

基于流量分析的软件升级漏洞自动检测方法

腾金辉, 光焱, 舒辉, 张冰

(战略支援部队信息工程大学, 河南 郑州 450001)

摘要: 软件升级过程中, 缺乏对升级信息或升级包的认证可能会导致基于中间人攻击的远程代码执行漏洞。为此, 提出一种升级漏洞自动检测方法。该方法通过提取升级过程中的网络流量, 对升级机制自动画像, 将其与漏洞特征向量匹配, 预判升级漏洞; 在模拟验证环境中, 利用画像信息实施中间人攻击, 验证检测结果。基于该方法设计了升级漏洞自动分析与验证系统, 对 184 个 Windows 应用软件样本进行测试, 检测出 117 个样本的升级漏洞, 证明了本方法的有效性。

关键词: 软件升级; 流量分析; 漏洞检测; 自动化分析与验证

中图分类号: TP309.5

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2020004

Automatic detection method of software upgrade vulnerability based on network traffic analysis

TENG Jinhui, GUANG Yan, SHU Hui, ZHANG Bing

Strategic Support Force Information Engineering University, Zhengzhou 450001, China

Abstract: During the software upgrade process, the lack of authentication for upgrade information or packages can lead to remote code execution vulnerabilities based on man-in-the-middle attack. An automatic detection method for upgrading vulnerabilities was proposed. The method described the upgrade mechanism by extracting the network traffic during the upgrade process, then matched it with the vulnerability feature vector to anticipate upgrading vulnerabilities. In a validation environment, the man-in-the-middle attack using the portrait information was carried out to verify the detection results. In addition, an automatic vulnerability analysis and verification system based on this method was designed. 184 Windows applications samples was test and 117 upgrade vulnerabilities were detected in these samples, which proved validity of the method.

Key words: software upgrade, network traffic analysis, vulnerability detection, automated analysis and validation

收稿日期: 2019-05-14; 修回日期: 2019-08-12

通信作者: 腾金辉, 1013787991@qq.com

基金项目: 国家重点研发计划基金资助项目 (No.2018YFB08011601)

Foundation Item: The National Key R & D Program of China(No.2018YFB08011601)

论文引用格式: 腾金辉, 光焱, 舒辉, 等. 基于流量分析的软件升级漏洞自动检测方法[J]. 网络与信息安全学报, 2020, 6(1): 94-108.

TENG J H, GUANG Y, SHU H, et al. Automatic detection method of software upgrade vulnerability based on network traffic analysis [J]. Chinese Journal of Network and Information Security, 2020, 6(1): 94-108.

1 引言

在整个软件生命周期中,软件的维护升级工作是最重要的环节^[1],通过升级至新版本,可以达到增添软件功能、修复已知错误与漏洞、提升用户体验等目的。然而,从软件安全性角度考虑,一方面,升级过程通常包括可执行文件的下载与自动执行;另一方面,用户对升级过程中下载的可执行文件,往往有着远超一般可执行文件的信任程度。因此,若升级过程存在漏洞,由此引发的升级文件被篡改或替换,极易导致用户遭受远程代码自动执行攻击,对用户系统的安全性造成严重损害。此外,基于软件升级的攻击方法是多种攻击手段的综合利用,如中间人攻击、流量劫持、软件逆向等,因而单一的防御方法不能有效对抗这种攻击,较之平常的攻击手段,其成功率更高,更隐蔽^[2]。升级过程产生的流量也能更顺利地通过基于流量检测方法的防火墙、杀毒软件、IDS 等的防御监控体系,即使用户终端建立了较为完善的边界安全防护机制、用户具备较高的安全意识水平,也依然难以填补这一缺陷。

基于软件升级漏洞的攻击有较多案例。2017年7月,中国山东、山西、浙江、福建等省爆发大规模软件升级劫持事件,大量计算机在升级过程中感染木马病毒;2017年8月,攻击者通过劫持乌克兰流行专用会计软件 M.E.Doc 的升级服务器,向用户推送包含 NotPetya 病毒的升级包,对政府、银行、电力系统、通信系统等实施攻击;2018年12月,一款木马通过“驱动人生”的升级通道,利用“永恒之蓝”漏洞攻击了超过10万用户;2019年3月,卡巴斯基实验室曝光华硕(ASUS)的软件升级服务器遭到入侵,攻击者通过软件升级,在用户电脑中安装恶意程序。上述安全事件都反映出软件升级机制漏洞会导致巨大安全风险。

目前,已有的针对软件升级漏洞的相关研究主要围绕升级漏洞利用、检测以及安全防护展开。在升级漏洞利用方面,2017年,360天擎团队发布关于国内政企类软件安全性的分析报告,其中包含了针对软件升级的相关漏洞的案例以及攻击原理分析;2015年,彭先觉等^[2]提出一种 HTTP

协议通信校验值误用类型软件升级漏洞的攻击方法,并对某团购网站的旧版软件进行了实际攻击测试,验证了该攻击的可行性。在升级漏洞检测方面,2015年,傅建明等^[3]提出一种静态分析与动态分析相结合的针对杀毒软件升级流程安全性分析方法。2017年,Adam K 提出了一种基于特征信息匹配的升级漏洞自动检测方法,能够捕获针对软件升级过程的攻击。在安全防御方面,2018年,周振飞^[4]从机理与防御两个方面,对软件供应链污染攻击进行研究,并提出了针对特定种类攻击的解决方案;2015年,Catuogno 等^[5]提出一种升级框架,不同于已有的升级框架(如 WinGup)使用 SSL 协议保证升级过程的安全性,该框架基于非交互式密钥进行升级,同时依赖于静态属性策略的软件安装与升级框架,并对该方案进行了性能分析。

上述工作从多个方面揭示了软件升级漏洞的原理以及相应的检测方法。然而,这些工作存在两方面问题:一是缺乏清晰明确的升级漏洞分类特征描述模型,影响漏洞检测的范围和准确性;二是漏洞分析与检测的自动化程度较低,难以满足大批量样本条件下的检测要求。

为解决上述问题,本文提出一种基于流量分析的软件升级漏洞自动检测方法。该方法通过对软件升级过程中网络通信流量特征的分析,建立面向流量特征的升级漏洞特征分类模型;综合使用字符串模糊匹配和频率加权比对等方法,实现对升级过程网络数据流特征的提取与分析,实现对升级过程的自动画像,进而通过特征向量匹配,实现对多种类型升级漏洞的自动化检测。同时,设计漏洞自动化验证平台,通过模拟攻击的方法,对漏洞检测的结果进行自动验证,提高检测结果的准确性。

2 升级漏洞特征模型

2.1 升级漏洞的分类

软件在线升级的内容本身,无论是程序组件还是完整的升级包,都可视为公开的数据和资源,无须进行保密。软件升级过程的漏洞,主要体现在对升级服务器和升级包的认证机制漏洞,主要包括对升级服务器身份认证缺失,以及对升级包

的真实性、完整性与新鲜性认证的缺失等。从升级漏洞的具体形式来看,主要体现在升级过程的3个不同阶段。分别是发起阶段,客户端与服务端交互,获取相关配置信息与升级包URL;下载阶段,客户端根据URL下载升级包;安装阶段,安装升级包。涉及网络数据的主要为前两个阶段,根据这两个阶段网络数据特征,对升级漏洞进行分类,如表1所示。

(1) HTTP 协议通信类漏洞

对于无防护,在流量侧可直接通过中间人的方式劫持并篡改发起阶段的数据包中的升级包下载链接,将其替换为恶意代码的下载链接从而实现恶意代码的下载,并在安装阶段实现远程代码与程序的自动执行,具体实例如CNTV央视影音客户端(测试版本4.4.0.0,受影响版本4.6.2.1及以下,CNNVD=201903-1283)。

对于校验误用,在流量侧可直接通过中间人的方式劫持并篡改发起阶段的数据包中的新版本下载链接以及对应的校验值,将其替换为恶意代码的下载链接以及对应的校验值,从而实现恶意代码的下载并在安装阶段实现远程代码与程序的自动执行,具体实例如优酷视频播放器(测试版本7.2.6.8110,受影响版本7.7.7.4191及以下,

CNVD-2018-06299)。

(2) HTTPS 协议通信类漏洞

对于HTTPS保护缺失,在流量侧可通过中间人的方式劫持篡改完成。对于发起阶段使用HTTP协议的,可劫持该阶段的配置信息并进行篡改。对于下载阶段可通过强制返回虚假重定向数据包,完成发起阶段或下载阶段的HTTPS协议绕过,实现恶意代码的下载,从而在安装阶段实现远程代码与程序的自动执行,具体实例如5K Player(加密传输配置信息,下载阶段使用HTTP协议,测试版本5.0,受影响版本5.7及以下),GOM AUDIO(HTTP协议传输HTTPS链接以及配置信息,测试版本2.3.35.5296,受影响版本2.3.38.5300及以下)。

对于HTTP重定向误用,在流量侧可分别劫持下载阶段或发起阶段的重定向的返回数据包并对重定向链接进行篡改,完成HTTPS协议绕过,实现恶意代码的下载,从而在安装阶段实现远程代码与程序的自动执行,具体实例如SweetHome3D(下载过程使用重定向,测试版本5.5,受影响版本5.7及以下),TunesKit Audiobook Converter(发起阶段使用重定向,测试版本3.0.3,受影响版本3.0.8及以下)。

表1 漏洞分类特征描述
Table 1 The description of vulnerability classification features

I级分类	II级分类	分类简述	阶段特征		
			发起阶段	下载阶段	安装阶段
HTTP 协议通信类漏洞	无防护	采用 HTTP 协议明文传输配置信息	配置信息传输采用 HTTP 协议明文/简单编码,其中无升级包校验信息	采用 HTTP 协议发起, TCP 协议下载	无消息校验机制,直接安装升级包
	校验误用	采用 HTTP 协议明文传输配置信息	配置信息传输采用 HTTP 协议明文/简单编码,其中包含升级包校验信息	采用 HTTP 协议发起, TCP 协议下载	安装升级包之前校验升级包校验值
HTTPS 协议通信类漏洞	HTTPS 保护缺失	发起阶段使用 HTTPS 协议, 下载阶段用 HTTP 协议	采用 HTTPS 协议传输配置信息,其中无升级包校验信息	采用 HTTP 协议发起或将 HTTPS 服务器降级提供 HTTP 服务发起, TCP 协议下载	无消息校验机制,直接安装升级包
		发起阶段使用 HTTP 协议下载阶段使用 HTTPS 协议	采用 HTTP 协议明文/简单编码传输含有 HTTPS 下载链接的配置信息	采用 HTTPS 协议发起, TCP 协议下载	消息校验机制可选,直接安装升级包
	重定向误用	下载阶段使用 HTTP 重定向	采用 HTTPS 协议传输配置信息,其中无升级包校验信息	采用 HTTP 重定向将新版程序的 HTTP 链接升级为 HTTPS 链接,随后使用 HTTPS 协议发起与下载	无消息校验机制,直接安装升级包
		发起阶段使用 HTTP 重定向	采用 HTTP 重定向将配置信息由 HTTP 链接升级为 HTTPS 链接	采用 HTTPS 协议发起, TCP 协议下载	消息校验机制可选,直接安装升级包

2.2 升级漏洞特征向量

为实现软件升级漏洞的自动化检测，首先需要对升级漏洞特征进行定义和描述。然后通过对实例分析总结，从升级过程的网络流量数据中，筛选出与升级漏洞检测相关的 22 种关键信息要素，将其取值对应的向量。作为升级漏洞的检测依据，即升级漏洞特征向量，信息要素按照升级的发起和下载两个阶段划分如表 2 所示。

将表 1 中的漏洞分类与表 2 中信息要素相结合，得到不同类型的升级漏洞特征向量，如表 3 所示。

3 升级漏洞的自动检测方法

基于漏洞特征向量，本节给出一种软件升级漏洞自动化检测方法，以关键信息要素的提取与分析为基础，按照“升级画像—漏洞检测—漏洞

验证—结果确认”的方法，实现升级漏洞自动分析与检测，流程如图 1 所示。

漏洞流程分为 4 步。

Step1 升级画像。结合文件操作记录与升级流量记录，对升级过程中与漏洞检测相关的关键数据包进行定位，提取其中的关键信息要素，构建描述升级过程与机制的“升级画像”。

Step2 漏洞检测。根据升级画像的向量描述，结合漏洞特征向量，进行基于向量匹配的漏洞检测，得到漏洞检测预结果。

Step3 漏洞验证。在模拟攻击环境下，根据检测结果进行验证形成最终的检测结论。该方法综合了理论分析与实际验证，有效确保了分析的准确性。

Step4 结果确认。根据模拟攻击结果对检测结论进行确认，并输出画像、检测结果、漏洞验证配置文件。

表 2 升级关键信息要素
Table 2 The key information elements during upgrade process

Name version		软件名 版本号
发 起 阶 段	protocol	发起阶段协议类型 (0: HTTP 1: HTTPS)
	request	发起升级请求的内容 (0: 无法提取 1: 可提取)
	redirect	是否使用重定向(0: 使用 1: 不使用)
	redirect_up	是否通过重定向进行协议升级 (0: 不升级 1: 升级)
	server_ip	升级配置信息服务器 IP 地址 (0: 无法提取 1: 可提取)
	server_port	升级配置信息服务器端口号 (0: 无法提取 1: 可提取)
	local_port	本地通信端口号 (0: 无法提取 1: 可提取)
	prog_name	升级程序 (或配置信息文件) 的文件名 (0: 无法提取 1: 可提取)
	file_url	升级程序 (或配置信息文件) 下载链接 (0: 无法提取 1: 可提取)
	up_file_url	重定向后配置信息文件下载链接*
下 载 阶 段	url_type	下载链接使用的协议类型 (0: HTTP 1: HTTPS)
	prog_verify	散列文件校验信息 (如 MD5、SHA-1)
	hash	散列文件校验信息 (如 MD5、SHA-1)
	crc	循环冗余文件校验信息 (如 CRC32)
	size	文件大小校验信息
	protocol	下载阶段协议类型 (0: HTTP 1: HTTPS)
	request	文件下载的请求数据包名称 (0: 无法提取 1: 可提取)
	redirect	是否使用重定向(0: 使用 1: 不使用)
	redirect_up	是否通过重定向进行协议升级 (0: 不升级 1: 升级)
	up_url	重定向后新版本程序下载链接*
	server_ip	升级文件下载服务器 IP 地址 (0: 无法提取 1: 可提取)
	server_port	升级文件下载服务器端口号 (0: 无法提取 1: 可提取)
	local_port	本地通信端口号 (0: 无法提取 1: 可提取)

【注】*项目实际情况可为空，括号中内容为表 3 中漏洞特征向量的对应值。

表 3 升级漏洞特征向量
Table 3 The vulnerability feature vector during upgrade process

漏洞类型		无防护	校验误用	HTTPS 保护缺失 (下载阶段)	HTTPS 保护缺失 (发起阶段)	重定向误用 (下载阶段)	重定向误用 (发起阶段)
画像内容		对应向量值					
发 起 阶 段	protocol	0	0		0	*	1
	request	1	1		1	*	1
	redirect	*	*		0	*	1
	redirect_up	0	0		0	*	1
	server_ip	1	1		1	*	1
	server_port	1	1		1	*	1
	local_port	1	1		1	*	1
	prog_name	1	1		1	*	1
	file_url	1	1		1	*	1
	up_file_url	*	*		0	*	1
	url_type	0	0		1	*	1
	prog_verify	hash	0		*	*	*
下 载 阶 段		crc	0	*	*	*	*
		size	0	*	*	*	*
	protocol	0	0	0	1	0	*
	request	1	1	1		1	*
	redirect	*	*	0		1	*
	redirect_up	0	0	0		1	*
	up_url	*	*	0		1	*
	server_ip	1	1	1		1	*
	server_port	1	1	1		1	*
	local_port	1	1	1		1	*

【注】*项目根据实际情况可选（具体值可为空/0/1），空白为使用 HTTPS 协议无法提取具体值，值的说明参见表 2 括号中内容。

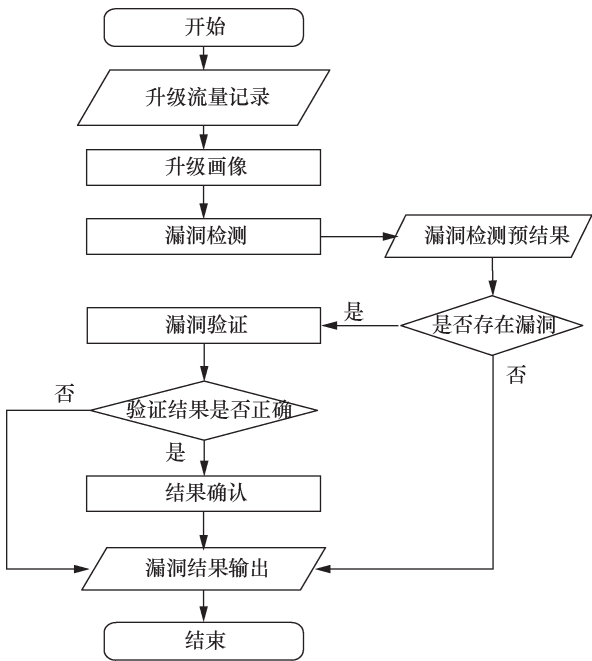


图 1 漏洞检测流程
Figure 1 The flow of vulnerability detection

下面按照流程分别介绍各个步骤所使用的具体方法。

3.1 升级流量记录

升级过程中软件客户端与升级服务器之间的网络通信数据中，包含关于升级方式、升级流程、升级安装包格式等一系列关键信息。为了对升级过程网络数据进行提取与分析，本文使用 tshark 进行实现。为便于对网络流量数据包的自动化分析，将流量记录以 json 格式保存，单个 json 格式流量包结构示例如图 2 所示。

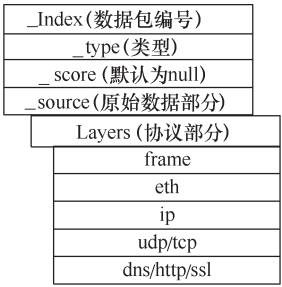


图 2 json 数据包结构
Figure 2 The structure diagram of json data packet

3.2 升级画像

升级画像所需的信息提取关键在于对发起阶

段和下载阶段的相关数据包进行准确定位。根据目标软件升级模式的不同,采用分别基于下载阶段信息和发起阶段信息两种信息定位和提取方法。

3.2.1 基于下载阶段信息的升级画像

该方法首先在流量记录的同时,通过文件监控,获取升级包文件名,并计算该文件对应的各种可能校验值(校验值类型见表2发起阶段prog_verify信息栏)。在此基础上,按照4个步骤实现基于下载阶段的升级画像,流程如图3所示。

Step1 通信协议扫描记录

对所有的通信数据包进行扫描,判断通信协议并记录,如果存在HTTP协议,进行Step2,否则结束。

Step2 下载阶段数据包定位

采用两种算法对下载阶段数据包进行定位

算法1 基于升级包文件名的精确搜索定位

输入 升级包文件名,升级流量数据包

输出 下载阶段数据包信息

算法步骤:

1) 对所有HTTP协议的流量数据包进行“GET”关键字匹配,提取下载请求及其对应的URL;

2) 在1)提取的所有下载请求及其对应的URL中,搜索文件监控到的升级包文件名,提取该文件下载请求的URL及其所在数据包的通信五元组并输出。

算法结束

算法2 基于文件类型的模糊匹配定位

输入 软件名,升级流量数据包

输出 下载阶段数据包信息

算法步骤:

1) 对所有HTTP协议的流量数据包进行“GET”关键字匹配,提取下载请求及其对应的URL;

2) 在1)提取的所有下载请求及其对应的

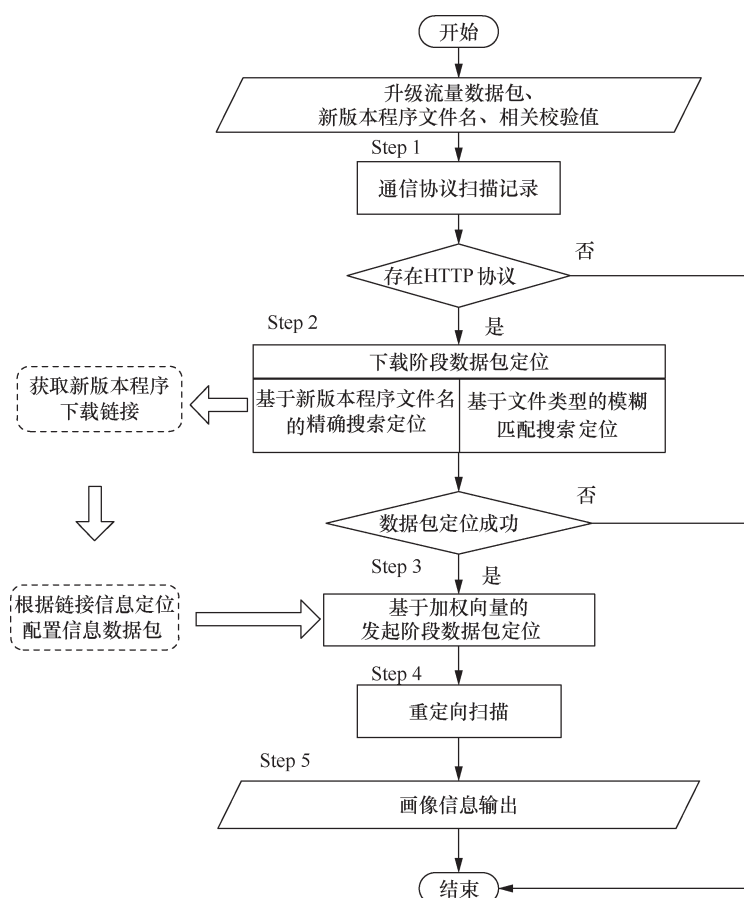


图3 基于下载阶段信息的升级画像流程

Figure 3 The flow of upgrade process based on download stage information

URL 中, 筛选 exe、msi、rar、zip 等类型的可执行文件的下载请求及其 URL;

3) 根据 Levenshtein Distance 算法^[6]分别计算所有下载请求 URL 中的文件名和测试软件名之间的相似度为:

对于 N_o (旧版本程序文件名)、 N_g (新下载程序的文件名), 字符串长度分别为 $|N_o|$ 、 $|N_g|$, 它们的 Levenshtein Distance 为

$$lev_{N_o, N_g}(i, j) = \begin{cases} \max(i, j), \min(i, j) = 0 \\ \min \begin{cases} lev_{N_o, N_g}(i-1, j) + 1 \\ lev_{N_o, N_g}(i, j-1) + 1 \\ lev_{N_o, N_g}(i-1, j-1) + temp \end{cases} \end{cases} \quad (1)$$

其中, 当 $N_{oi} = N_{gi}$ 时, $temp$ 为 0, 当 $N_{oi} \neq N_{gi}$ 时, $temp$ 为 1。

则文件名 N_o, N_g 的相似度为

$$Sim_{N_o, N_g} = 1 - \frac{lev_{N_o, N_g}(|N_o|, |N_g|)}{\max(|N_o|, |N_g|)} \quad (2)$$

4) 选取与旧版本程序文件名相似度最大的新下载程序文件名, 提取对应下载请求中的升级包 URL 及通信五元组, 并输出。

算法结束

Step3 发起阶段数据包定位

根据 Step2 提取到的升级包 URL, 采用基于加权向量的配置信息数据包定位算法对数据包进行定位。

算法 3 基于加权向量的发起阶段数据包定位

输入 升级包 URL, 升级流量数据包

输出 发起阶段数据包信息

算法步骤:

1) 对算法 2 提取的升级包 URL 按照协议名称: //服务器名称 (IP 地址) /路径/文件名进行拆分, 形成搜索字典;

2) 对 1) 中的搜索字典, 在所有 HTTP 协议的流量数据包的 http_file_data 数据段内容中进行搜索匹配, 统计搜索字典中各个字符串在该数据段中出现的次数, 形成各个数据包相应的字典统计向量;

3) 分别计算各个数据包对应的字典统计向

量加权, 权重及计算公式如表 4 所示。

表 4 权重信息
Table 4 The information of weight

字符串类型	主域名	路径	文件名
权重	0.2	0.2	0.6

T_m 、 T_p 、 T_n 分别为单个数据包中 http_file_data 数据段主域名、路径以及文件名字符串出现的次数, 加权向量和的计算如式(3)所示。

$$Sum = T_n \cdot (0.2 \cdot T_m + 0.2 \cdot T_p + 0.6 \cdot T_n) \quad (3)$$

上述公式中, 以 T_n 作为整体系数, 使文件名字符串的出现成为必要条件, 如果该数据包 http_file_data 数据段中不包含文件名, 则 $Sum=0$, 转至 4); 若 $Sum \neq 0$, 选取加权和最大的对应数据包, 并在该数据包 http_file_data 数据段中搜索相关的校验信息, 记录使用的相关校验方法以及校验值, 提取通信五元组并输出, 转至 5);

4) 在所有 HTTP 协议的流量数据包的 http_file_data 数据段内容中, 对以 http://或 https:// 开头, 以 .ini/.dat/.xml/.cfg 等配置文件类型扩展名结尾的 URL 进行搜索匹配, 匹配成功, 则截取其 URL, 通信五元组并输出, 转至 5);

5) 根据 3)、4) 定位数据包对应的通信五元组进行发起阶段数据包定位, 提取发起升级请求的内容及对应的升级请求 URL 并输出。

算法结束

Step4 重定向扫描

算法 4 重定向扫描

输入 发起阶段和下载阶段通信五元组, 升级流量数据包

输出 升级过程重定向信息

算法步骤:

1) 对所有 HTTP 协议的流量数据包进行 “301/302/303 Found” 匹配, 记录所有的重定向请求包以及回应包;

2) 在 1) 提取的所有重定向数据包中, 对 Step2 中发起阶段请求 URL 和 Step3 中下载阶段升级包 URL 进行扫描匹配, 记录匹配结果中重定向请求包以及响应包中的 URL、协议类型以及对应的升级阶段并输出。

算法结束

Step5 画像信息输出

将 Step1~Step4 输出的信息进行整理, 填充到表 2 升级画像的对应位置, 输出画像。

3.2.2 基于发起阶段信息的升级画像

该方法首先在流量记录的同时, 通过文件监控, 获取升级包文件名, 并计算该文件对应的各种可能校验值 (校验值类型见表 2 发起阶段 prog_verify 信息栏)。在此基础上, 按照 4 个步骤实现基于下载阶段的升级画像, 流程如图 4 所示。

Step1 通信协议判别

对所有的通信数据包进行扫描, 判断通信协议并记录, 如果存在 HTTP 协议, 则进行 Step2, 否则结束。

Step2 发起阶段数据包定位

采用两种算法对新版本程序下载数据包进行定位。

算法 5 基于升级包文件名的精确搜索定位

输入 升级包文件名, 升级流量数据包

输出 发起阶段数据包信息

算法步骤:

1) 在所有 HTTP 协议的流量数据包 of http_file_data 数据段内容中进行搜索匹配, 截取所有以 http://或 https://开头, 以.exe/.msi/.zip/.rar 等可执行文件后缀名结尾的 URL 字符串, 并记录;

2) 在 1)提取的结果中, 对文件监控获得的升级包文件名进行匹配, 如果匹配成功, 记录该 URL 及其所在数据包的通信五元组, 并在该数据段中搜索相关的校验信息, 记录使用的相关校验方法以及校验值并输出, 转至 4), 如果 URL 提取失败, 转至 3);

3) 在所有 HTTP 协议的流量数据包 of http_file_data 数据段内容中对以 http://或 https://

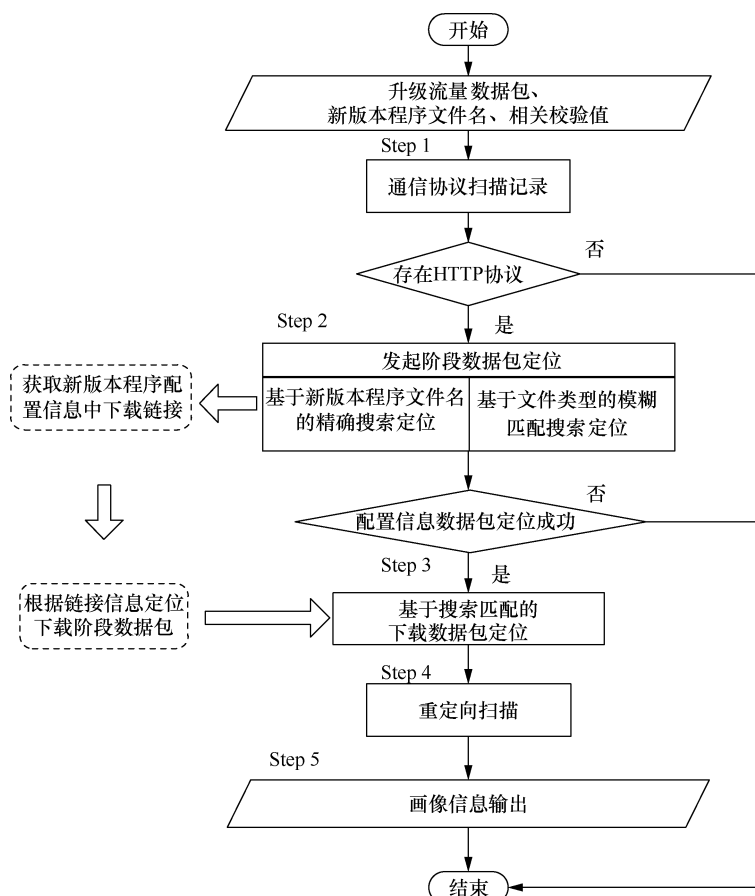


图 4 基于发起阶段信息的升级画像流程
Figure 4 The flow of upgrade process based on request stage information

开头,以.ini/.dat/.xml/.cfg 等配置文件类型扩展名结尾的 URL 进行搜索匹配,若匹配成功,则将其 URL 进行截取,提取其通信五元组并输出;

4) 根据 2)3)定位数据包对应的通信五元组进行发起阶段数据包定位,提取发起升级请求的内容及对应的升级请求 URL 并输出。

算法结束

算法 6 基于文件类型的模糊匹配定位

输入 程序名,升级流量数据包

输出 发起阶段数据包信息

算法步骤:

1) 在所有 HTTP 协议的流量数据包的 http_file_data 数据段内容中进行搜索匹配,截取数据包中所有以 http:// 或 https:// 开头,以.exe/.msi/.zip/.rar 等可执行文件后缀名结尾的字符串,并记录;

2) 在 1)提取的结果中,提取所有 URL 中文件名部分,根据 Levenshtein Distance 算法分别计算每个 URL 中文件名与旧版本程序文件名的相似度,选取超过阈值且最大的即为升级包 URL,并在该数据段中搜索相关的校验信息,记录使用的相关校验方法以及校验值,提取其所在数据包的通信五元组并输出,转至 4);如果 URL 提取失败,转至 3);

3) 在所有 HTTP 协议的流量数据包的 http_file_data 数据段内容中对以 http: //或 https:// 开头,以.ini/.dat/.xml/.cfg 等配置文件类型扩展名结尾的 URL 进行搜索匹配,匹配成功,则截取其 URL,提取其通信五元组并输出,转至 4);

4) 根据 2)、3)定位数据包对应的通信五元组进行发起阶段数据包定位,提取发起升级请求的内容及对应的升级请求 URL 并输出。

算法结束

Step3 下载阶段数据包定位

算法 7 基于搜索匹配的下载阶段数据包定位

输入 升级包 URL,升级流量数据包

输出 下载阶段数据包信息

算法步骤:

1) 对所有 HTTP 协议的包进行“GET”关键字匹配,提取下载请求及其对应的 URL

2) 在 1)提取的所有下载请求及其对应的 URL 中,搜索匹配 Step2 中提取的升级包 URL,记录匹配成功的下载请求数据包及其信五元组并输出。

算法结束

Step4 重定向扫描

方法同基于下载阶段信息的升级画像中对应部分。

Step5 画像信息输出

将 Step1—Step4 输出的信息进行整理,填充到表 2 升级画像的对应位置,输出画像。

以上两种算法相结合可针对表 1 中包含的四大类升级漏洞完成信息提取,对比如表 5 所示。

两种升级画像算法的核心都是获取升级包的 URL,并以此为线索完成对发起阶段和下载阶段数据包的定位,基于发起阶段信息的升级画像算法是先通过对 HTTP 协议数据包 http_file_data 数据段扫描获取升级包的 URL 及其配置信息,然后根据该 URL 完成对下载阶段数据包的定位,基于下载阶段信息的升级画像算法是先通过对 HTTP 协议数据包中所有的“GET”请求扫描获取升级包 URL,然后根据该升级包 URL 完成对发起阶段数据包的定位。

以央视影音客户端(无防护类,漏洞编号:CNNVD=201903-1283)为示例,升级画像算法在分析过程中提取的关键信息如红框所示,升级画像如图 5 所示。

表 5 算法能力分析
Table 5 The analysis of the capabilities for different algorithms

升级画像算法	无防护	散列误用	HTTPS 保护缺失		重定向误用	
			配置信息	下载阶段	配置信息	下载阶段
基于下载阶段信息	√	√	×	√	×	√
基于发起阶段信息	√	√	√	×	√	×

【注】“√”表示算法可提取该情况下画像信息,“×”表示算法不可提取该情况下画像信息。

No.	Time	Source	Protocol	Destination	Length	Info
175	3.073321	192.168.150.140	TCP	61.54.7.208	66	50583 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
176	3.104417	61.54.7.208	TCP	192.168.150.140	60	80 → 50583 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
177	3.104481	192.168.150.140	TCP	61.54.7.208	54	50583 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
178	3.173245	192.168.150.140	HTTP	61.54.7.208	228	GET /cbox/update_config.txt HTTP/1.1
179	3.174252	61.54.7.208	TCP	192.168.150.140	60	80 → 50583 [ACK] Seq=1 Ack=175 Win=64240 Len=0
180	3.221208	61.54.7.208	TCP	192.168.150.140	1514	80 → 50583 [ACK] Seq=1 Ack=175 Win=64240 Len=1460 [TCP segment of a reassembled PDU]
181	3.221209	61.54.7.208	HTTP	192.168.150.140	1027	HTTP/1.1 200 OK (text/plain)
182	3.221530	192.168.150.140	TCP	61.54.7.208	54	50583 → 80 [ACK] Seq=175 Ack=2434 Win=64240 Len=0
2584	21.768472	192.168.150.140	TCP	61.54.7.208	54	50583 → 80 [FIN, ACK] Seq=175 Ack=2434 Win=64240 Len=0
2585	21.769031	61.54.7.208	TCP	192.168.150.140	60	80 → 50583 [ACK] Seq=2434 Ack=176 Win=64239 Len=0
2586	21.808520	61.54.7.208	TCP	192.168.150.140	60	80 → 50583 [FIN, PSH, ACK] Seq=2434 Ack=176 Win=64239 Len=0

图 5 央视影音客户端升级发起阶段数据包

Figure 5 The data packet of request stage during CNTV client upgrade process

发起阶段：程序使用 HTTP 协议请求新版本的配置信息（编号 178），随后服务器使用 HTTP 协议对请求进行了回应（编号 181）。

回应的配置信息以明文方式传输，其中包含了新版本程序的下载链接以及相关校验值如图 6

所示。

下载阶段：程序根据配置信息中的 URL 使用 HTTP 协议进行升级包的下载（编号 2256）。央视影音客户端升级下载阶段数据包如图 7 所示。

央视影音客户端升级画像如图 8 所示。

```
{
  "result": {
    "update_url": "http://download.cntv.cn/cbox/v4/ysyy_v4.4.3.0_Setup_1004.exe",
    "version": [
      {
        "features": "",
        "type": "2",
        "version": "4.1.0.0"
      },
      {
        "features": "",
        "type": "2",
        "version": "4.2.0.0"
      }
    ]
  }
}
```

图 6 央视影音客户端升级发起阶段信息

Figure 6 The information of request stage during CNTV client upgrade process

No.	Time	Source	Protocol	Destination	Length	Info
2065	19.756001	192.168.150.140	TCP	113.207.4.71	66	50633 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
2248	20.109584	113.207.4.71	TCP	192.168.150.140	60	80 → 50633 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
2250	20.109667	192.168.150.140	TCP	113.207.4.71	54	50633 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
2256	20.174604	192.168.150.140	HTTP	113.207.4.71	354	GET /cbox/v4/ysyy_v4.4.3.0_Setup_1004.exe HTTP/1.1
2257	20.174829	113.207.4.71	TCP	192.168.150.140	60	80 → 50633 [ACK] Seq=1 Ack=301 Win=64240 Len=0
2263	20.388395	113.207.4.71	TCP	192.168.150.140	1514	80 → 50633 [ACK] Seq=1 Ack=301 Win=64240 Len=1460 [TCP segment of a reassembled PDU]
2264	20.388397	113.207.4.71	TCP	192.168.150.140	1514	80 → 50633 [ACK] Seq=1461 Ack=301 Win=64240 Len=1460 [TCP segment of a reassembled PDU]
2265	20.388397	113.207.4.71	TCP	192.168.150.140	606	80 → 50633 [PSH, ACK] Seq=2921 Ack=301 Win=64240 Len=552 [TCP segment of a reassembled PDU]

图 7 央视影音客户端升级下载阶段数据包

Figure 7 The data packet of download stage during CNTV client upgrade process

```
{
  "Process_start": {
    "protocol": "http",
    "request": "GET /cbox/update_config.txt HTTP/1.1",
    "redirect": "none",
    "redirect_up": "none",
    "server_ip": "61.54.7.208",
    "server_port": "80",
    "local_port": "50583",
    "prog_name": "ysyy_v4.4.3.0_Setup_1004.exe",
    "file_url": "http://download.cntv.cn/cbox/v4/ysyy_v4.4.3.0_Setup_1004.exe",
    "up_file_url": "none",
    "url_type": "http",
    "prog_verify": {
      "hash": "none",
      "crc": "none",
      "size": "none"
    }
  },
  "Process_download": {
    "protocol": "http",
    "request": "GET /cbox/v4/ysyy_v4.4.3.0_Setup_1004.exe HTTP/1.1",
    "redirect": "none",
    "redirect_up": "none",
    "up_url": "none",
    "server_ip": "113.207.4.71",
    "server_port": "80",
    "local_port": "50633"
  }
}
```

图 8 央视影音客户端升级画像

Figure 8 The portrait of CNTV client upgrade process

3.3 漏洞检测

本节采用基于自动验证的漏洞检测模式并提出基于画像信息升级向量匹配算法进行升级流程漏洞的自动化检测，其中漏洞检测算法具体实现如下。

算法8 漏洞检测算法

输入 升级流程预画像，漏洞检测规则集

输出 升级流程漏洞检测预结果

算法步骤：

- 1) 读入升级流程预画像，用于漏洞检测；
- 2) 对升级流程预画像各个阶段信息数据进行处理，形成升级流程向量形式的描述；
- 3) 根据漏洞检测规则集中的向量特征进行匹配，如果匹配成功，输出对应的匹配结果。

算法结束

以央视影音客户端为例，如图9所示，根据其画像，得到的升级流程向量为[0 1 0 0 1 1 1 1 1 0 0 0 0 0 1 0 0 0 1 1 1]，将该向量与表3中的漏洞特征向量进行匹配，可判定其漏洞类型为无防护漏洞。

3.4 漏洞验证

为提高检测结果的准确性，防止漏报、误报，构建模拟验证环境，对漏洞检测结果进行验证，整体思路如图10所示。

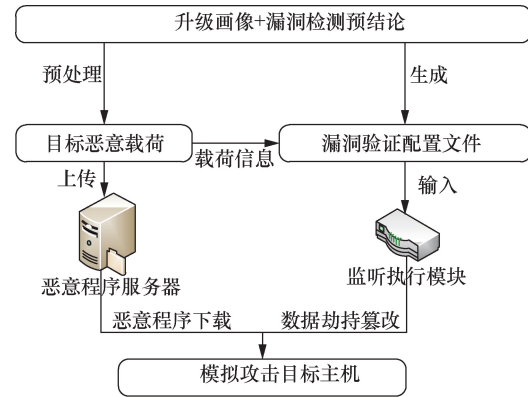


图10 漏洞验证思路

Figure 10 The method of vulnerability validation

(1) 目标载荷预处理

对目标载荷进行预处理，处理内容如下。

- 1) 根据画像检查升级包是否有压缩，是否特殊格式，若有则在不影响载荷功效的情况下将载荷调整为相同格式。
- 2) 根据画像检查升级包是否有校验，且读取是哪一类校验，调用 openssl 库函数的对应校验函数对步骤1)处理后的载荷计算校验值，将计算出的校验值替换原升级包的散列值。
- 3) 根据画像检查升级包是否有文件大小校验，计算经过步骤2)处理后的载荷大小，将结果替换原有画像的数值。



图9 央视影音漏洞检测向量

Figure 9 The vulnerability detection vectors of CNTV client

4) 根据画像信息更改恶意载荷名称与原升级包名称一致。

5) 恶意载荷上传至恶意程序服务器等待目标主机下载。

(2) 漏洞验证配置文件生成

验证配置信息中包含几项内容。

1) 升级关键数据定位信息

依据画像信息中的发起阶段和下载阶段相关的协议类型以及关键信息如 URL、检验信息等进行实时流量的劫持与定位。

2) 中间人攻击待篡改信息

根据漏洞检测预结论，针对不同类型的漏洞，结合画像信息和恶意载荷信息将实时升级过程中的配置文件数据包，重定向数据包中的 URL、相关

校验信息等关键字段信息篡改为恶意载荷的信息。

漏洞验证配置文件生成如图 11 所示。

(3) 模拟攻击目标主机

将漏洞验证配置文件输入到监听执行模块，恶意程序服务器启动，目标主机进行软件升级时进行关键数据劫持与篡改，促使目标主机下载恶意载荷，完成攻击。

以央视影音客户端为例，漏洞验证过程如图 12 所示。

根据漏洞检测预结论，指导提取画像中升级流量定位信息与待篡改关键信息，同时将恶意载荷上传至服务器并获取 URL，将以上信息整理形成漏洞验证配置文件并上传至中间人攻击监听执行模块，实施中间人攻击。

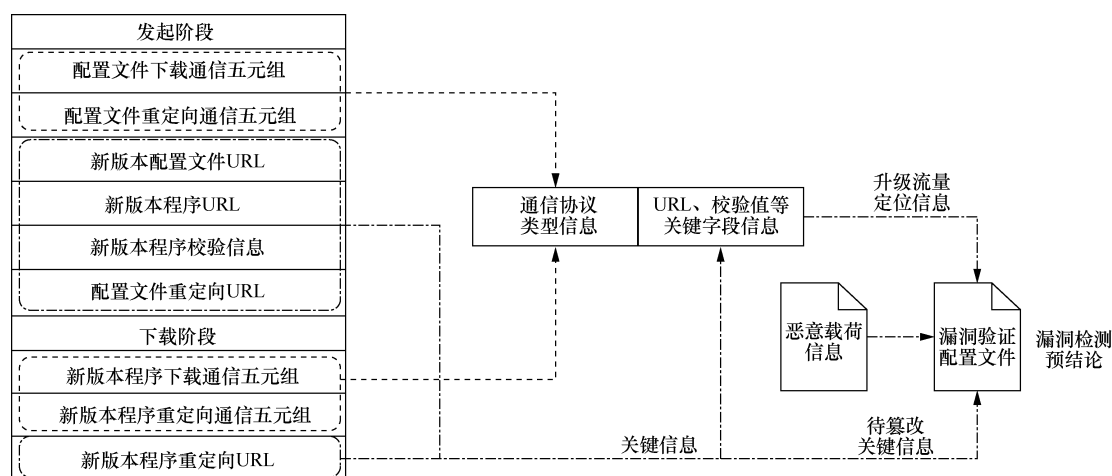


图 11 漏洞验证配置文件生成

Figure 11 The generation of configuration files for vulnerability validation



图 12 漏洞验证过程

Figure 12 The vulnerability validation process

3.5 结果确认

验证过程中，若模拟环境监测到目标主机完成了恶意载荷的下载，校验与自动执行则判定预检测结果正确，输出最终的升级漏洞检测结果。

4 方法实现与验证

为了验证上述检测方法的有效性，本文设计并实现了一个基于该方法的升级漏洞自动分析与验证系统。系统包含数据分析模块、漏洞检测模块与漏洞验证模块，分别实现升级流量记录与画像生成、升级过程漏洞检测，以及对检测结果验证的功能。利用该系统，对 184 个 Windows 应用软件样本进行了分析测试，证明了方法的有效性。

4.1 系统组成

设计升级漏洞自动分析与验证系统，系统架构如图 13 所示。

数据分析模块：由升级流量记录模块和升级流程画像生成模块两部分组成，该部分负责升级流程的网络数据信息的记录与分析，系统使用 tshark（wireshark 的命令行工具）对升级流程的所有网络数据流量进行记录，并将记录的 pcapng 格式数据包转为 json 格式数据包后，进行流量数据分析，形成升级流程画像。

漏洞检测模块：根据输入的画像信息，结合升级流程漏洞检测规则集，依据升级漏洞检测算法对升级漏洞进行检测并生成检测结论。

漏洞验证模块：验证检测结果的正确性，模块基于 Kali Ettercap 中间人攻击框架^[7]实现，根据画像信息和漏洞检测预结论生成指导中间人攻击的 Ettercap 配置文件，在攻击配置文件的指导下，在模拟环境中对软件升级过程进行攻击，根据验证结果，得到最终的漏洞检测结论。

4.2 测试环境与结果

升级漏洞自动化检测系统分为宿主机和虚拟机两部分，主程序 Updanalyze.exe 运行在宿主机上，负责将待测程序安装包、自动化分析脚本上传至虚拟机与从虚拟机回传结果。由 VMware Workstation 构建的虚拟化 Windows 操作系统负责安装待测程序、自动化分析升级过程与自动化验证。测试环境网络拓扑如图 14 所示。

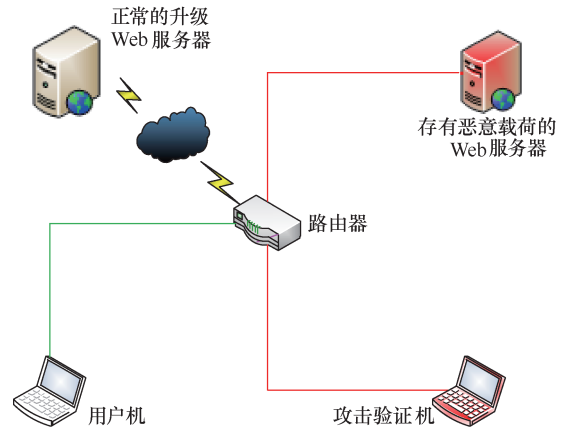


图 14 测试环境网络拓扑
Figure 14 The network topology diagram of testing environment

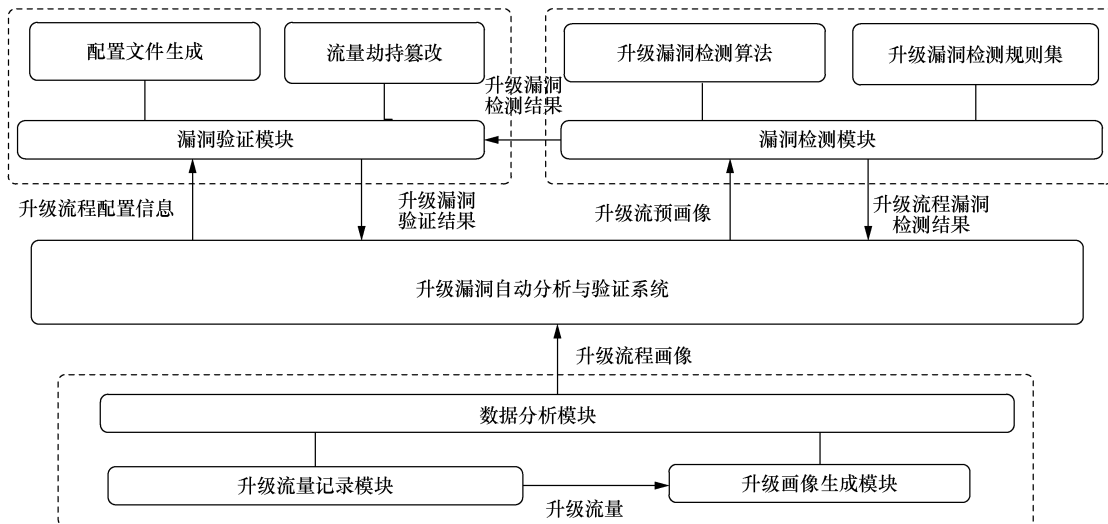


图 13 升级漏洞自动分析与验证系统架构
Figure 13 The structure of automatic vulnerability analysis and verification system

(1) 软件样本测试结果

对 12 类 184 款国内外的 Windows 应用软件进行测试, 包含有升级漏洞的样本数量为 117 个, 所占比例超过 60%, 涵盖了浏览器、输入法等 12 个软件功能类别, 根据漏洞类型和目标类别, 对测试结果进行统计。漏洞分布统计如图 15 所示, 不同软件类别含漏洞占比如图 16 所示。

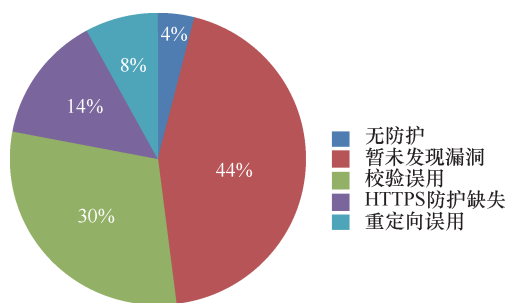


图 15 漏洞分布统计

Figure 15 The statistics of vulnerability distribution

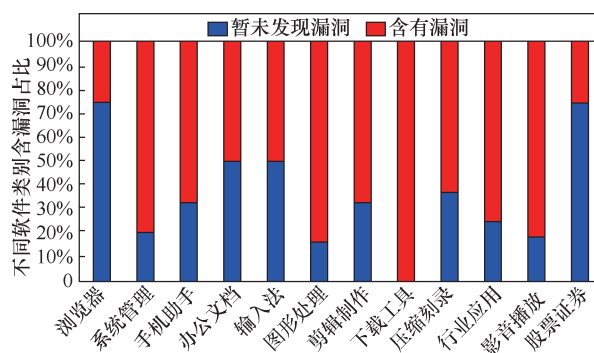


图 16 不同软件类别含漏洞占比

Figure 16 The proportion of vulnerabilities in different software types

从图 15 可以看出, 目前主要的漏洞形式是无防护与校验误用两个类型, 占全部已分析样本总数的 58%, 而重定向误用和 HTTPS 防护缺失的情况相对较少, 合计仅占全部样本的 12%。

从图 16 可以看出, 在目前已分析的 12 类软件样本中, 浏览器类中漏洞样本占比相对较少, 仅为 20%, 下载工具类中漏洞的样本占比例最高, 达到 100%, 其余类别中, 含漏洞软件的比例从 40%到 80%不等。

(2) 系统性能测试结果

系统性能测试分析如表 6 所示。

经过实际测试, 系统误报率为 7%, 系统漏报率为 10%, 原因分析如下。

表 6 系统性能测试分析

Table 6 The analysis of system performance test

类别	实际含有漏洞且	实际不含有
	分类正确样本数	漏洞样本数
系统检测含有漏洞的样本数	106	8
系统检测不含有漏洞的样本数	7	63

系统误报的样本共 8 个, 其中 3 个样本在校验过程中使用了非标准密码算法, 如截短 MD5, 系统未能有效检测, 误报为“无防护”, 实际应为“校验误用”; 5 个样本由于发起阶段使用 HTTPS 协议传输配置信息时, 其中包含了校验信息, 由于使用加密传输, 仅根据流量数据无法检测, 导致在下载阶段使用了 HTTP 协议情况下误报为“HTTPS 保护缺失”, 实际应为“无漏洞”。针对以上问题, 可在后续的工作中, 通过动态二进制插桩监控软件升级过程中的关键函数调用并提取其参数^[8]的方法对其加解密并在校验过程进行详细分析。

系统漏报的漏洞共 7 个, 其中 3 个样本在软件安装结束后, 立刻自动完成了升级的发起阶段, 系统未能记录到该阶段的流量数据, 实际 2 个样本应为“无防护”, 1 个样本应为“散列误用”, 针对此问题, 可在软件安装前即开始网络流量记录, 确保升级流量记录的完整性; 4 个样本在升级过程中未进行发起阶段, 直接通过 HTTP 协议访问固定域名下载升级包, 实际应为“无防护”, 针对此问题, 可通过人工进行辅助判断以提高准确性。

5 结束语

软件升级是应用软件的必备模块, 如果在线升级的流程设计或代码实现中存在漏洞, 将对用户主机的安全性构成严重威胁。然而现有的升级漏洞检测方法和手段, 缺乏统一的方法和标准, 效率低下、严重依赖人工, 难以满足对大批量高效分析和检测的要求。

为解决这一问题, 对大量软件样本升级流程和机制进行分析与研究, 本文提出了一种基于网络流量的升级漏洞检测方法。首先记录和分析流量, 生成画像, 并根据画像通过特征匹配进行漏洞检测, 最后对检测结果进行验证得出最终结论。本文还设计并实现基于此方法的

面向 Windows 应用软件的升级漏洞自动检测与验证系统, 系统由数据提取与分析、漏洞检测与漏洞自动验证 3 个子系统组成, 实现了升级流程信息的自动提取和分析, 以及基于此的漏洞检测与验证功能。通过对 184 个软件样本的分析测试, 检测出 117 个样本的升级漏洞, 证明了该方法和系统能够高效准确地发现软件升级机制中存在的漏洞。

参考文献:

- [1] 张海藩. 软件工程导论[M]. 北京: 清华大学出版社, 1998.
ZHANG H F. Introduction to software engineering[M]. Beijing: Tsinghua University Press, 1998.
- [2] 彭先觉, 胡勇. 利用软件在线升级的攻击[J]. 通信技术, 2015(10): 1183-1186.
PENG X J, HU Y. Attacks using online software upgrade[J]. Communications Technology, 2015 (10): 1183-1186.
- [3] 傅建明, 刘高, 李鹏伟. 一种杀毒软件升级流程的安全性分析方法[J]. 武汉大学学报(理学版). 2015(6): 509-516.
FU J M, LIU G, LI P W. A security analysis method of antivirus software upgrade process[J]. Journal of Wuhan University (Natural Science Edition). 2015 (6): 509-516.
- [4] 周振飞. 软件供应链污染机理与防御研究[D]. 北京: 北京邮电大学, 2018.
ZHOU Z F. Research on software supply chain pollution mechanism and defense[D]. Beijing: Beijing University of Posts and Telecommunications, 2018.
- [5] CATUOGNO L, GALDI C, PERSIANO G. Guaranteeing dependency enforcement in software updates[M]. Secure IT Systems. Springer International Publishing, 2015.
- [6] 赵作鹏, 尹志民, 王潜平, 等. 一种改进的编辑距离算法及其在数据处理中的应用[J]. 计算机应用, 2009, 29(2): 424-426.
ZHAO Z P, YIN Z M, WANG Q P, et al. An improved edit distance algorithm and its application in data processing [J]. Journal of Computer Applications, 2009, 29 (2): 424-426.
- [7] 刘倩. 基于 Kali Linux 的网络安全技术探讨与研究[D]. 长春: 吉林大学, 2018.
LIU Q. Discussion and research on network security technology based on Kali Linux [D]. Changchun: Jilin University, 2018.
- [8] 王乾. 基于动态二进制分析的关键函数定位技术研究[D]. 郑州: 信息工程大学, 2012.
WANG Q. Research on key function location technology based on dynamic binary analysis [D]. Zhengzhou: Information Engineering University, 2012.

[作者简介]



腾金辉 (1996-), 男, 河南洛阳人, 战略支援部队信息工程大学助理研究员, 主要研究方向为基于大数据的安全机制分析以及漏洞利用。



光焱 (1983-), 男, 安徽枞阳人, 博士, 战略支援部队信息工程大学讲师, 主要研究方向为安全机制分析与漏洞挖掘。



舒辉 (1974-), 男, 江苏盐城人, 战略支援部队信息工程大学教授、博士生导师, 主要研究方向为网络安全、漏洞检测。



张冰 (1989-), 男, 河南郑州人, 硕士, 主要研究方向为软件工程。

《网络与信息安全学报》

第二届编辑委员会

主 任 委 员：方滨兴

常务副主任委员：李风华

副 主 任 委 员：李 晖 俞能海 Kui Ren (USA) 刘建伟

邹德清 杜学绘 刘华鲁

委 员：（按姓氏笔画排序）

丁 勇 马建峰 王 标 王布宏 王连海

王良民 王国胤 王明生 王美琴 牛 犇

仇洪冰 田有亮 田俊峰 史国振 付章杰

冯 涛 朱建明 延志伟 仲 盛 向 涛

刘 哲 刘 渊 刘西蒙 刘哲理 闫 峥

孙东红 杨宏宇 肖 甫 肖 亮 吴 铤

吴晓平 邱卫东 何道敬 沈 超 张小松

张云勇 张红旗 张宏莉 张建标 张振江

陆月明 陈立全 陈兴蜀 陈性元 陈雷霆

苗春雨 林 鹏 易东山 赵 波 赵琉涛

咸鹤群 禹 勇 姚志强 秦 拯 贾 焰

殷丽华 翁 健 黄欣沂 曹春杰 彭国军

蒋文保 韩伟力 韩劲松 程 光 阚海斌

David Naccache (FRA) Gang Qu (USA) Jinjun Chen (AUS)

Willy Susilo (AUS) Yier Jin (USA)

收录声明

本刊对发表的文章,拥有出版电子版、网络版版权,并拥有和其他网站交换信息的权利。本刊支付的稿酬中已经包含上述费用。

Chinese Journal of Network and Information Security has the copyright to publish electronic edition, online edition of the published articles, and has the right to exchange information with other sites. The expenses have been included in the fee paid by editorial department.

道德声明

本刊发表的论文是作者独立取得的原创性研究成果,无一稿多投;论文内容不涉及国家机密;未曾以任何形式用任何文种在国内外公开发表过;论文内容不侵犯他人著作权和其他权利。若发生一稿多投、侵权、泄密等问题,论文作者将承担全部责任。

The authors of *Chinese Journal of Network and Information Security* guarantee that their submitted articles are original and contain nothing confidential. The said article is only submitted to *Chinese Journal of Network and Information Security*. The said article has not been published before and has not been submitted elsewhere for print or electronic publication consideration. The said article is no way whatever a violation or an infringement of any existing copyright or license from the third party. Otherwise, the authors of the said article shall take the blame for the violation or infringement of the related copyright and the leakage of secrets.