延分布情况，继而得出不同策略情况下的检出率。实验结果如图 6 所示，相较于原防御策略，所提的三种防御策略有效地扰乱了攻击者的利用样本均值和方差的统计特征来传输信息，极大地增加了攻击者的攻击难度。

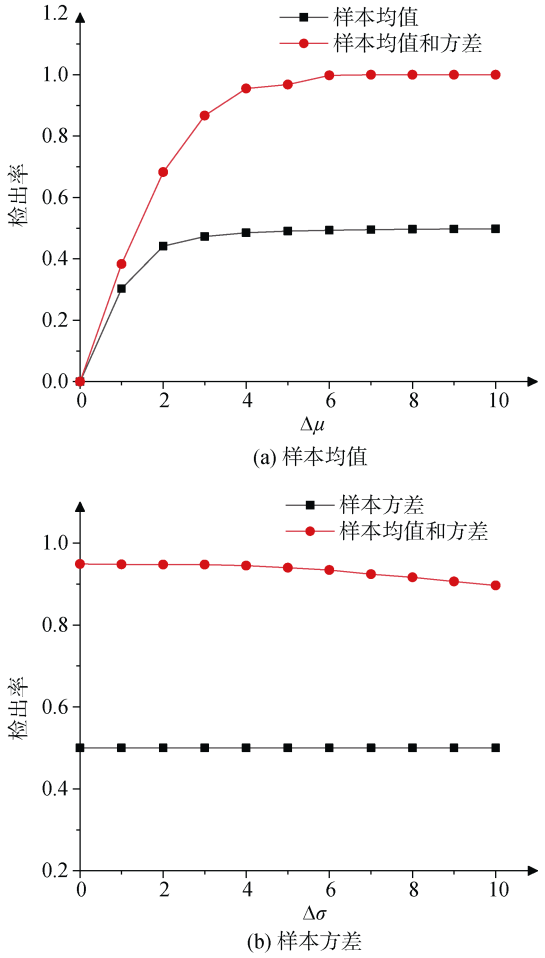


图 5 样本均值和方差对检出率的影响

Figure 5 The influence of sample mean and variance on detection rate

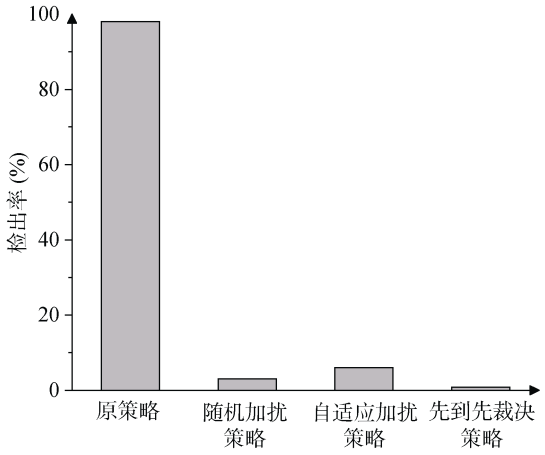


图 6 不同防御策略下的检出率

Figure 6 Detection rate under different defense strategies

5.3 系统响应时延

本节设计一个简单的隐蔽信道攻击实验场景来模拟隐蔽信道攻击的具体实施过程, 同时验证所提三种防御策略的可行性和有效性。在攻击者掌握一定系统运行信息的条件下, 设置客户端等间隔地对系统某一服务发起请求。为了模拟攻击者控制少部分变体服务器响应时延, 在攻击者正式实施隐蔽信道攻击时控制变体服务器对客户端第 2、4、6、8、10 次请求增加 20ms 响应延迟, 此时攻击者在客户端捕获的响应时延数据如图 7 所示。攻击者通过数模信号转化可以解码出变体侧预传输的信息(如延迟响应时攻击者视为传输高电平 1, 正常响应时攻击者视为传输低电平 0), 继而泄露相关信息。

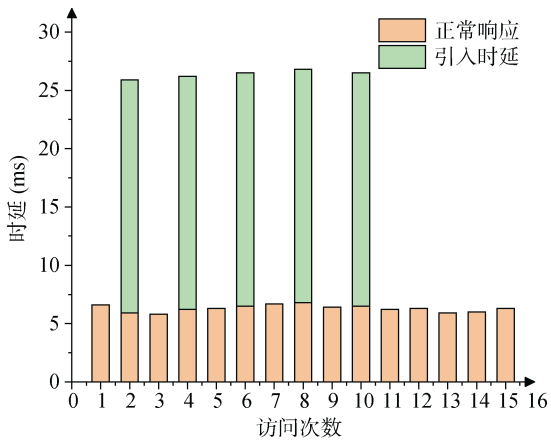


图 7 系统响应时延

Figure 7 System response delay

基于上述分析, 本节对代理服务器 Nginx 侧实现了三种防御策略来解决上述隐蔽信道攻击安全威胁, 实验结果如图 8 所示。在系统正常响应的情况下,

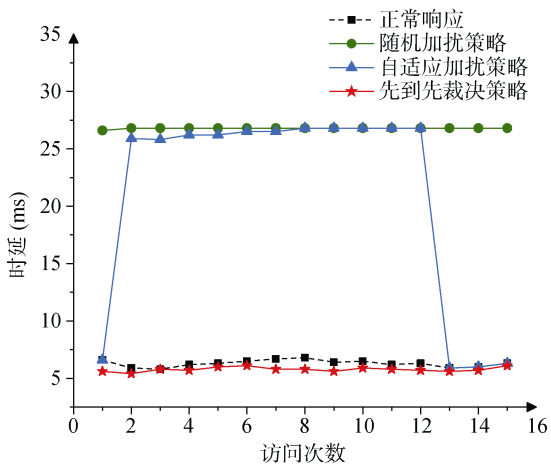


图 8 不同防御策略下系统响应时延

Figure 8 System response delay under different defense strategies

系统平均响应时延为 6.4ms。随机加扰策略对用户请求引入大量时延使得所有请求的响应时延满足相同的特征分布, 此时系统平均响应时延为 26.8ms。自适应加扰策略通过维护自身响应时延状态集, 动态地调整裁决策略来平衡系统安全性和系统运行效率, 此时系统平均响应时延为 19.7ms。先到先裁决策略在满足基本大数裁决条件时就立即进行裁决输出, 此时系统平均响应时延为 5.7ms。相较于正常系统响应, 先到先裁决策略降低了系统 10%的响应时延, 提升了系统运行效率。

5.4 性能测试

本节在基于 Nginx 开发的原型系统上进行了广泛的测试, 对比了原防御策略与所提的 3 种应对策略在不同响应包体大小、请求并发数和变体数目下整个系统的性能, 以验证所提防御策略的可行性与有效性。系统性能是计算机系统优劣的重要衡量指标, 这里主要从系统响应时延、吞吐量和 CPU 利用率 3 个常用指标来衡量 N 变体系统在不同防御策略下的性能。

5.4.1 响应包体大小

本节在异构变体服务器侧部署了相同的 Web 应用, 每个 Web 应用上部署 2~10KB 的静态资源来研究响应包体大小对系统性能的影响, 同时对比了 3 种防御策略下系统的性能。

设置客户端以每秒 100 并发访问应用服务、同时总请求数设置为 5000 的情况下, 对比了 N 变体系统中原防御策略和 3 种优化策略在不同响应包体大小情况下系统的性能。实验结果如图 9 所示, 从响应时延、吞吐量和 CPU 利用率 3 个指标详细对比了系统的性能。随着响应包体大小的增加, 请求的处理和传输时延会增加, 系统响应时延和 CPU 利用率也会相应增加, 系统吞吐量反而会降低。随机加扰策略记录了系统最长响应时延, 对所有业务流量引入一定的延迟, 因此系统响应时延最长, 吞吐量和 CPU 利用率较低。自适应加扰策略平衡了系统安全性和运行效率, 防御效果优于自适应加扰策略。先到先裁决策略在满足基本裁决条件时就进行裁决和输出, 系统响应时延最低, 吞吐量和 CPU 利用率较高, 防御效果最优。且相较于原策略, 当代理服务器使用先到先裁决策略时, 系统平均响应时延降低了 10%, 吞吐量提升了 18%, CPU 利用率提升了 3%。

5.4.2 请求并发量

本节主要对比不同并发量条件下 3 种防御策略下系统的性能。配置客户端访问 2KB 的 Web 资源, 客户端并发量设置为 200~1000, 请求过程持续 5 min,

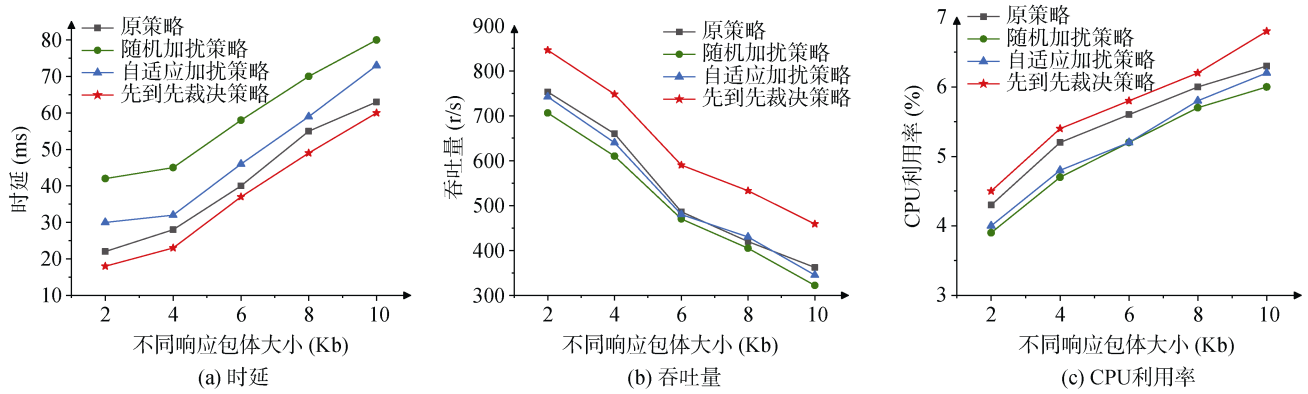


图 9　不同响应包体大小下系统性能对比

Figure 9　System performance comparison under different response packet sizes

对比原防御策略和 3 种优化策略在不同并发量条件下系统的性能。实验结果如图 10 所示, 随着请求并发量的增加, 系统响应时延随之增加, 吞吐量先增加, 而后趋于稳定, CPU 利用率也随之增加。相较于原防御策略, 随机加扰策略和自适应加扰策略通过引入一定时延以扰乱攻击者分析流量, 使得攻击者离线分析阶段获得的先验知识无效。两种策略都消耗了一部分系统运行效率以保证系统安全, 防止系统隐私泄露。先到先裁决策略通过优化原有裁决算法以防御时延隐蔽信道攻击, 同时凭借着 Nginx 出色的性能, 先到先裁决策略能够有效地平衡系统安全性和系统运行效率。在考虑不同并发量的情况下, 相较于其余 3 种防御策略, 先到先裁决策略在时延、吞吐量和 CPU 利用率等性能上皆有一定的提升, 防御效果最优。

5.4.3　变体数目

N 变体系统中变体的数目对性能影响较大, 根据相对正确公理, 多个异构性变体在同一时刻出错的可能性极低。通常变体副本的数量越多, 系统的异构性越大, 裁决模块感知到异常的可能性越大, 系统安全性越高。因此本节研究变体数目对系统性能的影响是非常有意义的。

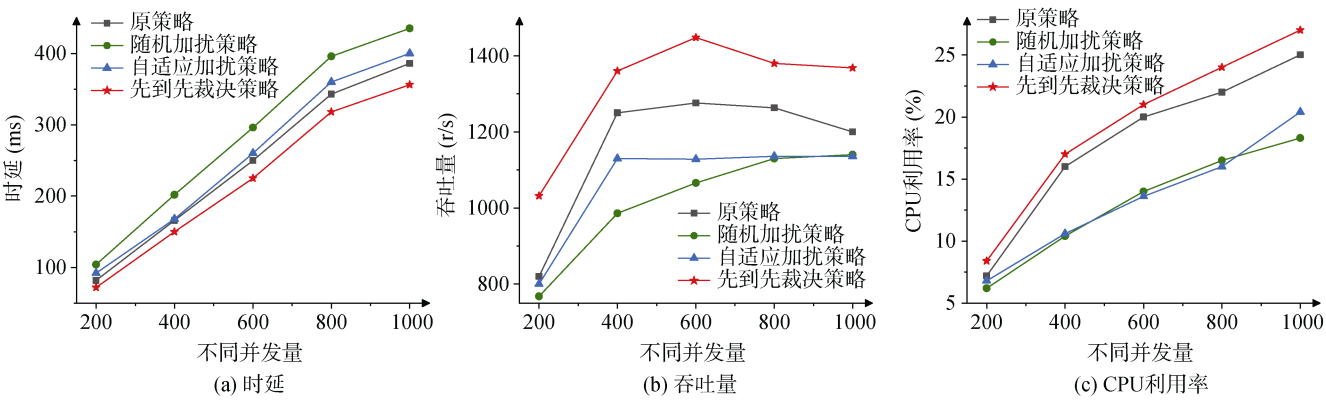


图 10　不同并发量条件下系统性能对比

Figure 10　System performance comparison under different concurrency conditions

设置服务器端静态资源大小为 2KB, 客户端以 1000 并发量对系统中该资源进行访问, 整个访问过程持续 5 min。实验结果如图 11 所示, 随着变体数目的增加, N 变体系统裁决模块的工作负载增加, 客户端请求的排队和处理时延相应增加, 系统响应时延和 CPU 利用率也随之增加。由于系统需要监视多个变体副本的响应输出, 单位时间内系统处理的请求数目减少, 系统吞吐量随着变体数目的增加而降低。

本节在基于 Nginx 开发的原型系统上进行大规模实验, 从响应时延、吞吐量和 CPU 利用率 3 个指标详细对比了所提 3 种应对策略在不同响应包大小、请求并发量和变体数目下的系统性能。实验结果证明了所提 3 种策略在防御基于响应时延隐蔽信道攻击安全威胁上的有效性, 同时验证了 3 种防御策略在真实网络环境中的可行性。先到先裁决策略

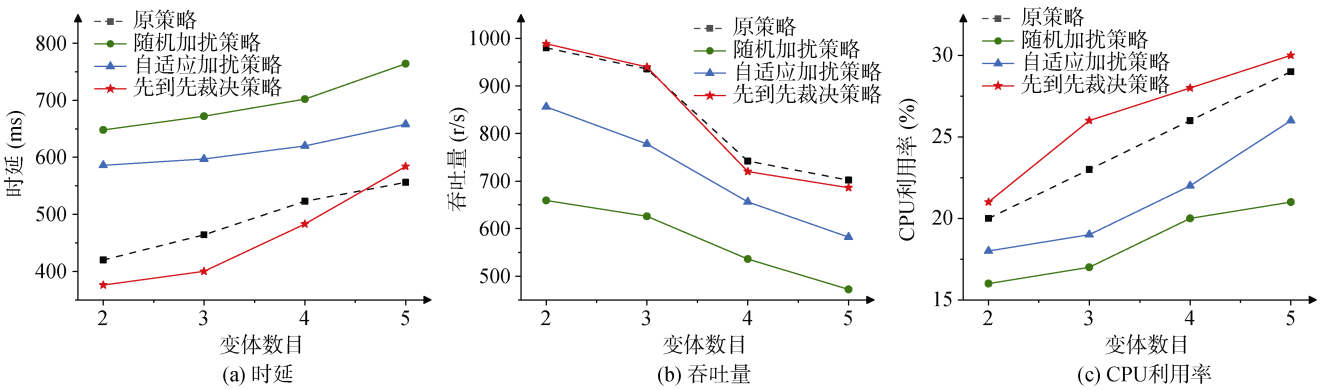


图 11　不同变体数目下系统性能对比

Figure 11　System performance comparison under different number of variants

充分考虑了攻击者攻击策略, 屏蔽了攻击者利用少部分变体地响应时延来控制系统的整体响应时延, 有效地保证了 N 变体系统的高安全性和高可靠性。值得一提的是, 先到先裁决策略在引入较少存储与裁决开销的情况, 提升了系统性能, 系统平均响应时延降低了 10%, 吞吐量提升了 18%, CPU 利用率提升了 3%。

6　结论

本文提出了一种面向 N 变体系统的时延隐蔽信道攻击方法并给出了 3 种应对策略。在所提攻击方法中, 攻击者首先对预传输信息进行编码, 通过有规律地控制一部分变体的响应时延来控制系统响应时延, 同时对捕获的系统数据流量进行分析, 利用样本均值和样本方差作为特征统计量来传输信息, 造成系统信息泄露。针对该攻击方法, 本文给出 3 种防御策略以应对此类威胁, 其中包括随机加扰策略、自适应加扰策略和先到先裁决策略。随机加扰策略通过引入网络延迟, 使得所有请求的响应时延具有相似的统计特征, 以扰乱攻击者进行流量分析。自适应性加扰策略通过维护多个变体服务器响应时延状态集, 根据流量特征动态调整裁决策略。先到先裁决策略通过优化裁决算法, 使得攻击者难以利用数据包响应时延差异特征来传输信息, 同时降低了系统响应时延, 提升了系统运行效率。实验首先证明了基于响应时延的隐蔽信道攻击方法的可行性和有效性, 然后基于构建的 Nginx 原型系统对 3 种防御策略进行测试。测试结果表明, 3 种防御策略都有效地扰乱了攻击者对流量的分析, 使得攻击者先验知识无效, 有效地防御了隐蔽信道攻击安全威胁, 保证了 N 变体系统的安全。但值得注意的是, 随机加扰策略和自适应策略会给系统响应引入较大的时延, 降低系统运行效率。先到先裁决策略可以从根本上防御基于响应时延的隐蔽信道攻击安全威胁, 且相较于其余几种防御, 先到先裁决策略的防御效果最优, 能够降低系统响应时延, 同时提升系统性能。

本文的最后, 我们提出了值得进一步挖掘的研究点和未来的研究方向:

(1) 本文仅考虑了响应基于时延的隐蔽信道攻击与防御, 值得注意的是, 隐蔽信道攻击的攻击形式、攻击特征和载体特征具有较大的不确定性, 防御者较难从根源上对此类型攻击进行防御, 且一般都是“补丁式”防御。在未来的工作中, 可以从攻击者的角度研究隐蔽信道攻击中如何构造隐蔽信道以及如何通过隐蔽信道传输信息。

(2) 本文设计了 3 种防御策略以应对基于时延的隐蔽信道攻击, 并开发了基于 Nginx 的原型系统, 在真实网络环境中, 需要考虑从用户需求, 动态调整防御策略。未来可以考虑研究基于安全服务级别协议(Service Level Agreement, SLA)驱动的 N 变体系统动态防御架构。

(3) 本文仅考虑到了 N 变体系统中攻击者利用响应时延时序特征来传输信息, 造成裁决模块逃逸。值得注意的是, 这种基于时延隐蔽信道攻击同样适用于具有裁决模块的冗余架构系统中, 如拟态防御系统[50]等。未来可以结合拟态架构基本特征, 针对拟态系统中可能存在的隐蔽信道攻击和多模裁决逃逸进行进一步研究。

参考文献

[1] Fischer-Hbner S, Berthold S. Privacy-Enhancing Technologies[M]. Computer and Information Security Handbook. Amsterdam: Elsevier, 2017: 759-778.

[2] Nguyen-Tuong A, Evans D, Knight J C, et al. Security through Redundant Data Diversity[C]. *2008 IEEE International Confer-*

ence on Dependable Systems and Networks With FTCS and DCC, 2008: 187-196.

[3] Hu H C, Wu J X, Wang Z P, et al. Mimic Defense: A Designed-in Cybersecurity Defense Framework[J]. *IET Information Security*, 2018, 12(3): 226-237.

[4] Cho J H, Sharma D P, Alavizadeh H, et al. Toward Proactive, Adaptive Defense: A Survey on Moving Target Defense[J]. *IEEE Communications Surveys & Tutorials*, 2020, 22(1): 709-745.

[5] Voulimeneas A, Song D, Parzefall F, et al. Distributed Heterogeneous N-Variant Execution[M]. Springer, Cham, 2020.

[6] Cox B, Evans D, Filipi A, et al. N-Variant Systems: A Secretless Framework for Security through Diversity[C]. *The 15th conference on USENIX Security Symposium - Volume 15*, 2006: 105-120

[7] Heda Y, Shah R. Covert Channel Design and Detection Techniques: A Survey[C]. *2015 IEEE International Conference on Electronics, Computing and Communication Technologies*, 2015: 1-6.

[8] Wu Z Y, Xu Z, Wang H N. Whispers in the Hyper-Space: High-Bandwidth and Reliable Covert Channel Attacks Inside the Cloud[J]. *IEEE/ACM Transactions on Networking*, 2015, 23(2): 603-615.

[9] Devanathan J and Prince chelladurai S. Preventing Side Channel Attack in Web Based Applications[J]. Aust. J. Basic & Appl. Sci., 2015, 9(21): 109-114.

[10] Liu W M, Wang L Y, Ren K, et al. Background Knowledge-Resistant Traffic Padding for Preserving User Privacy in Web-Based Applications[C]. *2013 IEEE 5th International Conference on Cloud Computing Technology and Science*, 2013: 679-686.

[11] Wang C, Wang X L, Lü Y R, et al. Categorization of Covert Channels and Its Application in Threat Restriction Techniques[J]. *Journal of Software*, 2020, 31(1): 228-245.
(王翀, 王秀利, 吕荫润, 等. 隐蔽信道新型分类方法与威胁限制策略[J]. *软件学报*, 2020, 31(1): 228-245.)

[12] Wu J. Cyberspace mimic defense[M]. Springer International Publishing, 2020.

[13] Duan J, Hamlen K W, Ferrell B. Better Late than Never: An N-Variant Framework of Verification for Java Source Code on CPU x GPU Hybrid Platform[C]. *The 28th International Symposium on High-Performance Parallel and Distributed Computing*, 2019: 207-218.

[14] Tan L, Krings A. A Hierarchical Formal Framework for Adaptive N-Variant Programs in Multi-Core Systems[C]. *2010 IEEE 30th International Conference on Distributed Computing Systems Workshops*, 2010: 7-12.

[15] Zhang Z, Wang L Q, Li W C. Research on Formal Model for an Information System's Attack Surface with Dissimilar Redundant Architecture[J]. *Journal on Communications*, 2018, 39(S2): 223-230.
(张铮, 王立群, 李卫超. 面向非相似余度信息系统的攻击面模型[J]. *通信学报*, 2018, 39(S2): 223-230.)

[16] Szefer J. Survey of Microarchitectural Side and Covert Channels, Attacks, and Defenses[J]. *Journal of Hardware and Systems Security*, 2019, 3(3): 219-234.

[17] Liu F F, Yarom Y, Ge Q, et al. Last-Level Cache Side-Channel Attacks are Practical[C]. *2015 IEEE Symposium on Security and Privacy*, 2015: 605-622.

[18] Brumley B. Covert Timing Channels, Caching, and Cryptography[M]. Aalto University, 2011.

[19] Luo Y, Luo W, Sun X N, et al. Whispers between the Containers: High-Capacity Covert Channel Attacks in Docker[C]. *2016 IEEE Trustcom/BigDataSE/ISPA*, 2016: 630-637.

[20] Kadloor S, Kiyavash N, Venkitasubramaniam P. Mitigating Timing Side Channel in Shared Schedulers[J]. *IEEE/ACM Transactions on Networking*, 2016, 24(3): 1562-1573.

[21] Wu P L, Liu K E, Zheng K, et al. A Road Network Modeling Method for Map Matching on Lightweight Mobile Devices[J]. *Distributed and Parallel Databases*, 2015, 33(2): 145-164.

[22] Lyu Y D, Mishra P. A Survey of Side-Channel Attacks on Caches and Countermeasures[J]. *Journal of Hardware and Systems Security*, 2018, 2(1): 33-50.

[23] Mirzargar S S, Stojilović M. Physical Side-Channel Attacks and Covert Communication on FPGAs: A Survey[C]. *2019 29th International Conference on Field Programmable Logic and Applications*, 2019: 202-210.

[24] Wang A, Ge J, Shang N, et al. Practical Cases of Side-Channel Analysis[J]. *Journal of Cryptologic Research*, 2018, 5(4): 383-398.
(王安, 葛婧, 商宁, 等. 侧信道分析实用案例概述[J]. *密码学报*, 2018, 5(4): 383-398.)

[25] Liu A Y, Chen J, Yang L. Real-Time Detection of Covert Channels in Highly Virtualized Environments[M]. Critical Infrastructure Protection V. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011: 151-164.

[26] Wang J C, Lee H M, Chen C W, et al. Estimating Intent-Based Covert Channel Bandwidth by Time Series Decomposition Analysis in Android Platform[C]. *2017 IEEE Conference on Application, Information and Network Security*, 2017: 31-36.

[27] A.K, Akshaya, et al. Preventing Side-Channel Leaks in Web Traffic Using SVSD[J]. International Journal of Scientific Research in Science, 2015, 1(2): 16-19.

[28] Ling Z, Luo J Z, Yu W, et al. A New Cell Counter Based Attack Against Tor[C]. *The 16th ACM conference on Computer and communications security*, 2009: 578-589.

[29] Kadloor S, Gong X, Kiyavash N, et al. Low-Cost Side Channel Remote Traffic Analysis Attack in Packet Networks[C]. *2010 IEEE International Conference on Communications*, 2010: 1-5.

[30] Gong X, Kiyavash N. Timing Side Channels for Traffic Analysis[C]. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013: 8697-8701.

[31] Li K, Li H, Zhu H S, et al. Side-Channel Information Leakage of Traffic Data in Instant Messaging[C]. *2019 IEEE 38th International Performance Computing and Communications Conference*, 2019: 1-8.

[32] Fei X R, Xie Y, Tang S S, et al. Identifying Click-Requests for the Network-Side through Traffic Behavior[J]. *Journal of Network and Computer Applications*, 2021, 173: 102872.

[33] Ling Z, Luo J Z, Zhang Y, et al. A Novel Network Delay Based Side-Channel Attack: Modeling and Defense[J]. *2012 Proceedings IEEE INFOCOM*, 2012: 2390-2398.

[34] Mehta A, Alzayat M, de Viti R, et al. Pacer: Network Side-Channel

Mitigation in the Cloud[EB/OL]. 2019: arXiv: 1908.11568[cs.CR]. https://arxiv.org/abs/1908.11568

[35] Stergiopoulos G, Talavari A, Bitsikas E, et al. Automatic Detection of Various Malicious Traffic Using Side Channel Features on TCP Packets[C]. *Computer Security*, 2018: 346-362.

[36] Sabbagh M, Fei Y S, Wahl T, et al. SCADET: A Side-Channel Attack Detection Tool for Tracking Prime-Probe[C]. *2018 IEEE/ACM International Conference on Computer-Aided Design*, 2018: 1-8.

[37] Li H D, Zhang F Q, Yu L, et al. Towards Efficient Traffic Monitoring for Science DMZ with Side-Channel Based Traffic Winnowing[C]. *The 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*, 2018: 55-58.

[38] Li L, Zhai Z D. Web Security Enhancement Scheme Based on Web Application Firewall[J]. *Computer Engineering and Applications*, 2011, 47(25): 104-106.
(李莉, 翟征德. 一种基于 Web 应用防火墙的主动安全加固方案[J]. *计算机工程与应用*, 2011, 47(25): 104-106.)

[39] Hu Y R, Chen X S, Wang J F, et al. Anomalous Traffic Detection Based on Traffic Behavior Characteristics[J]. *Netinfo Security*, 2016(11): 45-51.
(胡洋瑞, 陈兴蜀, 王俊峰, 等. 基于流量行为特征的异常流量检测[J]. *信息网络安全*, 2016(11): 45-51.)

[40] Ullrich J, Zseby T, Fabini J, et al. Network-Based Secret Communication in Clouds: A Survey[J]. *IEEE Communications Surveys & Tutorials*, 2017, 19(2): 1112-1144.

[41] Levitin G. Optimal Structure of Fault-Tolerant Software Systems[J]. *Reliability Engineering & System Safety*, 2005, 89(3): 286-295.

[42] Gersting J L, Nist R L, Roberts D B, et al. A Comparison of Voting Algorithms for N-Version Programming[C]. *The Twenty-Fourth Annual Hawaii International Conference on System Sciences*, 1991: 253-262.

[43] Latif-Shabgahi G, Bennett S. Adaptive Majority Voter: A Novel Voting Algorithm for Real-Time Fault-Tolerant Control Systems[C]. *The 25th EUROMICRO Conference. Informatics: Theory and Practice for the New Millennium*, 1999: 113-120.

[44] Leung Y W. Maximum Likelihood Voting for Fault-Tolerant Software with Finite Output-Space[J]. *IEEE Transactions on Reliability*, 1995, 44(3): 419-427.

[45] Latif-Shabgahi G, Bass J M, Bennett S. History-Based Weighted Average Voter: A Novel Software Voting Algorithm for Fault-Tolerant Computer Systems[C]. *The Ninth Euromicro Workshop on Parallel and Distributed Processing*, 2001: 402-409.

[46] Polinsky I, Martin K, Enck W, et al. N-m-Variant Systems: Adversarial-Resistant Software Rejuvenation for Cloud-Based Web Applications[C]. *The Tenth ACM Conference on Data and Application Security and Privacy*, 2020: 235-246.

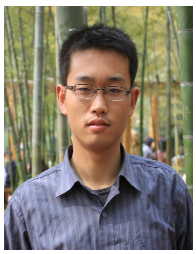
[47] Lindsay B G. Efficiency Versus Robustness: The Case for Minimum Hellinger Distance and Related Methods[J]. *The Annals of Statistics*, 1994, 22(2): 1081-1114.

[48] Adobe flash player. https://www.flash.cn/, 2021.

[49] Tcpdump and Libpcap. https://www.tcpdump.org/, 2021.

[50] Luo X G, Tong Q, Zhang Z, et al. Mimic Defense Technology[J]. *Engineering Sciences*, 2016, 18(6): 69-73.
(罗兴国, 仝青, 张铮, 等. 拟态防御技术[J]. *中国工程科学*, 2016, 18(6): 69-73.)

曾威 于 2019 年在中南大学信息安全专业获得学士学位。现在解放军信息工程大学网络空间安全专业攻读硕士学位。研究领域为网络主动防御。Email: zengwei19970605@163.com



霍树民 解放军信息工程大学信息技术研究所副研究员, 博士。研究领域为网络空间安全。Email: huoshumin123@163.com

扈红超 国家数字交换系统工程技术研究中心研究员。主要研究方向为云计算和网络安全。Email: 13633833568@139.com

周大成 于 2018 年在上海交通大学电子科学与技术专业获得学士学位。现为战略支援部队信息工程大学信息与通信工程专业博士生。研究兴趣包括网络安全、云计算。Email: bigchengz@163.com